

UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA QUÍMICA

ÁREA DE CONCENTRAÇÃO

SISTEMAS DE PROCESSOS QUÍMICOS E INFORMÁTICA

**CONTROLE *FEEDFORWARD* BASEADO EM REDES NEURAIS APLICADO A  
COLUNA DE ABSORÇÃO DO PROCESSO DE PRODUÇÃO DE ETANOL**

Autor: Eduardo Eyng

Orientadora: Dra. Ana Maria Frattini Fileti

Co-orientador: Dr. Fernando Palú

Dissertação de Mestrado apresentada à Faculdade de Engenharia Química como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Química.

Campinas – São Paulo

Janeiro/2006

UNIDADE BC  
Nº CHAMADA \_\_\_\_\_  
V \_\_\_\_\_ EX \_\_\_\_\_  
TOMBO BC/ 68104  
PROC 16-123-06  
C \_\_\_\_\_ D x  
PREÇO 11,00  
DATA 19-04-06  
Nº CPD \_\_\_\_\_

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Ey56c Eyng, Eduardo  
Controle *feedforward* baseado em redes neurais aplicado  
a coluna de absorção do processo de produção de etanol /  
Eduardo Eyng.--Campinas, SP: [s.n.], 2006.

Orientadores: Ana Maria Frattini Fileti, Fernando Palú  
Dissertação (mestrado) - Universidade Estadual de  
Campinas, Faculdade de Engenharia Química.

1. Sistema de controle Feedforward. 2. Absorção  
química. 3. Redes neurais (Computação). 4. Controle de  
processo. I. Fileti, Ana Maria Frattini. II. Palú, Fernando.  
III. Universidade Estadual de Campinas. Faculdade de  
Engenharia Química. IV. Título.

Título em Inglês: Feedforward control based on neural networks applied to an  
absorption column of ethanol production process

Palavras-chave em Inglês: Absorption column, Artificial neural networks, Feedforward  
control

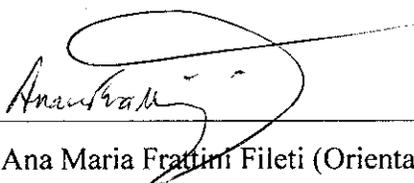
Área de concentração: Sistemas de Processos Químicos e Informática

Titulação: Mestre em Engenharia Química

Banca examinadora: Flávio Vasconcelos da Silva, Alexandre Nunes Ponezi

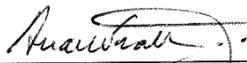
Data da defesa: 05/01/2006

Este exemplar corresponde à versão final da Dissertação de Mestrado em Engenharia  
Química.

A handwritten signature in black ink, appearing to read 'Ana Maria Frattini Fileti', is written over a horizontal line. The signature is stylized and somewhat cursive.

Prof. Dra. Ana Maria Frattini Fileti (Orientadora)

Dissertação de Mestrado defendida por Eduardo Eyng e aprovada em 05 de janeiro de 2006  
pela banca examinadora constituída pelos doutores:



---

Prof. Dra. Ana Maria Frattini Fileti  
(orientadora)



---

Prof. Dr. Flávio Vasconcelos da Silva  
DESQ/FEQ/UNICAMP



---

Prof. Dr. Alexandre Nunes Ponezi  
CPQBA/UNICAMP

## AGRADECIMENTOS

A Deus acima de tudo, pela iluminação e força.

Aos meus pais, pelo apoio incondicional.

À minha noiva, pelo incentivo e compreensão.

À minha irmã, por sua paciência.

À Dra. Ana Maria Frattini Fileti, pela orientação, experiência e sugestões.

Ao Dr. Fernando Palú, pelo conhecimento transmitido.

Ao Cnpq pelo apoio financeiro.

## RESUMO

O etanol perdido por evaporação durante o processo de produção por fermentação pode ser recuperado por uma coluna de absorção, a qual requer um sistema de controle robusto de acordo com sua complexidade. Sendo assim, no presente trabalho foi proposto e testado, o emprego de um controlador *feedforward*, baseado em um modelo inverso de redes neurais, para realizar a manipulação da vazão de solvente que é alimentado a esta coluna.

O desempenho do controlador *feedforward* proposto foi superior ao proporcionado pelo Controle por Matriz Dinâmica (DMC) (Palú, 2001), para situações nas quais uma perturbação degrau foi inserida na composição do gás à entrada da coluna.

Outro teste realizado foi o de implementar um distúrbio não previsto na etapa de treinamento da rede neural, e analisar o comportamento do sistema de controle RNA. Houve uma queda de eficiência, sendo gerado *off set*. Para solucionar este problema, agregou-se ao sistema de controle, um dispositivo de segurança.

Também se procurou testar a tolerância do sistema de controle a distorções nos dados de entrada da rede. Neste sentido, acrescentou-se às variáveis de entrada do tipo concentração, um ruído. O desempenho do controlador RNA foi prejudicado, no entanto, a faixa de oscilação da variável controlada foi reduzida à aproximadamente 50% da faixa de ruído empregada. Aplicando-se um filtro aos dados distorcidos, esta redução foi ainda maior, cerca de 75%.

## ABSTRACT

Some of ethanol lost by evaporation during its fermentation production process may be recovered using an absorption column, which requires a robust control system. In this work the use of a feedforward controller (NN controller), based on a neural network inverse model, was proposed and tested to manipulate the solvent flow, which was fed in the analyzed column.

The feedforward controller performance was better than the Dynamic Matrix Control (Palú, 2001) one, when step disturbances were inserted on the inlet gas composition.

Another test done was the insert of a not trained disturbance. The NN controller performance decrease, and a off set could be observed. To solve this problem, a security device was added to the control system.

The control system tolerance to modified input neural network data also was tested. In this way, a noise was added to input data. The NN control performance decreased, but the amplitude of the control variable oscillation band was reduced to the half of the noise amplitude employed. When the modified data was filtered the NN control performance decrease was smaller, about 75% of the noise amplitude.

## SUMÁRIO

<b>NOMENCLATURA</b>	<b>x</b>
<b>CAPÍTULO 1 – INTRODUÇÃO</b>	<b>1</b>
1.1 OBJETIVOS	4
<b>CAPÍTULO 2 – REVISÃO BIBLIOGRÁFICA</b>	<b>5</b>
2.1 INTRODUÇÃO	6
2.2 COLUNAS DE ABSORÇÃO	6
2.2.1 MODELO DINÂMICO DA COLUNA (MAIA, 1994)	10
2.3 REDES NEURAIS ARTIFIAIS	17
2.3.1 TREINAMENTO	17
2.3.2 PERCEPTRON	18
2.3.3 MULTICAMADAS DE NEURÔNIOS ARTIFICIAIS	20
2.3.3.1 ALGORITMO DA RETROPROPAGAÇÃO (OU <i>BACKPROPAGATION</i> )	21
2.3.3.2 ALGORITMO DE LEVENBERG-MARQUARDT COM REGULARIZAÇÃO BAYESIANA	29
2.4 CONTROLADOR PROPORCIONAL INTEGRAL (PID)	31
2.5 PROPOSTA DA DISSERTAÇÃO	32
<b>CAPÍTULO 3 – CONTROLE DA COLUNA DE ABSORÇÃO DE PRATOS</b>	<b>33</b>
3.1 INTRODUÇÃO	34
3.2 CONTROLE POR MATRIZ DINÂMICA	34
3.3 CONTROLE VIA REDES NEURAIS	36
3.3.1 CONSTRUÇÃO DA REDE NEURAL	36
3.3.2 TREINAMENTO DA REDE NEURAL	39
3.3.3 IMPLEMENTAÇÃO DO DISPOSITIVO DE SEGURANÇA	40
3.3.4 DISTORÇÃO E FILTRAGEM DE DADOS DE ENTRADA DA REDE	41
<b>CAPÍTULO 4 – RESULTADOS</b>	<b>42</b>
4.1 INTRODUÇÃO	43
4.2 RESULTADOS PRELIMINARES	43

4.3 TREINAMENTO DA REDE NEURAL ARTIFICIAL	46
4.3.1 CONSTRUÇÃO DOS BANCOS DE DADOS DE TREINAMENTO E VALIDAÇÃO	46
4.3.2 DETERMINAÇÃO DA ESTRUTURA DA REDE	47
4.4 COMPARAÇÃO DE DESEMPENHO (DMC E RNA)	50
4.5 SIMULAÇÃO COM O DISPOSITIVO DE SEGURANÇA (PERTURBAÇÕES NÃO PREVISTAS NO TREINAMENTO)	57
4.6 IMPLEMENTAÇÃO DO RUÍDO	63
4.7 IMPLEMENTAÇÃO DO FILTRO	66
<b>CAPÍTULO 5 – CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS</b>	<b>72</b>
5.1 CONCLUSÕES	73
5.2 SUGESTÕES PARA TRABALHOS FUTUROS	74
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>76</b>
<b>ANEXOS</b>	<b>79</b>
ANEXO 01 – PROGRAMA COMPUTACIONAL EM MATLAB® MODELO_COLUNA_DE_ABSORCAO.M	80
ANEXO 02 – PROGRAMA COMPUTACIONAL EM MATLAB® REDE_COLUNA_DE_ABSORCAO_TREINAMENTO.M	90
ANEXO 03 – ARQUIVOS SECUNDÁRIOS EM MATLAB®	94

## NOMENCLATURA

$A_p$  = área do prato

$b_k$  = Termo Limiar

$c$  = constante

$d$  = saída desejada para a rede

$e_j$  = erro correspondente a uma iteração para o neurônio  $j$

$E$  = erro correspondente a todos os nódulos de saída

$\bar{E}$  = função objetivo

$f$  = fator de supressão

FO = função objetivo para o método de Levenberg-Marquardt com regularização Bayesiana

$G$  = vazão de gás

$H$  = aproximação da matriz de Hessian

$h_v$  = Altura do vertedouro

ISE = integral do quadrado do erro

$J$  = Matriz Jacobiana

$K_c$  = ganho do controlador

$L$  = vazão de solvente

$L_0$  = vazão de solvente à entrada da coluna

$L_w$  = comprimento do vertedouro

$M$  = massa molar de líquido contida em cada estágio

MSE = média dos quadrados dos erros

$N$  = número de estágios

$N'$  = número total de parâmetros da rede

SSE = Soma dos quadrados dos erros

SSW = Soma dos quadrados dos pesos

$T$  = horizonte do processo

$T_a$  = tempo de amostragem

$U$  = horizonte de controle

$V$  = horizonte de predição

$v_k$  = entrada da função de ativação

$w_{ik}$  = peso sináptico do neurônio  $k$

$x$  = entrada da rede

$X$  = fração molar do soluto na fase líquida

$X_0$  = fração molar do álcool no solvente à entrada da coluna

$y$  = saída calculada pela rede

$Y$  = fração molar do soluto na fase gasosa

$Y_1$  = fração molar do álcool no gás à saída da coluna

$Y_{1SP}$  = *set point*

$Y_{N+1}$  = fração molar do álcool no gás à entrada da coluna

### Letas Gregas

$\alpha$  = termo de momentum

$\alpha'$  = parâmetro da função objetivo FO

$\beta'$  = parâmetro da função objetivo FO

$\gamma$  = parâmetros efetivamente utilizados pela rede

$\Delta t$  = passo de integração

$\Delta w$  = correção dos pesos sinápticos

$\eta$  = taxa de aprendizagem

$\theta_d$  = atraso de transporte

$\lambda$  = parâmetro do filtro

$\mu$  = escalar

$\rho_n$  = massa específica molar média da mistura

$\tau$  = constante de tempo

$\tau_I$  = constante de tempo integral

$\varphi$  = função de ativação

# **CAPÍTULO 1**

## **INTRODUÇÃO**

Durante a operação de uma determinada planta química, alguns objetivos específicos devem ser alcançados, para que esta se mostre viável. Entre eles, deve-se destacar a garantia de uma operação segura tanto para as pessoas diretamente ligadas à indústria, quanto para a sociedade que a cerca. Assim, devido ao padrão de qualidade exigido, mercado competitivo, constantes mudanças na economia, além do alto nível de integração entre as várias unidades de processamento de uma planta química, faz-se necessário a implementação de um robusto sistema de controle.

Segundo NASTARI (2005) a área destinada ao plantio de cana-de-açúcar no Brasil, no ano de 2005, foi de aproximadamente 2,6 milhões de hectares. Com um rendimento médio de 5900 litros de álcool por hectare, a produção de etanol brasileira atingiu a marca de 15,34 bilhões de litros. No entanto com o crescimento da demanda de álcool impulsionado principalmente pelo aumento do preço do barril de petróleo, prevê-se que a demanda para o ano de 2013 chegue a 30,85 bilhões de litros. Neste sentido, toda alternativa para o aumento da produtividade do etanol merece ser analisada.

Ao longo do processo de fermentação, para a produção de álcool etílico ( $C_2H_6O$ ), acontecem perdas do produto por evaporação, diminuindo assim, o rendimento alcoólico. Estas perdas variam com a concentração dos mostos, temperatura de fermentação, formato das dornas, entre outros fatores, sendo mais significativas em países tropicais, onde se encontra dificuldade em se realizar uma boa refrigeração do mosto de fermentação. A recuperação deste álcool pode ser feita alimentando-se os vapores captados sobre as dornas a uma coluna de absorção de pratos. Para tanto, água é adicionada em contracorrente à mistura gasosa de álcool e dióxido de carbono ( $CO_2$ ). O objetivo do sistema de controle consiste em assegurar que a composição do gás à saída da coluna permaneça o mais próximo possível da desejada (*set point*), sendo que para tanto, é efetuada a manipulação da vazão de solvente (água). Entretanto, a coluna de absorção de pratos é um equipamento caracterizado pela sua não linearidade, e em consequência disso, técnicas convencionais de controle encontram dificuldades na garantia de um desempenho satisfatório.

Na busca de um sistema de controle compatível com a complexidade da maioria dos processos industriais, várias pesquisas têm sido elaboradas de acordo com o desenvolvimento de novas ferramentas e tecnologias. Neste sentido, o uso de redes neurais artificiais (RNAs) torna-se cada vez mais, uma opção atrativa, haja vista a atual ampliação da capacidade computacional. Devido às características não-lineares inerentes ao

mapeamento entre camadas de RNAs *feedforward*, estas se caracterizam como ferramentas de modelamento bastante apropriadas para o modelamento e controle de sistemas. Ferramentas matemáticas disponíveis para o controle de sistemas não-lineares normalmente utilizam técnicas de linearização, transformando a tarefa de controle não-linear em pequenas tarefas de controle linear. Apesar de muitas vezes eficiente, esta abordagem não retrata a realidade dos sistemas físicos, podendo resultar em soluções subótimas. Devido à capacidade de modelar com fidelidade ações não-lineares, as RNAs são uma importante ferramenta para o controle de sistemas (BRAGA, 2000).

Em virtude da complexidade do sistema estudado, e conhecendo-se a capacidade de processamento das redes neurais, acredita-se que uma RNA *feedforward*, depois de devidamente treinada, seja capaz de determinar como deve ser feita a manipulação da vazão de solvente em uma coluna de absorção de pratos. A avaliação desta metodologia de controle será realizada pela sua aplicação na recuperação do álcool perdido por evaporação em um processo de fermentação.

Depois de uma breve explanação sobre o problema da perda, por evaporação, de parte do álcool etílico produzido durante o processo de fermentação, além da necessidade de implementação de um sistema de controle robusto para a coluna de absorção de pratos destinada a minimizar tal problema, segue-se com a organização do presente trabalho.

No capítulo 2 é realizada uma revisão dos trabalhos disponíveis na literatura, que são relacionados à modelagem e controle de colunas de absorção. Além disso, são apresentados, de forma sucinta, conceitos teóricos sobre o algoritmo de treinamento utilizado para a rede neural. A partir deste levantamento bibliográfico foi possível direcionar a pesquisa no sentido de buscar soluções alternativas para o problema estudado.

Já no capítulo 3, é abordada a caracterização do sistema estudado, incluindo-se a definição das condições de operação, além do desenvolvimento da técnica de controle proposta.

No capítulo 4 são apresentados os resultados das simulações realizadas, demonstrando-se o desempenho da estratégia de controle proposta. Para avaliar o comportamento desta, efetuou-se a comparação com a estratégia de Controle por Matriz Dinâmica (DMC), aplicada ao sistema estudado, por PALÚ (2001), sendo os parâmetros

demonstrativos de desempenho, a integral do quadrado do erro (ISE) e a quantidade de solvente gasta.

Finalmente, no capítulo 5, encontram-se as conclusões do presente trabalho, bem como as sugestões para trabalhos futuros.

## **1.1 OBJETIVOS**

Tomando como base o levantamento bibliográfico realizado, pode-se perceber que o controle de uma coluna de absorção de pratos tem sido estudado por alguns pesquisadores. Entretanto, as técnicas empregadas até o momento não possuem a capacidade de processamento demonstrada pelas redes neurais artificiais (RNA's), as quais já foram aplicadas na modelagem desta operação unitária. Deste modo, os objetivos deste trabalho são o desenvolvimento e a utilização de um controlador *feedforward* baseado em modelo inverso de redes neurais para a manipulação da vazão de solvente de uma coluna de absorção empregada em processo de fermentação alcoólica.

**CAPÍTULO 2**  
**REVISÃO BIBLIOGRÁFICA**

## **2.1 INTRODUÇÃO**

Neste capítulo serão apresentados conceitos relevantes sobre colunas de absorção, assim como redes neurais artificiais e o controlador convencional PI. Estes foram obtidos de trabalhos disponíveis na literatura, os quais representam os resultados de pesquisas realizadas anteriormente, sendo valiosos na verificação do que foi estudado até o momento e no direcionamento do presente trabalho, conforme apresentado no item Proposta da Dissertação, ao final deste mesmo capítulo.

## **2.2 COLUNAS DE ABSORÇÃO**

A absorção gasosa é uma operação que envolve o contato entre as fases líquida e gasosa, sendo que um ou mais componentes presentes no gás, são absorvidos pelo líquido.

O campo de aplicação desta operação é muito vasto, encontrando espaço tanto na recuperação de produtos quanto no tratamento de efluentes. Desta forma, este assunto tem atraído a atenção de muitos pesquisadores, que visam o aprimoramento deste processo através de inovações no projeto de equipamentos, ou ainda, no desenvolvimento de novos sistemas de controle.

Seguindo esta linha de pesquisa, é possível encontrar vários trabalhos na literatura, entre os quais se destaca o elaborado por MAIA (1994), no qual é implementada a uma coluna de absorção de gases uma das técnicas de controle preditivo (Matriz Dinâmica de Controle (DMC)). A autora considerou o caso monovariável e sem restrições deste algoritmo de controle para que se pudesse fazer um estudo comparativo entre os desempenhos deste controlador preditivo (DMC) e um controlador convencional por retroalimentação (PI). Perturbações em degrau foram implementadas na composição do gás à entrada da coluna. Foram comparadas as respostas dos dois tipos de controladores (PI e DMC), tendo como variáveis controlada e manipulada, a composição do gás à saída da coluna, e a vazão de solvente, respectivamente. Ficou caracterizada a superioridade da estratégia de controle por matriz dinâmica, quando comparada à técnica convencional por

retro-alimentação. Com a utilização do DMC, a variável controlada apresentou-se com menor sobrelevação e com um rápido reestabelecimento do estado estacionário.

Outra proposta de controle para colunas de absorção bem sucedida foi apresentada por NAJIM (1995). Neste trabalho foi proposta a utilização do controle adaptativo em uma coluna de absorção com recheio. Este equipamento teve por finalidade a redução da concentração de  $\text{CO}_2$  em uma mistura gasosa até um valor desejado. Para tanto, fez-se uso de uma solução aquosa de Dietanolamina (DEA). O modelo desenvolvido para simular a coluna foi baseado em balanço de massa, fenômenos de transporte e na reação química entre  $\text{CO}_2$  e DEA, que ocorre na fase líquida. Este modelo consiste de um sistema não linear de três equações diferenciais parciais. A vazão de solvente e a concentração de  $\text{CO}_2$  no topo da coluna foram, respectivamente, a variável manipulada e a controlada. A vazão de DEA foi controlada por uma válvula, sendo que a voltagem que alimenta o motor elétrico associado a essa válvula foi limitada, assegurando assim que a coluna não sofresse inundação. Os resultados obtidos asseguram a eficácia do modelo, assim como, a do sistema de controle proposto.

Já em 1997, SILVA realizou na primeira etapa de seu trabalho, o estudo dos perfis de concentração e fluxos mássicos ao longo de colunas recheadas de absorção com reação química, utilizando-se para tanto, as equações fundamentais de transferência de massa com o termo de reação química. Estas equações foram resolvidas numericamente, pelo método da colocação ortogonal em elementos finitos. Visando proceder a uma simulação mais rigorosa do processo, o autor utilizou o método das Características. Pelo fato deste necessitar do cálculo dos fluxos de transferência de massa em diversos pontos ao longo da coluna, dificultando assim, a sua utilização em conjunto com a resolução numérica das equações de transferência de massa, foi desenvolvido um modelo neural híbrido capaz de prever tais fluxos. Desta forma, o modelo utilizado nas simulações proporcionou a inclusão de importantes características hidrodinâmicas do processo. Na etapa final do trabalho, o autor avaliou o desempenho do controle preditivo com modelo linear (convolução) e não-linear (baseado em redes neurais artificiais). Os resultados demonstraram uma diferença de desempenho pouco significativa entre os dois controladores. Segundo SILVA, isto se deve em parte, a robustez proporcionada pelo controle preditivo, independentemente do modelo interno utilizado. Além disso, outro fator responsável pelo nivelamento de desempenho entre os controladores, seria a grande inércia

do sistema frente a variável manipulada. Desta forma, pela simplicidade de aplicação e o fornecimento de resultados razoáveis, a utilização do controle preditivo com modelo de convolução pode ser justificada.

Em 1998, RÓJ e WILK apresentaram uma aplicação de redes neurais, tornando-se possível a simulação de uma coluna de absorção. Utilizando-se uma rede de arquitetura 14-14-10, treinada através do algoritmo da retropropagação com uso da regra delta generalizada, foi possível simular a coluna de absorção com sucesso. Os dados utilizados no processo de treinamento foram obtidos do modelo rigoroso da coluna, que também gerou um conjunto de dados adicional, destinado à comparação com os resultados fornecidos pela rede. Apesar de o sistema ser não linear e multivariável, a rede neural desenvolvida foi capaz de modelar e simulá-lo com sucesso.

No ano de 1999, PARTHASARATHY e colaboradores publicaram um artigo no qual apresentavam um modelo capaz de prever a inundação de uma coluna de absorção, sendo este utilizado como uma variável controlada em um controlador multivariável. Inicialmente, identificou-se um conjunto de variáveis representativas do processo de inundação. Diferentes tipos de modelos foram construídos e testados *off-line*, e devido a imprecisão do modelo linear, verificou-se que o processo apresentava não linearidades suficientes para justificar a utilização de um modelo baseado em redes neurais artificiais. Vários modelos baseados em redes neurais foram desenvolvidos e testados *on-line*, ficando caracterizado que a robustez e precisão destes, dependia tanto da qualidade, quanto da quantidade dos dados utilizados na etapa de treinamento. Desta forma, o modelo que apresentou os melhores resultados teve como variáveis de entrada a queda de pressão em uma determinada seção da coluna no instante atual e três minutos no passado, a pressão no topo da coluna, a vazão de gás purificado, nível de distúrbio na vazão de gás purificado, temperatura do gás a entrada da coluna e do solvente e a concentração de gás carbônico no gás alimentado a coluna. Assim, com uma estrutura composta por uma camada escondida com cinco neurônios, e utilizando a função de ativação tangente hiperbólica, o modelo desenvolvido foi capaz de prever, com sucesso, a queda de pressão em um instante de tempo de três minutos no futuro, fornecendo, portanto, informações valiosas ao sistema de controle central.

Dando continuidade ao trabalho de MAIA, PALÚ (2001) realizou um estudo do desempenho do Controle por Matriz Dinâmica (DMC), quando aplicado a uma coluna de

absorção, tanto para o caso monovariável (SISO), como para o caso multivariável (MIMO). Além de realizar o controle do sistema quando submetido a uma única perturbação no instante de tempo inicial, também se avaliou o desempenho do controlador para perturbações múltiplas. Estas perturbações, de magnitudes aleatórias, foram inseridas de forma periódica, visando aproximar o funcionamento da coluna de uma situação real. Em ambos os casos, comparou-se a resposta obtida pela estratégia de controle DMC com a resposta fornecida por um controlador convencional de retro-alimentação (PI), sendo a qualidade da primeira, muito superior a da segunda. Finalizando o trabalho, se propôs otimizar o funcionamento da coluna através da escolha adequada do(s) estágio(s) usado(s) para medir a composição do gás. Para quantificar o desempenho do sistema de controle foram propostos dois parâmetros: a quantidade de solvente gasta e a integral do quadrado do erro (ISE). Com o uso destes dois critérios foi possível estabelecer que a composição do gás deve ser medida no primeiro prato da coluna, assim, assegura-se o melhor desempenho do controlador.

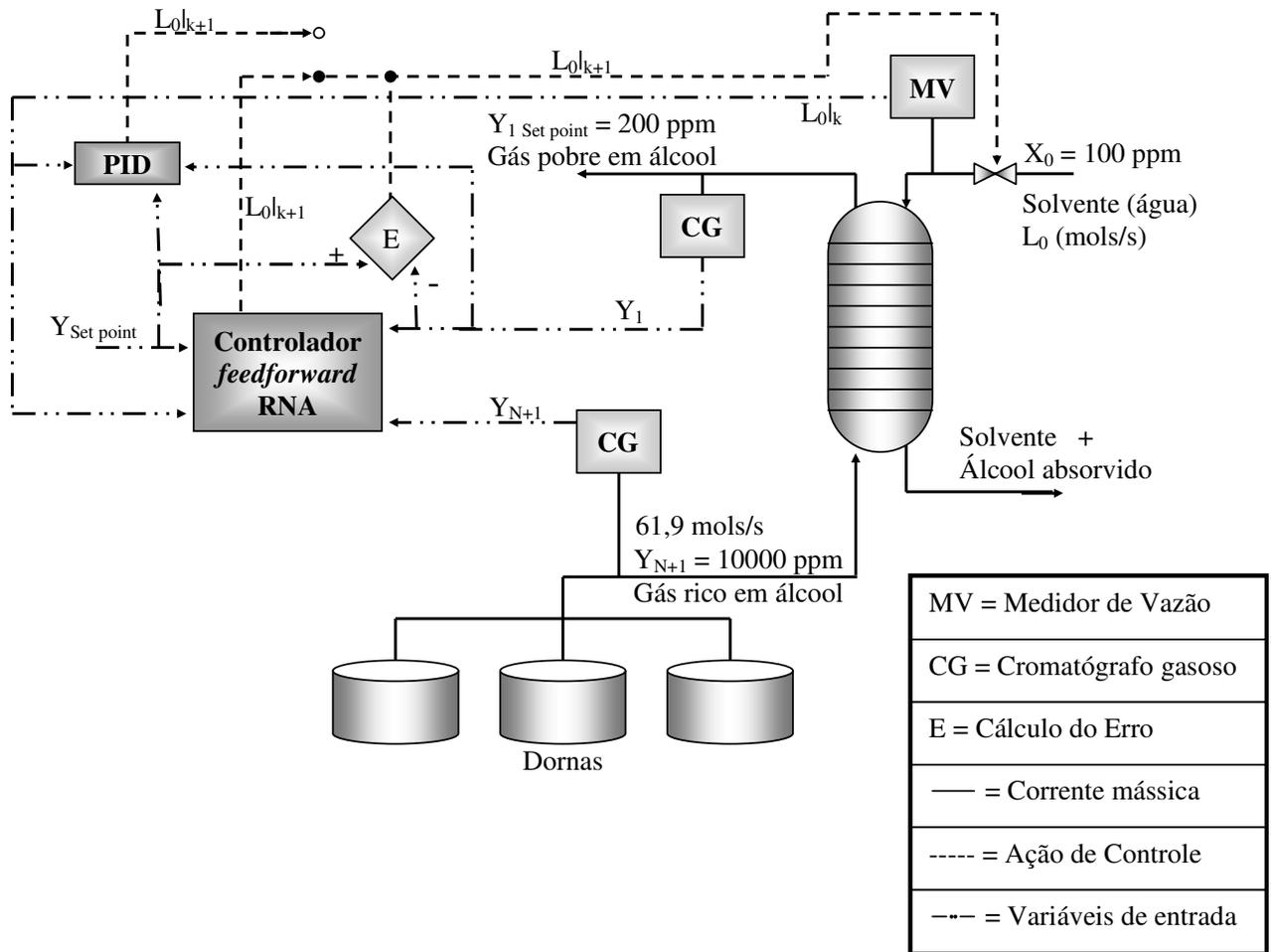
Devido a dificuldade de previsão de estabilidade quando implementado um conjunto de parâmetros em um sistema de controle preditivo, NUNES (2003) e colaboradores apresentaram um método rigoroso de análise. A técnica empregada permite derivar expressões explícitas para as funções transferência. Assim, os pólos do sistema podem ser determinados encontrando as raízes de dois polinômios característicos, permitindo, portanto, a caracterização da estabilidade do sistema. Devido ao grande número de parâmetros a serem ajustados para o projeto de um controlador preditivo multivariável torna-se complexo. Uma estimativa equivocada destes parâmetros pode deixar algumas relações entrada-saída instáveis. Estas instabilidades podem ser “camufladas” pelas restrições físicas do processo que impedem que o sistema siga descontroladamente. No entanto, estas restrições fazem com que o sistema opere em condições sub-ótimas. Com a ferramenta de análise apresentada neste artigo, o projetista pode testar a estabilidade do sistema antes de implementar um conjunto de parâmetros. Para validar a técnica, o autor aplicou-a à uma coluna de absorção. Após a implementação de um primeiro conjunto de parâmetros, de acordo com o que fora detectado pela técnica, o sistema apresentou instabilidade. Já depois da implementação de um segundo conjunto de parâmetros, o sistema mostrou-se estável, conforme previsto pela análise.

### 2.2.1 MODELO DINÂMICO DA COLUNA (MAIA, 1994)

O modelo matemático para uma coluna de absorção de N estágios, empregado neste trabalho foi obtido por meio de balanços de massa globais para cada estágio n, balanços de massa globais para o componente que é absorvido em cada estágio n, relação de equilíbrio, equação de Francis, além de algumas simplificações, sendo elas:

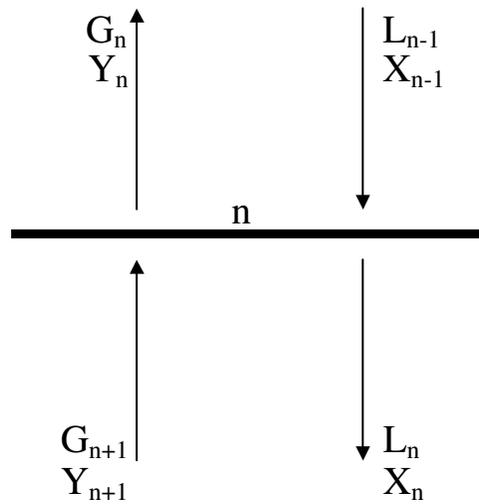
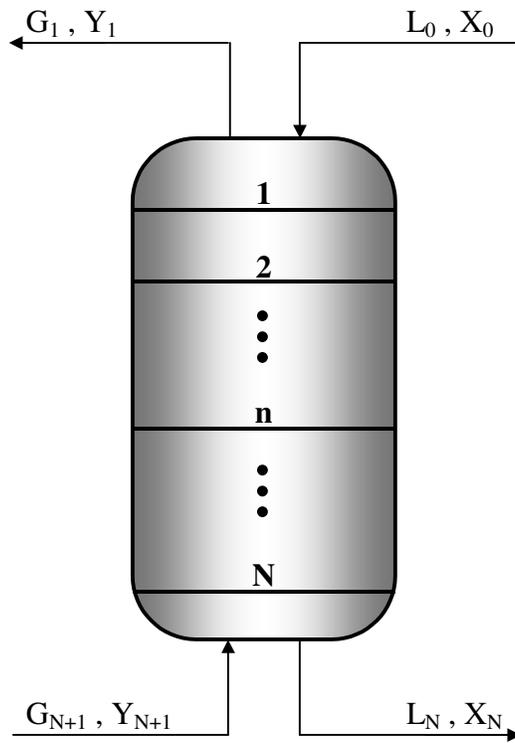
1. Apenas um componente é transferido de uma fase para outra.
2. A absorção é considerada isotérmica.
3. Considera-se cada estágio como ideal.
4. A transferência de soluto entre as fases não altera as vazões, nem do gás e nem do líquido.
5. A pressão é constante ao longo da coluna.
6. Assume-se que nenhuma quantidade de gás fica retida entre os estágios.

Para uma melhor visualização do sistema estudado, na Figura 2.1 é apresentado um esquema geral. Nesta, é possível observar as correntes de entrada e saída da coluna de absorção estudada, assim como as variáveis de entrada dos controladores RNA e PID (este controlador opera como dispositivo de segurança, sua função será mais bem explanada no item 3.3.3) e a ação de controle.



**Figura 2.1 – Esquema geral do sistema estudado**

A numeração dos pratos da coluna de absorção é feita de forma descendente. Deste modo, a coluna é alimentada com o gás no N-ésimo prato, e com o solvente no primeiro. A seguir apresenta-se (Figura 2.2) um esquema da coluna de absorção, com suas correntes de entrada e saída, bem como suas respectivas frações molares. Além disso, é apresentado o estágio genérico “n” em detalhe.



**Figura 2.2 – Esquema da coluna de absorção**

Onde:

$X$  = Fração molar do soluto na fase líquida.

Y = Fração molar do soluto na fase gasosa.

L = Vazão de solvente.

G = Vazão de gás.

N = Número de estágios.

Visando obter equações que forneçam suporte para a simulação da dinâmica da coluna de absorção, tem-se:

Balanço de massa global para o estágio n:

$$\frac{dM_n}{dt} = L_{n-1} - L_n \quad (2.01)$$

Balanço de massa global para o componente que é absorvido no estágio n:

$$\frac{d(M_n X_n)}{dt} = L_{n-1} X_{n-1} - L_n X_n + G(Y_{n+1} - Y_n) \quad (2.02)$$

Onde:

M = Massa molar de líquido contida em cada estágio.

A relação de equilíbrio líquido-vapor utilizada neste trabalho é da forma:

$$Y_n = m \cdot X_n \quad (2.03)$$

Para obter a massa específica molar média de uma mistura binária (componentes A e B) faz-se uso da seguinte expressão:

$$\rho_{m\u00e9dia} = \rho_A X_A + \rho_B (1 - X_A) \quad (2.04)$$

A reten\u00e7\u00e3o de l\u00edquido no prato \u00e9 dada pela Equa\u00e7\u00e3o de Francis:

$$M_n = A_p \rho_n \left[ h_v + c \left( \frac{L_n}{\rho_n L_w} \right)^{2/3} \right] \quad (2.05)$$

na qual:

$\rho_n$  = Massa espec\u00edfica molar m\u00e9dia da mistura (moles/cm<sup>3</sup>)

$A_p$  = \u00c1rea do prato (cm<sup>2</sup>)

$c$  = Constante (cm<sup>-1/3</sup>s<sup>2/3</sup>)

$h_v$  = Altura do vertedouro (cm)

$L_w$  = Comprimento do vertedouro (cm)

Para se obter a vaz\u00e3o de l\u00edquido que deixa cada prato ( $L_n$ ), basta isolar esta vari\u00e1vel na Equa\u00e7\u00e3o (2.05):

$$L_n = \rho_n L_w \left[ \frac{1}{c} \left( \frac{M_n}{A_p \rho_n} - h_v \right) \right]^{3/2} \quad (2.06)$$

Al\u00e9m das equa\u00e7\u00f5es apresentadas at\u00e9 ent\u00e3o, para que se possa realizar a simula\u00e7\u00e3o da coluna de absor\u00e7\u00e3o de pratos, as Equa\u00e7\u00f5es diferenciais ordin\u00e1rias 2.01 e 2.02 devem ser integradas numericamente. Neste trabalho, o m\u00e9todo escolhido para realizar tal tarefa foi o

de Euler explícito, pois segundo LUYBEN (1990), este método, dentre as várias opções disponíveis na literatura, possui a vantagem de ser eficaz e facilmente programável, desde que o passo de integração utilizado seja adequado (quanto menor o valor deste parâmetro, mais preciso será o resultado da integração, sendo o limite máximo recomendado o valor correspondente a duas vezes a constante de tempo do processo).

As Equações 2.01 e 2.02, depois de serem integradas passam a apresentar a seguinte forma:

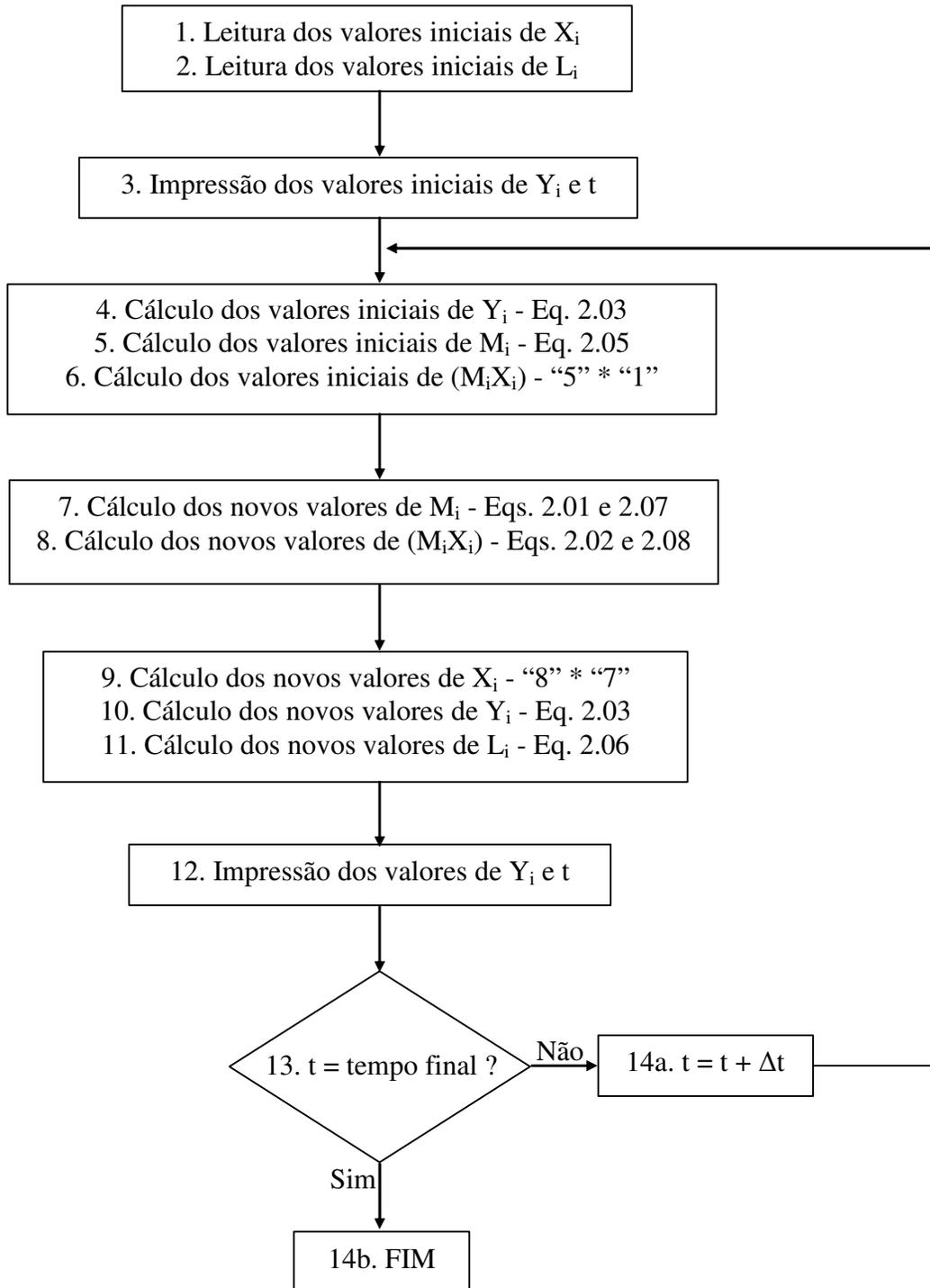
$$M_n|_{t+\Delta t} = M_n|_t + \left[ \left( \frac{d}{dt}(M_n) \right) \Big|_t \right] \Delta t \quad (2.07)$$

$$(M_n X_n)|_{t+\Delta t} = (M_n X_n)|_t + \left[ \left( \frac{d}{dt}((M_n X_n)) \right) \Big|_t \right] \Delta t \quad (2.08)$$

Onde:

$\Delta t$  = Passo de integração.

De posse das Equações 2.01 a 2.08 é possível simular o comportamento dinâmico da coluna. Assim sendo, na Figura 2.3 é apresentado um fluxograma contendo todas as etapas deste processo.



**Figura 2.3 – Fluxograma do processo de simulação da coluna de absorção**

## 2.3 REDES NEURAIIS ARTIFICIAIS

### 2.3.1 TREINAMENTO

A utilização da rede neural artificial para um determinado fim está diretamente vinculada à eficiência com que esta é capaz de fornecer respostas próximas, o suficiente, dos dados de saída reais. Para tanto, o neurônio artificial deve ser capaz de aprender uma determinada tarefa.

Na fase de aprendizagem, a rede extrai informações relevantes de padrões de informação apresentados a ela, criando assim uma representação própria do problema. Nesta etapa, portanto, há o ajuste dos parâmetros da rede, que são caracterizados pelos pesos das conexões entre as unidades de processamento. Deste modo, ao final do processo de treinamento, a rede terá adquirido conhecimento sobre o ambiente em que está operando, sendo este, “armazenado” em seus parâmetros.

Existem vários métodos para realizar o treinamento de redes neurais artificiais, no entanto, quando estas são aplicadas na modelagem de processos, utiliza-se o aprendizado supervisionado.

O objetivo deste método é o de ajustar os parâmetros da rede, de modo a estabelecer uma ligação entre os pares de entrada e saída fornecidos por um supervisor externo.

O supervisor é responsável por direcionar o processo de treinamento. Para tanto, a resposta calculada pela rede é comparada a resposta desejada, fornecida pelo próprio supervisor, a qual constitui uma ação ótima a ser tomada pela rede. A diferença existente entre estes dois valores representa o erro gerado pelo cálculo da rede, o qual deve ser minimizado através do ajuste dos pesos das conexões.

A minimização do erro é incremental, pois a cada etapa de treinamento, isto é, a cada resposta gerada pela rede e, conseqüentemente, a cada atualização do valor do erro, são efetuados pequenos ajustes nos pesos das conexões.

A implementação do aprendizado supervisionado pode ser feita tanto *off-line* quanto *on-line*, sendo que no primeiro caso, os dados do conjunto de treinamento não

mudam, e caso seja necessário a inclusão de novos dados, um outro banco de dados destinado ao treinamento deve ser elaborado, contendo os dados antigos acrescido dos novos. Já para o segundo caso, a rede opera em constante processo de adaptação, sendo atualizada continuamente.

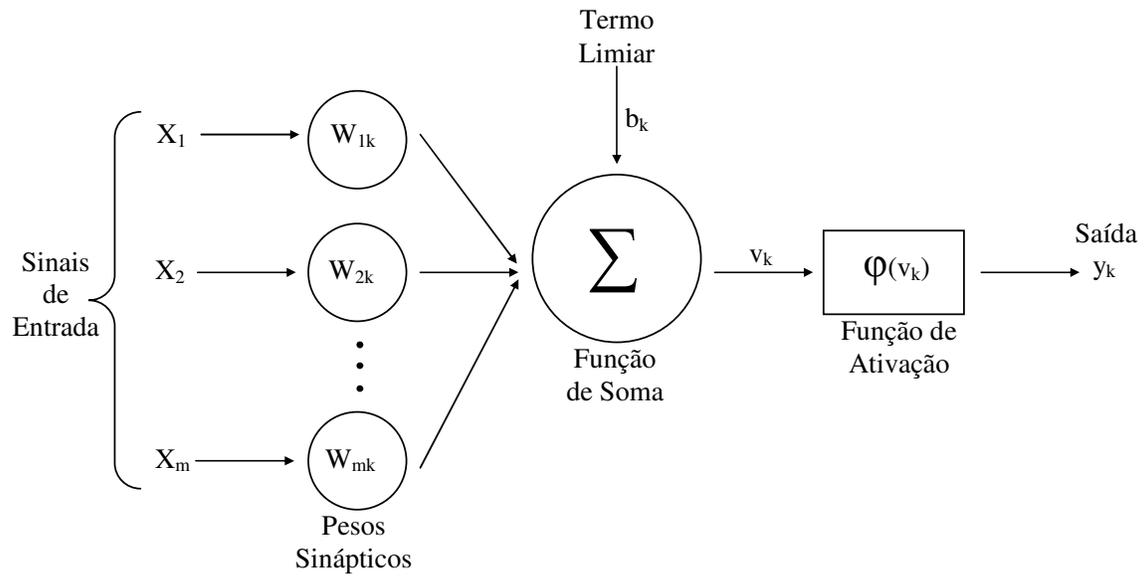
### **2.3.2 PERCEPTRON**

A função básica do neurônio biológico é somar todas as suas entradas, e produzir uma saída se esta soma for maior que um determinado valor, conhecido como valor limiar. As entradas chegam até os neurônios ao longo dos dendritos, os quais são conectados com as saídas por meio de outros neurônios através de junções chamadas sinapses. Estas junções alteram a eficiência com que o sinal é transmitido; algumas sinapses são boas junções, e passam um forte sinal adiante, enquanto que outras, muito ruins, permitem a passagem de um sinal fraco. O corpo celular recebe todas estas entradas, e dispara, se o total de entradas exceder o valor limiar. (BEALE, 1990).

Em 1943, MCCULLOCH e PITTS deram ênfase à modelagem de um neurônio biológico e sua capacidade computacional, empregada na execução de funções booleanas. O modelo proposto (denominado, nodo MCP) substituía os dendritos por entradas, e estabelecia que as ligações destas com o corpo celular artificial, seriam realizadas através de elementos chamados de pesos (fatores que multiplicados às suas respectivas entradas, simulariam as sinapses). Deste modo, uma sinapse de maior eficiência, ou seja, que transmite um sinal forte, seria representada pela associação de um peso de maior valor à sua entrada. Além disso, os estímulos captados pelas entradas seriam processados em uma função de soma, sendo que o resultado desta, seria alimentado à uma função de ativação, responsável pela produção da saída.

Apesar de muito importante, o trabalho de MCCULLOCH e PITTS (1943) não incluía técnicas de aprendizagem para o modelo desenvolvido. Mais tarde, em 1958, esta lacuna seria preenchida pelo modelo proposto por Frank Rosenblatt, denominado perceptron, que era composto por uma estrutura de rede, tendo como unidades básicas nodos MCP, e por um algoritmo de treinamento.

A Figura 2.4 demonstra a topologia do perceptron, para uma única saída.



**Figura 2.4 – Topologia de um perceptron simples**

O modelo neural anteriormente descrito também inclui o termo limiar, denotado por  $b_k$ . Este termo possui a função de causar um decréscimo na entrada da função transferência,  $v_k$ .

Desta forma, pra o neurônio  $k$ , temos:

$$v_k = \sum_{j=1}^N w_{jk} x_j + b_k \quad (2.09)$$

$$y_k = \varphi(v_k) \quad (2.10)$$

Onde:

$x_1, x_2, \dots, x_N$  = Sinais de entrada.

$w_{1k}, w_{2k}, \dots, w_{Nk}$  = Pesos sinápticos do neurônio  $k$ .

$b_k$  = Termo Limiar.

$\varphi(v_k)$  = Função de ativação.

$v_k$  = Entrada da função de ativação.

O termo limiar,  $b_k$ , é um parâmetro externo ao neurônio  $k$ . Sendo assim, este termo pode ser incluído no somatório da Equação 2.09, na forma de uma entrada adicional, caracterizada por  $x_0 = +1$ , além do peso sináptico a ela associado  $w_{k0} = b_k$ . Assim, a equação 2.09 pode ser escrita da seguinte forma:

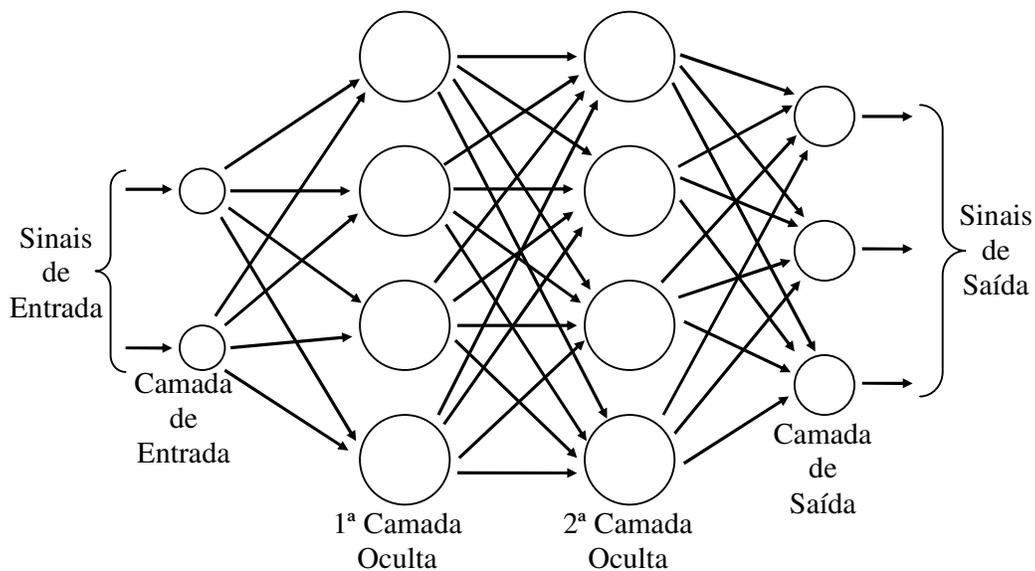
$$v_k = \sum_{j=0}^N w_{jk} x_j \quad (2.11)$$

A função de ativação, denotada por  $\varphi$ , é responsável por fornecer a saída do neurônio. Para o perceptron, a função de ativação utilizada, é a função degrau (Equação 2.12).

$$\varphi(v_k) = \begin{cases} y_k = 1 & \text{se } v_k \geq 0 \\ y_k = 0 & \text{se } v_k < 0 \end{cases} \quad (2.12)$$

### 2.3.3 MULTICAMADAS DE NEURÔNIOS ARTIFICIAIS

Para efetuar a modelagem de sistemas físicos faz-se necessário uma rede neural composta por multicamadas, semelhante à arquitetura demonstrada na Figura 2.5.



**Figura 2.5 – Arquitetura de uma rede neural artificial**

A arquitetura apresentada na Figura 2.5 é um exemplo de rede neural *feedforward* completamente conectada, ou seja, cada neurônio em uma determinada camada está conectado aos demais da camada seguinte.

Através da rede passam dois tipos de sinais, sendo o primeiro deles chamado de sinal função. Este sinal é recebido pelos nódulos da camada de entrada na forma de estímulos gerados pelo ambiente externo e propaga-se ao longo da rede, sendo processado em cada neurônio intermediário na forma de uma função das entradas, ponderadas pelos pesos à elas associados, até que emergindo dos nódulos da camada de saída, produz a resposta da rede. Já o segundo sinal, é constituído de uma função erro, que se propaga no sentido oposto à rede, sendo útil para o algoritmo de aprendizagem *Backpropagation*.

### **2.3.3.1 ALGORITMO DA RETROPROPAGAÇÃO (OU *BACKPROPAGATION*)**

O algoritmo da retropropagação, também conhecido como regra delta generalizada foi proposto por RUMELHART e MCCLELLAND em 1986. Nele, o aprendizado é efetuado com base no erro incluso na resposta fornecida pela rede, o que significa que a diferença entre a resposta atual da simulação e a desejada deve ser minimizada, sendo

necessário, portanto, o ajuste dos pesos sinápticos associados às entradas dos neurônios. Assim, o erro calculado na camada de saída é retropropagado às camadas anteriores, servindo de base para a modificação de todos os pesos, até que a resposta obtida pela rede seja satisfatória.

Matematicamente, o erro na iteração  $n$ , correspondente à análise do  $n$ -ésimo par de exemplos (dados de entrada – saída) apresentados à rede, é a diferença entre a resposta atual,  $y_j(n)$ , e a desejada,  $d_j(n)$ , conforme Equação 2.13.

$$e_j(n) = d_j(n) - y_j(n) \quad (2.13)$$

O erro calculado em relação à todos os  $M$  nódulos de saída pode ser obtido da seguinte maneira (Equação 2.14).

$$E(n) = \frac{1}{2} \sum_{j=1}^M e_j^2(n) \quad (2.14)$$

A função objetivo, que se deseja minimizar é dada pela Equação 2.15.

$$\bar{E} = \frac{1}{N} \sum_{n=1}^N E(n) \quad (2.15)$$

Onde  $N$  representa o número total de pares apresentados.

Admitindo-se que o neurônio  $j$  seja alimentado pelos sinais função  $y_i(n)$  emitidos pelos nódulos  $i$  da camada anterior, sendo cada sinal associado à um peso sináptico  $w_{ij}(n)$ , o nível de atividade do nódulo  $j$  pode ser expresso pela Equação 2.16.

$$v_j(n) = \sum_{i=0}^I w_{ij}(n) y_i(n) \quad (2.16)$$

Onde  $I$  é o número total de nódulos que enviam sinais função ao nódulo  $j$ , e  $w_{0j}(n)$  corresponde ao limiar de  $j$ .

Desta forma, a resposta produzida pela rede é apresentada na Equação 2.17.

$$y_j(n) = \varphi(v_j(n)) \quad (2.17)$$

Onde  $\varphi$  é uma função contínua e diferenciável, como por exemplo uma função sigmoideal.

O algoritmo de retropropagação baseia-se no ajuste dos pesos sinápticos proporcionalmente à diminuição do erro,  $E(n)$ . Assim, através da regra da cadeia pode-se escrever a Equação 2.18.

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = \left( \frac{\partial E(n)}{\partial e_j(n)} \right) \left( \frac{\partial e_j(n)}{\partial y_j(n)} \right) \left( \frac{\partial y_j(n)}{\partial v_j(n)} \right) \left( \frac{\partial v_j(n)}{\partial w_{ij}(n)} \right) \quad (2.18)$$

Diferenciando a Equação 2.14 em relação à  $e_j(n)$ , têm-se:

$$\left( \frac{\partial E(n)}{\partial e_j(n)} \right) = e_j(n) \quad (2.19)$$

Diferenciando a Equação 2.13 com relação à  $y_j(n)$ , têm-se:

$$\left( \frac{\partial e_j(n)}{\partial y_j(n)} \right) = -1 \quad (2.20)$$

O terceiro termo da Equação 2.19 é obtido por meio da diferenciação da Equação 2.17 em relação à  $v_j(n)$ .

$$\left( \frac{\partial y_j(n)}{\partial v_j(n)} \right) = \varphi'(v_j(n)) \quad (2.21)$$

Finalmente, diferenciando a Equação 2.16 em relação à  $w_{ij}(n)$ , têm-se:

$$\left( \frac{\partial v_j(n)}{\partial w_{ij}(n)} \right) = y_i(n) \quad (2.22)$$

Substituindo-se as Equações 2.19, 2.20, 2.21 e 2.22 na Equação 2.18, obtém-se:

$$\left( \frac{\partial E_j(n)}{\partial w_{ij}(n)} \right) = -e_j(n) \varphi'(v_j(n)) y_i(n) \quad (2.23)$$

A correção dos pesos sinápticos,  $\Delta w_{ij}(n)$  é obtida por meio da regra delta.

$$\Delta w_{ij}(n) = -\eta \left( \frac{\partial E_j(n)}{\partial w_{ij}(n)} \right) \quad (2.24)$$

Onde  $\eta$  é o parâmetro da taxa de aprendizagem, e o uso do sinal negativo deve-se ao fato do aprendizado ser efetuado no sentido do decréscimo do gradiente do erro.

Substituindo-se a Equação 2.23 na Equação 2.24, têm-se:

$$\Delta w_{ij}(n) = \eta \delta_i(n) y_j(n) \quad (2.25)$$

Onde:

$$\delta_i(n) = -\frac{\partial E(n)}{\partial v_j(n)} = e_j(n) \varphi_j'(v_j(n)) \quad (2.26)$$

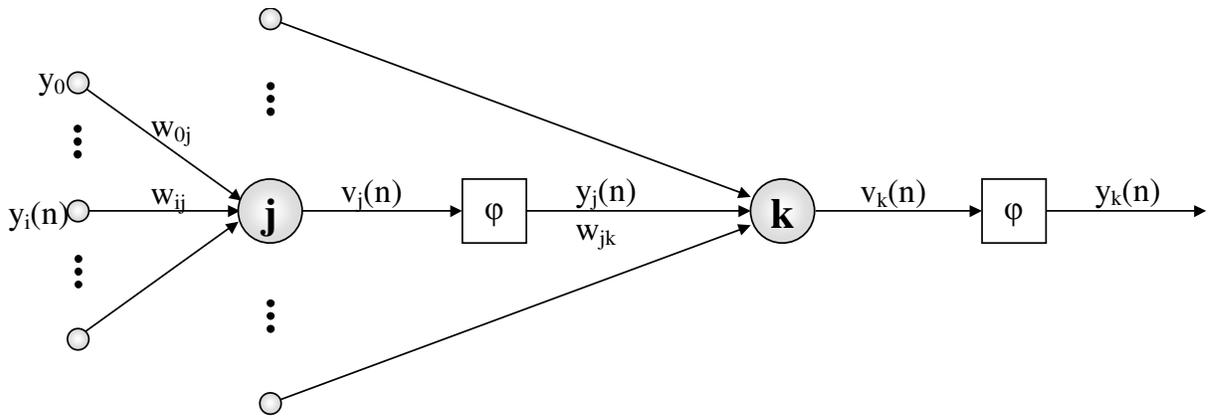
Observando-se a Equação 2.25 pode-se concluir que a dificuldade de se definir o ajuste dos pesos está concentrada no cálculo do erro,  $e_j(n)$ .

Se  $j$  é um nódulo de saída, então o cálculo do erro é extremamente simples (Equação 2.06), pois a resposta desejada é conhecida, o que não acontece quando o neurônio  $j$  está localizado em uma camada oculta. Neste caso, o erro deve ser determinado a partir de sinais provenientes dos neurônios da camada seguinte que estão conectados ao nódulo  $j$ .

Aplicando a regra da cadeia na Equação 2.26, têm-se:

$$\delta_i(n) = -\left(\frac{\partial E(n)}{\partial y_j(n)}\right)\left(\frac{\partial y_j(n)}{\partial v_j(n)}\right) = -\left(\frac{\partial E(n)}{\partial y_j(n)}\right)\varphi_j'(v_j(n)) \quad (2.27)$$

Considerando um neurônio  $k$ , integrante da camada de saída, conforme demonstrado na Figura 2.6, pode-se proceder o cálculo do erro segundo a Equação 2.28.



**Figura 2.6 – Detalhe da conexão entre os neurônios “j” e “k”**

$$E(n) = \frac{1}{2} \sum_k e_k^2(n) \quad (2.28)$$

Diferenciando a Equação 2.28 em relação à  $y_j(n)$ , têm-se:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k \left( \frac{\partial e_k(n)}{\partial y_j(n)} \right) \quad (2.29)$$

Aplicando a regra da cadeia à Equação 2.29, têm-se:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k(n) \left( \frac{\partial e_k(n)}{\partial v_j(n)} \right) \left( \frac{\partial v_j(n)}{\partial y_j(n)} \right) \quad (2.30)$$

Como o erro na última camada pode ser calculado com base na resposta desejada, para o nóculo k, têm-se:

$$e_k(n) = d_k(n) - y_k(n) = d_k(n) - \varphi_k(v_k(n)) \quad (2.31)$$

Diferenciando a Equação 2.31 em relação à  $v_k(n)$ , têm-se:

$$\left( \frac{\partial e_k(n)}{\partial v_j(n)} \right) = -\varphi_k'(v_k(n)) \quad (2.32)$$

O nível de atividade para o nódulo  $k$  é dado pela Equação 2.33.

$$v_k(n) = \sum_{j=0}^m w_{jk}(n) y_j(n) \quad (2.33)$$

Diferenciando a Equação 2.33 em relação à  $y_j(n)$  têm-se:

$$\left( \frac{\partial v_k(n)}{\partial y_j(n)} \right) = w_{jk}(n) \quad (2.34)$$

Substituindo-se as Equações 2.32 e 2.34 na Equação 2.29, obtém-se a relação desejada.

$$\frac{\partial E(n)}{\partial y_j(n)} = -\sum_k e_k(n) \varphi_k'(v_k(n)) w_{jk}(n) = -\sum_k \delta_k(n) w_{jk}(n) \quad (2.35)$$

Finalmente substituindo-se a Equação 2.35 na Equação 2.27, obtém-se a fórmula da retropropagação para  $\delta_j(n)$ , no caso em que  $j$  é um nódulo oculto.

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{jk}(n) \quad (2.36)$$

Neste momento já é possível determinar o ajuste dos pesos sinápticos vinculados as conexões com o neurônio  $j$  (Equação 2.25), seja ele integrante de uma camada intermediária ou da camada de saída.

A taxa de aprendizagem ( $\eta$ ), presente na Equação 2.25, constitui um importante parâmetro, pois estabelece uma relação entre a velocidade de convergência e a probabilidade de instabilidade no processo de aprendizagem. Caso  $\eta$  assumia valores muito pequenos, as mudanças nos pesos sinápticos também serão muito suaves, acarretando assim, um aumento do número de iterações requeridas pelo treinamento. Entretanto, se  $\eta$  for muito grande, as mudanças serão bruscas, provocando instabilidades no processo de aprendizagem.

Para facilitar a obtenção do equilíbrio entre a velocidade de convergência e a estabilidade do treinamento, RUMELHART e MCCLELLAND (1986) propôs a inclusão de mais um termo à Equação 2.25.

$$\Delta w_{ij}(n) = \eta \delta_j(n) y_j(n) + \alpha \Delta w_{ij}(n-1) \quad (2.37)$$

Onde:

$\alpha$  = Termo de momentum.

### 2.3.3.2 ALGORITMO DE LEVENBERG-MARQUARDT COM REGULARIZAÇÃO BAYESIANA

Segundo DEMUTH e BEALE (2002), pelo fato do algoritmo *backpropagation* apresentar um processo de convergência lento, foram desenvolvidos métodos alternativos que convergem a uma velocidade maior que aquele. Dentre estes métodos de otimização para treinamento de redes neurais, se destaca o algoritmo de Levenberg-Marquardt.

De acordo com KERNANI e colaboradores (2005), este algoritmo representa a “ponte” entre o método Gauss-Newton e o de gradiente descendente. O primeiro método possui uma velocidade de convergência muito alta, devida a suas propriedades quadráticas. No entanto, há uma dependência elevada dos valores iniciais dos pesos e *bias*, cujas estimativas podem ser muito difíceis em uma aplicação em um sistema real. Já o segundo método citado, apresenta uma maior precisão quando a região próxima ao mínimo da função objetivo é alcançada. Entretanto, sua velocidade de convergência é baixa, pois minimiza a função objetivo de maneira linear. Combinando os atributos positivos dos dois métodos, o algoritmo de Levenberg-Marquardt faz uso de uma técnica de otimização híbrida, a qual pode ser aplicada no tratamento de muitos problemas reais.

A equação de iteração para este método pode ser escrita da seguinte forma:

$$X_{k+1} = X_k - [J^T J + \mu I]^{-1} J^T e \quad (2.38)$$

Onde:

$J$  = Matriz Jacobiana, a qual contém as derivadas primeiras dos erros da rede com relação aos pesos e *bias*.

$H = J^T J$  = Aproximação da matriz de Hessian, a qual contém as derivadas segundas dos erros da rede com relação aos pesos e *bias*.

O valor do escalar  $\mu$  define qual método de otimização prevalece (Gauss-Newton ou gradiente descendente). Quando  $\mu$  assume um valor elevado o método utilizado é o gradiente descendente. Por outro lado, ao diminuir  $\mu$ , a técnica empregada aproxima-se do método de Gauss-Newton. Assim, a manipulação de  $\mu$  visa usufruir das vantagens dos dois métodos, minimizando o efeito das suas desvantagens. Quando um passo resultaria em um acréscimo da função objetivo,  $\mu$  é multiplicado por uma constante  $\mu' > 1$ , para direcionar o algoritmo ao método do gradiente descendente, e assim, obter maior estabilidade. Em contrapartida, quando um passo resultaria em um decréscimo na função objetivo,  $\mu$  é multiplicado por  $1/\mu'$ , para direcionar o algoritmo ao método de Gauss-Newton, aumentando assim, a velocidade de convergência (KERNANI, 2005).

Além da minimização da função objetivo, o processo de treinamento tem como meta, conferir à rede a capacidade de generalização. Entretanto, o fato de apresentar um erro muito pequeno na etapa de treinamento não implica em uma boa generalização posterior, sendo este problema conhecido como sobre-ajuste. A rede memorizou os exemplos de treinamento, mas não aprendeu generalizar em situações novas (DEMUTH e BEALE, 2002).

Sendo assim, no caso de sobre-ajuste, a rede pode estar sobreparametrizada, ou seja, a rede possui um número excessivo de parâmetros (pesos), necessitando ser redimensionada. O objetivo do treinamento deve ser então o de encontrar o número de parâmetros ideal. No entanto, estimar o número de parâmetros é uma tarefa árdua que requer conhecimento sobre a complexidade do problema. Este conhecimento normalmente não está disponível, principalmente tratando-se de problemas multidimensionais, pois muitas vezes é este conhecimento que se deseja obter por meio do processo de modelagem (BRAGA, 2000).

Segundo DEMUTH e BEALE (2002) uma opção para melhorar a generalização, e evitar o sobre-ajuste, é a modificação da função objetivo. Neste sentido, o algoritmo de Regularização Bayesiana prevê a substituição da média dos erros quadrados da rede (expressão normalmente utilizada como função objetivo) pela Equação 2.39.

$$FO = \alpha' SSE + \beta' SSW \quad (2.39)$$

Onde:

FO = Função objetivo.

SSE = Soma dos quadrados dos erros.

SSW = Soma dos quadrados dos pesos.

$\alpha'$  e  $\beta'$  = Parâmetros da função objetivo.

Os parâmetros  $\alpha'$  e  $\beta'$  são obtidos através da utilização de técnicas estatísticas, com a estrutura de *Bayesian*. Como a técnica de Regularização Bayesiana requer o cálculo da matriz de Hessian, o algoritmo de treinamento associado à regularização é o de Levenberg-Marquardt (HAGAN e FORESSE, 1997).

Uma das vantagens do algoritmo de Levenberg-Marquardt com Regularização Bayesiana é o fornecimento do valor de  $\gamma$ , parâmetros da rede que estão efetivamente sendo utilizados. Esta medida pode ser empregada no dimensionamento da rede, pois de acordo com HAGAN e FORESSE (1997), caso o valor de  $\gamma$  seja muito próximo do número total de parâmetros ( $N'$ ), então se deve aumentar o tamanho da rede (aumentando o número de neurônios na camada intermediária), pois a rede pode não ser grande o suficiente para tratar o problema em questão. Este procedimento deve ser repetido até que  $\gamma$  se torne constante, não importando quão grande seja  $N'$ .

## 2.4 CONTROLADOR PROPORCIONAL INTEGRAL (PID)

A lei de controle utilizada para o PID digital pode ser escrita da seguinte forma:

$$L_0(kTa) = L_0(kTa - Ta) + K_c e(kTa) + \frac{Ta}{\tau_I} \sum_{i=0}^k e(iTa) + \tau_D \left( \frac{e(iTa) - e(iTa - 1)}{Ta} \right) \quad (2.40)$$

A Equação 2.33 é conhecida como forma posição, pois a cada instante de amostragem, a lei de controle fornece diretamente, o valor atual (posição) da saída do controlador (OGUNNAIKE e RAY, 1994).

A utilização do controle digital deve ser precedida por uma análise de estabilidade. No entanto, dependendo da complexidade do processo, esta análise pode se tornar muito difícil de ser concretizada. Aproximações para processos mais simples, como os de primeira e segunda ordem, podem ser realizadas, entretanto, perde-se na precisão dos resultados. Sendo assim, simulações do processo, sob a ação do controle digital, podem ser efetuadas para que se avalie o comportamento deste.

## **2.5 PROPOSTA DA DISSERTAÇÃO**

Com base no levantamento bibliográfico realizado, foi possível perceber que o controle de uma coluna de absorção de pratos tem sido alvo de estudos de alguns pesquisadores. No entanto, as técnicas empregadas até o momento não possuem a capacidade de processamento demonstrada pelas redes neurais artificiais (RNA's), as quais já foram aplicadas na modelagem desta operação unitária. Sendo assim, o presente trabalho propõe o desenvolvimento e a utilização de um controlador *feedforward* baseado em modelo inverso de redes neurais para manipulação da vazão de solvente de uma coluna de absorção empregada em processo de fermentação alcoólica. Comparações serão realizadas em relação à aplicação de algoritmo de controle DMC.

**CAPÍTULO 3**  
**CONTROLE DA COLUNA DE ABSORÇÃO**  
**DE PRATOS**

### 3.1 INTRODUÇÃO

Neste capítulo são expostos alguns dos resultados obtidos por PALÚ (2001), ao aplicar a estratégia de controle DMC ao sistema estudado. O Controle por Matriz Dinâmica merece tal atenção, pois, no capítulo seguinte propiciará o embasamento para a avaliação do desempenho da nova estratégia de controle proposta.

Finalmente, a estratégia de controle via redes neurais artificiais é apresentada, assim como o procedimento a ser aplicado para sua implementação.

### 3.2 CONTROLE POR MATRIZ DINÂMICA

A seguir são apresentados alguns dos resultados obtidos por PALÚ (2001) ao aplicar a estratégia de controle DMC ao sistema estudado. Neste trabalho, dentre outros tópicos desenvolvidos, efetuou-se a sintonia dos parâmetros do controlador DMC objetivando a melhora do desempenho do mesmo. Desta forma, partindo dos valores iniciais do horizonte de controle,  $U=2$ , horizonte de predição,  $V=4$ , e horizonte de processo,  $T=10$ , além do fator de supressão,  $f=0$ , o autor procurou estudar a influência de tais parâmetros na resposta do processo. Depois de definidos os melhores valores para estes ( $f=0$ ,  $V=4$ ,  $U=2$  e  $T=10$ ), os quais agregam um desempenho satisfatório a um esforço computacional razoável, foi efetuada uma comparação entre este controlador e um controlador clássico, PI (Proporcional Integral). A Figura 3.1 demonstra a superioridade do DMC em relação ao PI, pois além da resposta apresentar uma menor sobrelevação, o estado estacionário é obtido mais rapidamente. Já na Figura 3.2 é exposto o comportamento da variável manipulada sob a ação do DMC.

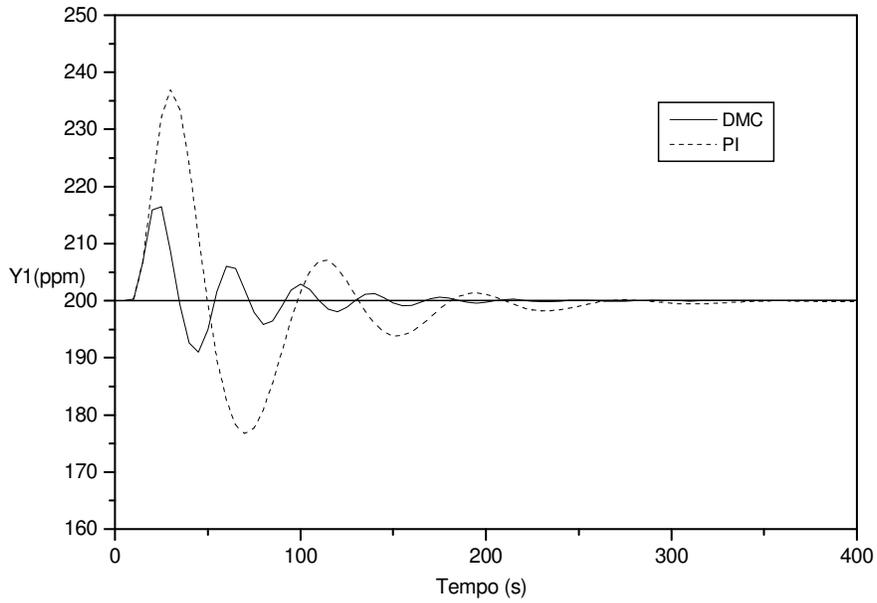


Figura 3.1 – Comparação entre o desempenho dos controladores DMC e PI (problema SISO, perturbação única) (PALÚ, 2001).

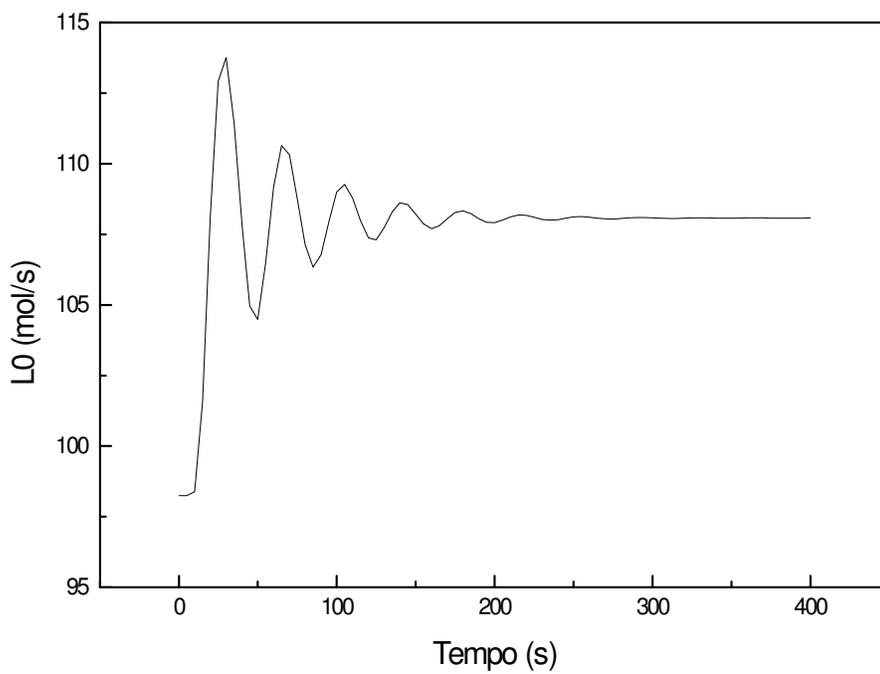


Figura 3.2 – Comportamento da variável manipulada sob a ação do controlador DMC (problema SISO, perturbação única) (PALÚ, 2001).

### 3.3 CONTROLE VIA REDES NEURAIS

#### 3.3.1 CONSTRUÇÃO DA REDE NEURAL

A nova estratégia de controle para a coluna de absorção de pratos estudada, desenvolvida neste trabalho, é a utilização de um controlador *feedforward* baseado em modelo inverso de redes neurais. O modelo inverso difere do direto na variável predita pela rede, sendo para o primeiro caso, a variável manipulada, e para o segundo, a variável controlada. Na Figura 3.3 é apresentado um esquema dos dois modelos, explicitando-se as variáveis de entrada e saída para o sistema estudado. Onde  $L_{0k}$  é a vazão de solvente no instante atual,  $Y_{1k}$  representa a fração molar do álcool no gás de saída da coluna no instante atual, e  $Y_{1k-1}$  no instante anterior. Além disso, fazem parte do conjunto de variáveis utilizadas pelo sistema de controle, a fração molar do álcool desejada (set point) no gás de saída da coluna no instante posterior ( $Y_{1spk+1}$ ), a fração molar do álcool no gás de entrada da coluna no instante atual ( $Y_{N+1k}$ ), e a vazão de solvente no instante posterior ( $L_{0k+1}$ ).

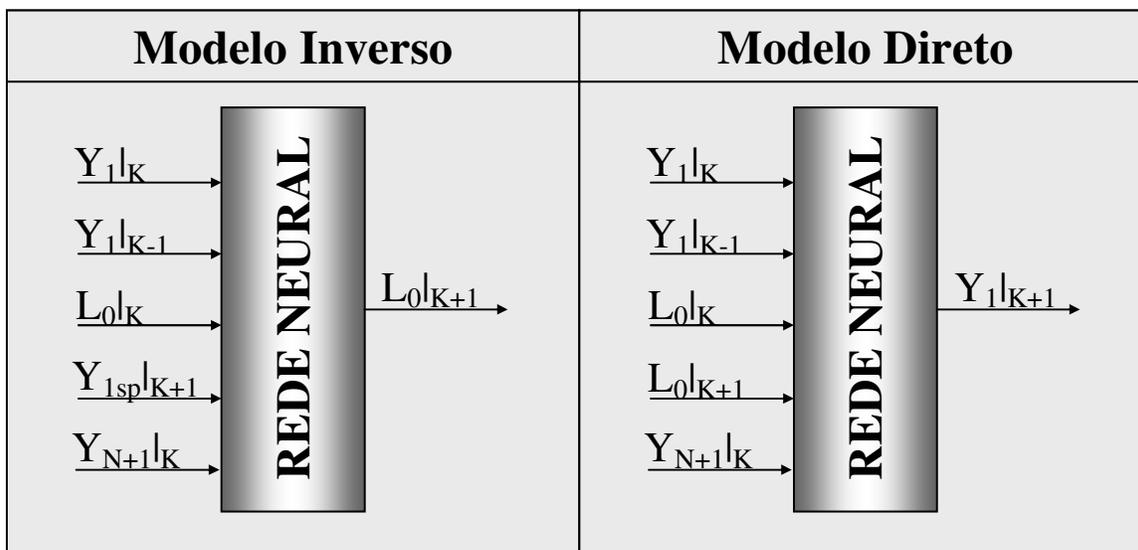


Figura 3.3 – Esquema dos modelos de rede neural

A utilização do modelo direto implica na necessidade de um processo de otimização para a obtenção da ação de controle. Desta maneira, a rede neural prevê a variável controlada para um dado valor da variável manipulada. Ao receber o valor predito da variável controlada, o otimizador avalia o erro propiciado por esta ação de controle, e caso este erro seja superior a um valor pré-determinado, a ação é descartada e uma nova é calculada. Este procedimento é repetido até que o erro gerado pela ação de controle seja tão pequeno quanto o desejável. Na Figura 3.4 é apresentado um diagrama representativo do controle através do modelo direto de redes neurais.

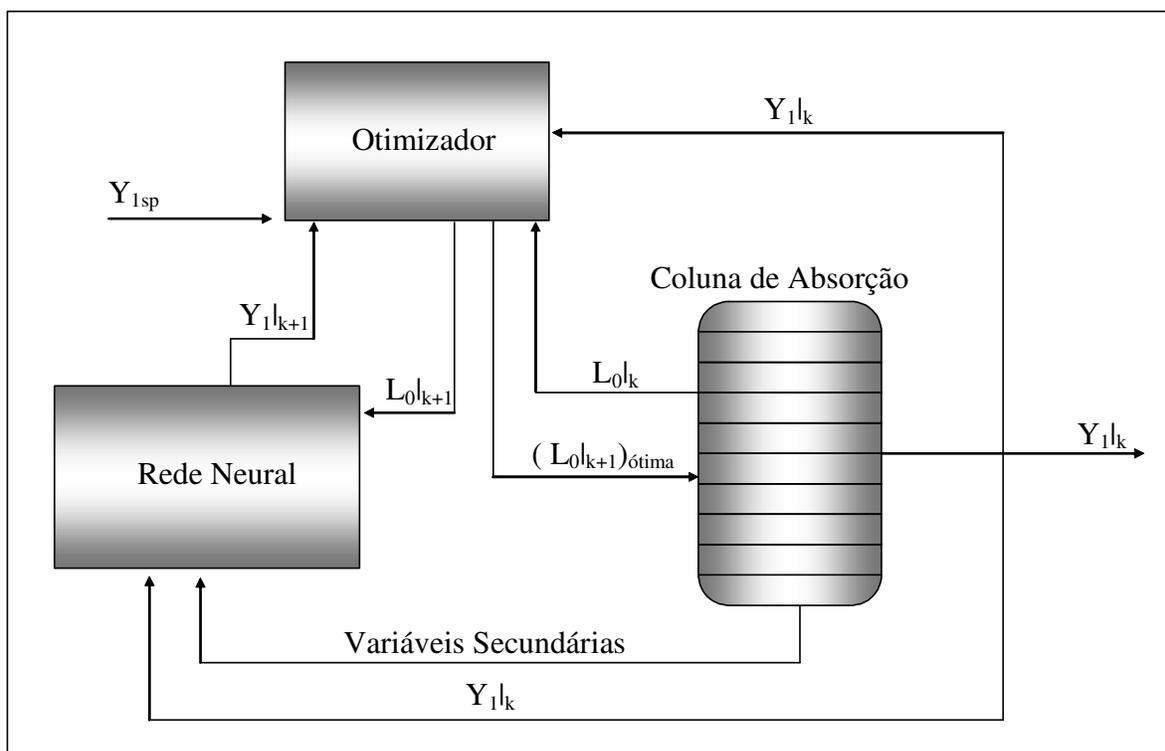


Figura 3.4 – Diagrama do controle da coluna de absorção de pratos através do modelo direto de redes neurais (Controle Preditivo com Modelo)

De acordo com o que foi apresentado, o modelo inverso, ao contrário do direto, não necessita de um processo de otimização para definir a ação de controle, o que lhe confere a vantagem de exigir menor esforço computacional. Deste modo, a ação de controle

é determinada pela própria rede neural, conferindo assim, uma maior agilidade ao sistema de controle, fator importante na implementação *on line* do controlador.

Na Figura 3.5 é demonstrado um fluxograma que ilustra a estratégia de controle através do modelo inverso de redes neurais. Este controlador é considerado *feedforward*, pois a ação de controle independe da avaliação do erro da variável controlada.

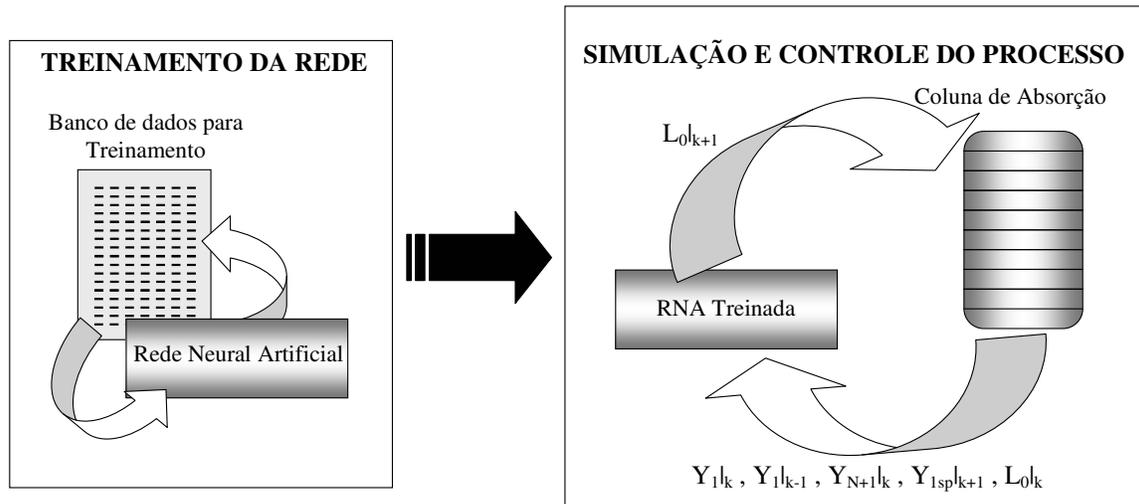


Figura 3.5 – Fluxograma representativo do controle da coluna de absorção de pratos através do modelo inverso de redes neurais (Controle *Feedforward*)

A determinação da arquitetura da rede depende de vários fatores, entre eles o número de variáveis de entrada e saída. Conforme demonstrado na Figura 3.3, para o modelo inverso de redes neurais aplicado ao sistema estudado, são utilizadas cinco variáveis de entrada e uma de saída.

Desta forma, a arquitetura da rede pode ser representada por  $5 - mM - 1$ , onde “m” e “M” correspondem ao número de camadas escondidas e ao número de neurônios em cada camada, respectivamente. Sendo que a determinação destes valores é efetuada de modo a propiciar uma rede que forneça bons resultados sem que haja um esforço computacional excessivo.

### 3.3.2 TREINAMENTO DA REDE NEURAL

Para que a rede neural forneça resultados satisfatórios, além de uma arquitetura adequada, o processo de treinamento deve ser efetuado com qualidade. Para tanto deve-se fornecer à rede, dados representativos da dinâmica do processo.

Neste trabalho, os dados destinados ao treinamento e validação da rede neural foram obtidos do programa computacional em linguagem Fortran, desenvolvido por PALÚ (2001), e gentilmente cedido. Deste modo, uma perturbação degrau era inserida na composição do gás à entrada da coluna e o Controle por Matriz Dinâmica implementado, fazendo com que a variável controlada (composição do gás à saída da coluna) retornasse ao valor desejado. Assim, o valor da variável manipulada a ser implementado na coluna de absorção era determinado pelo controlador DMC. Este procedimento foi repetido várias vezes de modo a realizar a cobertura da faixa de variação da composição do gás à entrada da coluna que se propunha trabalhar (de -10% até +10%), sem que houvesse a tomada de um número excessivo de dados (ocasionando assim, um esforço computacional desnecessário), ou ainda, a qualidade do treinamento fosse comprometida pela ausência de dados.

Neste ponto é necessário efetuar um esclarecimento quanto à necessidade do controlador DMC na obtenção do banco de dados para o treinamento da rede neural. Os dados destinados a este fim devem conter informações sobre a relação entre a variável manipulada (neste caso, a vazão de solvente) e a variável controlada. Esta relação é dada pela dinâmica do processo (ou pelas equações de modelagem fenomenológica), ou seja, o modo pelo qual o valor da variável manipulada é obtido, não apresenta relevância sobre a resposta fornecida pelo processo. Deste modo, este valor pode ser determinado tanto em malha aberta, quanto em malha fechada, de forma manual ou com o auxílio de um controlador qualquer, como o PID (Proporcional Integral Derivativo), ou ainda, o DMC.

O método de treinamento utilizado é o da retropropagação através de otimização de Levenberg-Marquardt com regularização Bayesiana. O algoritmo da retropropagação, também conhecido como regra delta generalizada foi proposto por Rumelhart, McClelland e Williams em 1986. Nele, o aprendizado é efetuado com base no erro incluso na resposta fornecida pela rede, o que significa que a diferença entre a resposta atual da simulação e a

desejada deve ser minimizada, sendo necessário, portanto, o ajuste dos pesos sinápticos associados às entradas dos neurônios. Assim, o erro calculado na camada de saída é retropropagado às demais camadas, servindo de base para a modificação de todos os pesos, até que a resposta obtida pela rede seja satisfatória.

Como apresentado no Capítulo 2, uma das vantagens do algoritmo de Levenberg-Marquardt com Regularização Bayesiana é o fornecimento do valor de  $\gamma$  (parâmetros da rede que estão efetivamente sendo utilizados). De acordo com o critério de HAGAN e FORESSE (1997), esta medida pode ser empregada no dimensionamento da rede. Caso o valor de  $\gamma$  seja muito próximo do número total de parâmetros ( $N$ ), então se deve aumentar o tamanho da rede (aumentando o número de neurônios na camada intermediária), pois a rede pode não ser grande o suficiente para tratar o problema em questão. Este procedimento deve ser repetido até que  $\gamma$  se torne constante, não importando quão grande seja  $N$ .

### **3.3.3 IMPLEMENTAÇÃO DO DISPOSITIVO DE SEGURANÇA**

O sucesso da estratégia de controle proposta depende da eficiência proporcionada pela rede neural artificial empregada. Como mencionado anteriormente, na etapa de treinamento, a rede extrai informações sobre o ambiente no qual ela deve operar. Entretanto, caso a rede seja exposta a um ambiente de trabalho diferente daquele para o qual ela foi treinada, a probabilidade da resposta fornecida ser diferente da desejada aumenta.

Considerando que as condições de operação impostas à coluna de absorção estudada são passíveis de mudanças, torna-se impossível treinar a rede para que esta seja capaz de manter sua precisão frente a qualquer distúrbio que possa surgir.

Sendo assim, a estratégia de controle proposta deve ser implementada em paralelo a um dispositivo de segurança, o qual deve ser requisitado sempre que o desempenho do controlador *feedforward* via redes neurais artificiais decaia a um nível crítico. Com tal objetivo, e devida à facilidade de implementação, se propõe a utilização de um controlador convencional *feedback* PI ou PID.

### **3.3.4 DISTORÇÃO E FILTRAGEM DE DADOS DE ENTRADA DA REDE**

Devido às limitações impostas pelos instrumentos de medida, os valores obtidos para a composição do gás à entrada e saída da coluna podem diferir dos reais. Visando enriquecer a simulação quanto à presença de ruído nas medidas de composição, é possível realizar uma aproximação, acrescentando-se ao valor medido da variável de interesse, um determinado escalar. Sendo este obtido através da multiplicação de um número aleatório compreendido entre 0 e 1 por uma função senoidal de amplitude controlada e alta frequência.

Depois de avaliar o comportamento do sistema de controle quando alimentado com dados distorcidos, a realização de um novo teste se torna interessante. Neste, os dados adulterados devem sofrer um processo de filtração. Deste modo, as variações bruscas nos valores da composição do gás são atenuadas, e uma melhora no desempenho do controlador é esperada.

# **CAPÍTULO 4**

## **RESULTADOS**

## 4.1 INTRODUÇÃO

No início deste capítulo são apresentados os resultados preliminares, obtidos através da simulação da coluna de absorção estudada em malha aberta. Através destes testes foi possível definir um instante de amostragem condizente com a necessidade do sistema.

Em seguida, são descritos os testes realizados para a construção dos bancos de dados de treinamento e validação da rede. Além disso, são demonstrados os resultados alcançados com os testes destinados à determinação da arquitetura da rede.

Na seqüência, realizou-se a comparação entre o desempenho da estratégia de controle *feedforward* baseada no modelo inverso de redes neurais e o da técnica de Controle por Matriz Dinâmica (DMC), avaliando-se a integral do quadrado do erro e a quantidade de solvente gasta para efetuar o controle da coluna.

Também neste capítulo, foi implementado o dispositivo de segurança da estratégia de controle, sendo este testado frente a distúrbios não previstos na etapa de treinamento da rede.

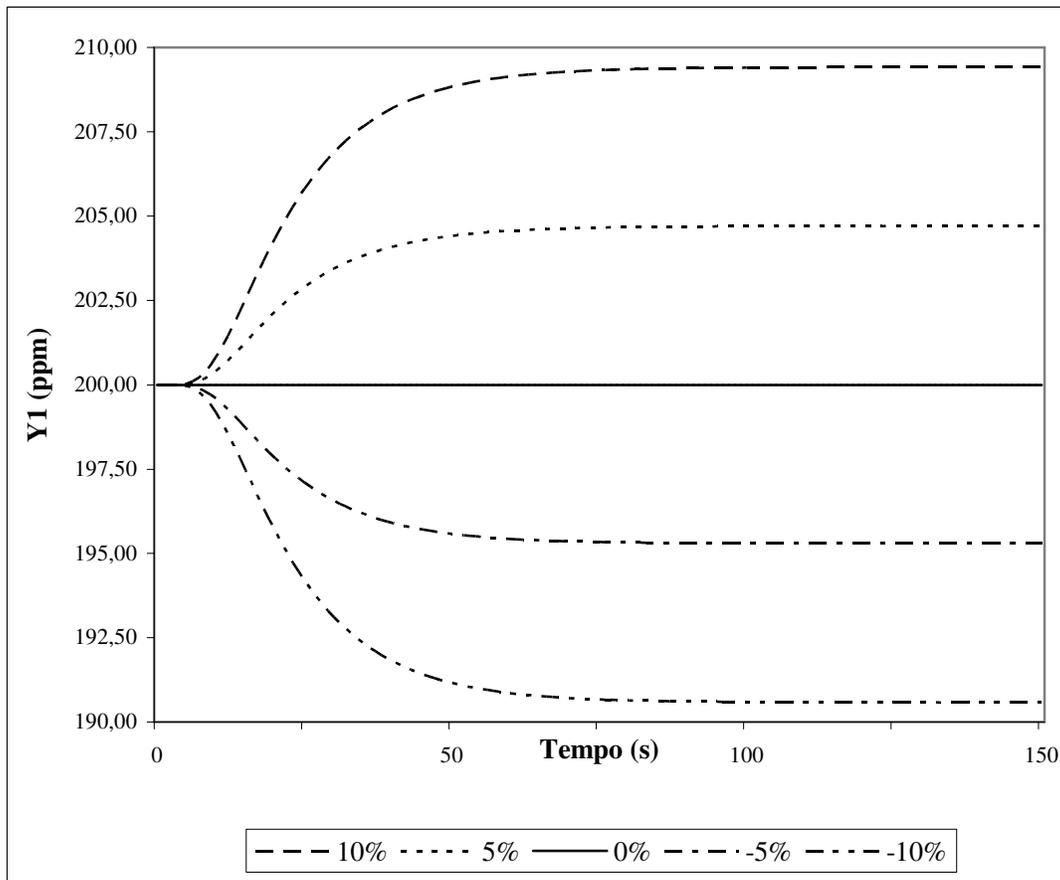
Por fim, visando aproximar os dados destinados à rede, dos dados disponíveis em uma situação real, distorceu-se os mesmos através da inserção de ruído. Posteriormente, implementou-se um filtro capaz de atenuar as variações provocadas pelo ruído. Nas duas situações foi avaliada a reação do sistema de controle.

## 4.2 RESULTADOS PRELIMINARES

De posse do modelo dinâmico da coluna de absorção de pratos, conforme demonstrado no tópico 2.2.1, e seguindo o fluxograma do processo de simulação (Figura 2.2), foi possível avaliar o comportamento do processo em malha aberta.

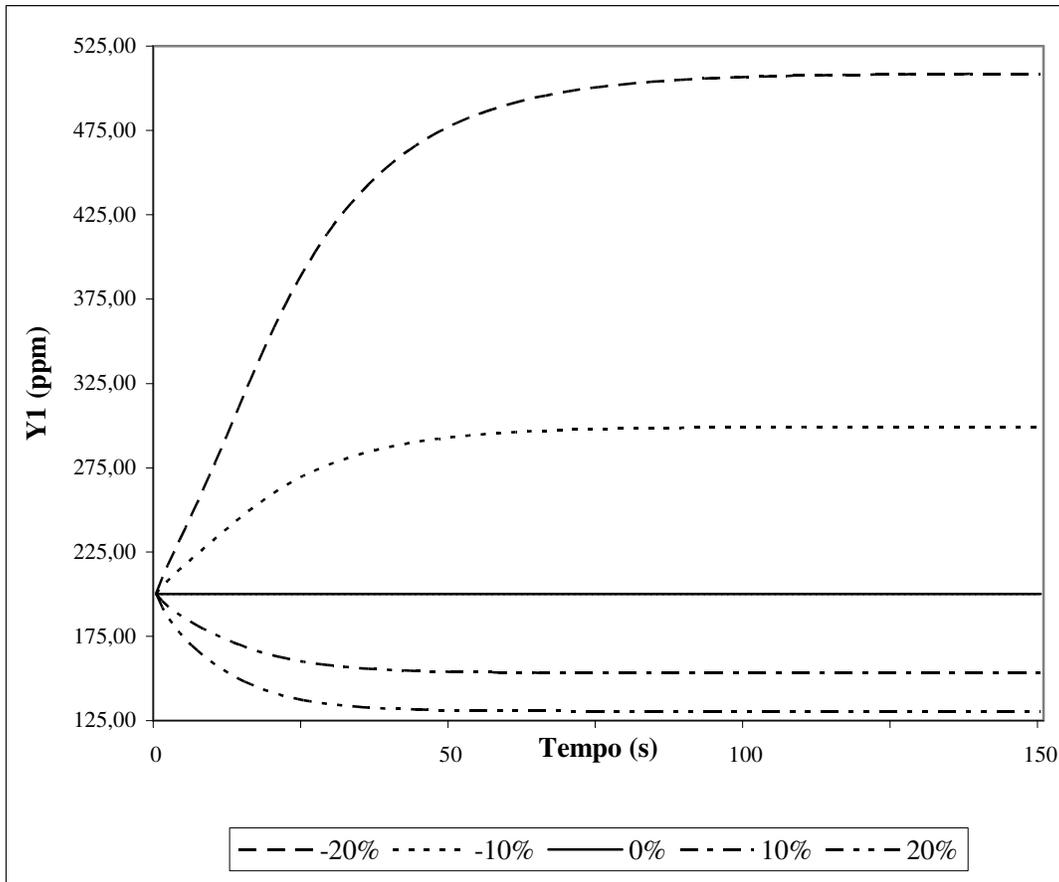
O intuito dos testes realizados foi o de avaliar a não linearidade do sistema, o que justificaria o uso de uma estratégia de controle mais complexa, tal qual a proposta.

O primeiro teste foi caracterizado pela inserção de perturbações degrau, de várias magnitudes, na composição do gás à entrada da coluna ( $Y_{N+1}$ ), procedendo-se na seqüência, o acompanhamento da resposta do sistema. O resultado obtido é demonstrado na Figura 4.1



**Figura 4.1 – Comportamento da variável controlada  $Y_1$  mediante inserção de perturbações de diferentes magnitudes em  $Y_{N+1}$**

Analogamente ao primeiro teste, foram inseridas perturbações degrau na variável manipulada (vazão de solvente,  $L_0$ ) e verificada a influência destes distúrbios na resposta do sistema. O resultado deste teste é apresentado na Figura 4.2.

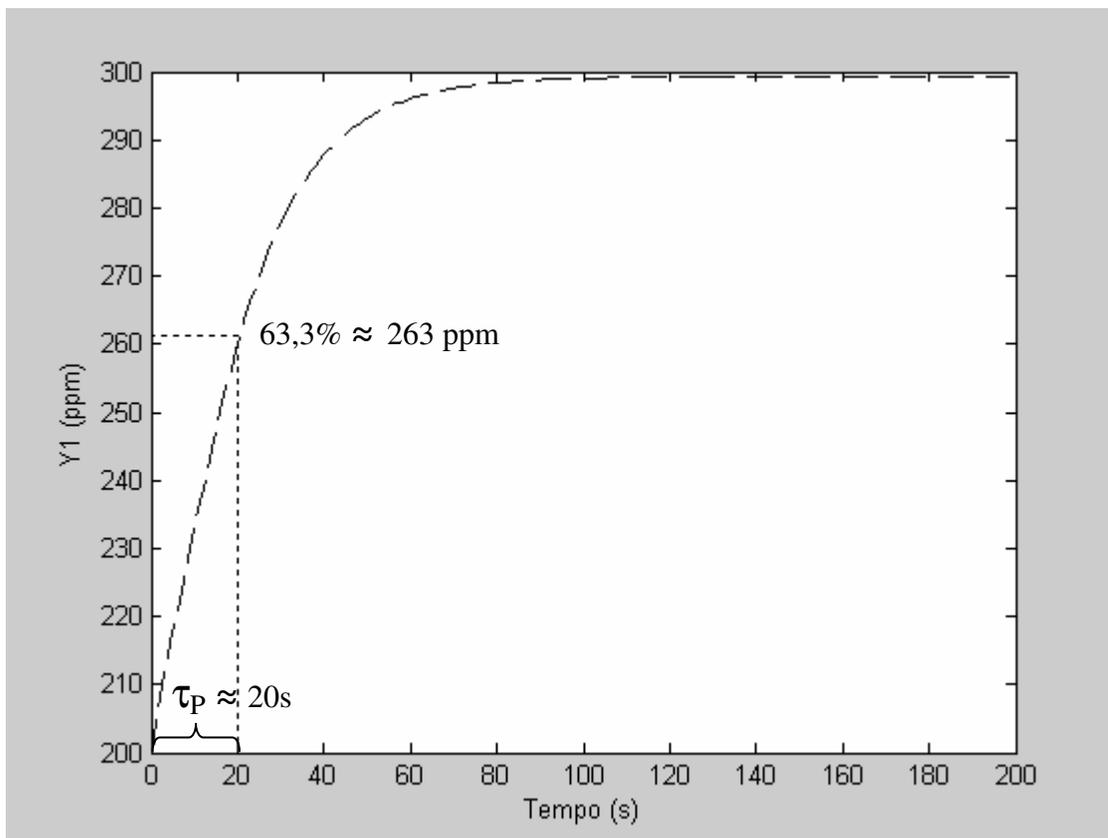


**Figura 4.2 – Comportamento da variável controlada  $Y_1$  mediante inserção de perturbações de diferentes magnitudes em  $L_0$**

A não linearidade do processo pode ser observada principalmente na relação entre a vazão do solvente e a composição do gás à saída da coluna (Figura 4.2).

Além de comprovar a não linearidade do processo, a Figura 4.2 também embasou a determinação do instante de amostragem que seria utilizado. Neste sentido, na Figura 4.3 é apresentada, em detalhe, a resposta do processo frente a uma perturbação de -10% na vazão de solvente (variável manipulada) no instante de tempo inicial.

A escolha do tempo de amostragem (TA) igual a 3 segundos foi realizada com base no critério prático 10 a 20% da constante de tempo do processo. Como o sistema é de primeira ordem, o valor da constante de tempo é obtido no ponto onde a resposta do processo assume 63,3% do valor final da resposta, pela Figura 4.3 pode-se notar que isto ocorre quando  $Y_1$  assume valor igual a 263 ppm, portanto obtém-se  $\tau_p \approx 20s$ .



**Figura 4.3 – Determinação do instante de amostragem (TA)**

### **4.3 TREINAMENTO DA REDE NEURAL ARTIFICIAL**

#### **4.3.1 CONSTRUÇÃO DOS BANCOS DE DADOS DE TREINAMENTO E VALIDAÇÃO**

Como mencionado anteriormente, para efetuar o treinamento da rede neural foram utilizados dados gerados pelo programa computacional em linguagem Fortran, desenvolvido por PALÚ (2001).

O procedimento adotado para a construção do banco de dados de treinamento foi o da inserção de perturbações degrau de diferentes magnitudes na composição do gás à entrada da coluna, e posterior verificação da resposta do processo.

A faixa de distúrbio que se propôs trabalhar foi de -10% à +10% para  $Y_{N+1}$ . Assim, realizou-se a cobertura desta faixa, implementando-se perturbações degrau com incremento de 1%.

Para a construção do banco de dados destinado à validação do treinamento da rede, também se obedeceu a faixa de trabalho utilizada, entretanto, os valores das perturbações foram escolhidos a esmo, incluindo tanto valores presentes no banco de dados de treinamento quanto inéditos, testando a capacidade de generalização da rede.

#### **4.3.2 DETERMINAÇÃO DA ESTRUTURA DA REDE**

Foram testadas várias arquiteturas para a rede neural, no entanto, a primeira camada (entrada) e a última (saída) foram mantidas inalteradas. Sendo assim, na primeira camada foram dispostos cinco neurônios, cada qual trabalhando como um sensor na identificação do valor referente a uma variável de entrada, e analogamente, um nodo constituiu a camada de saída, sendo responsável pelo fornecimento da resposta da rede.

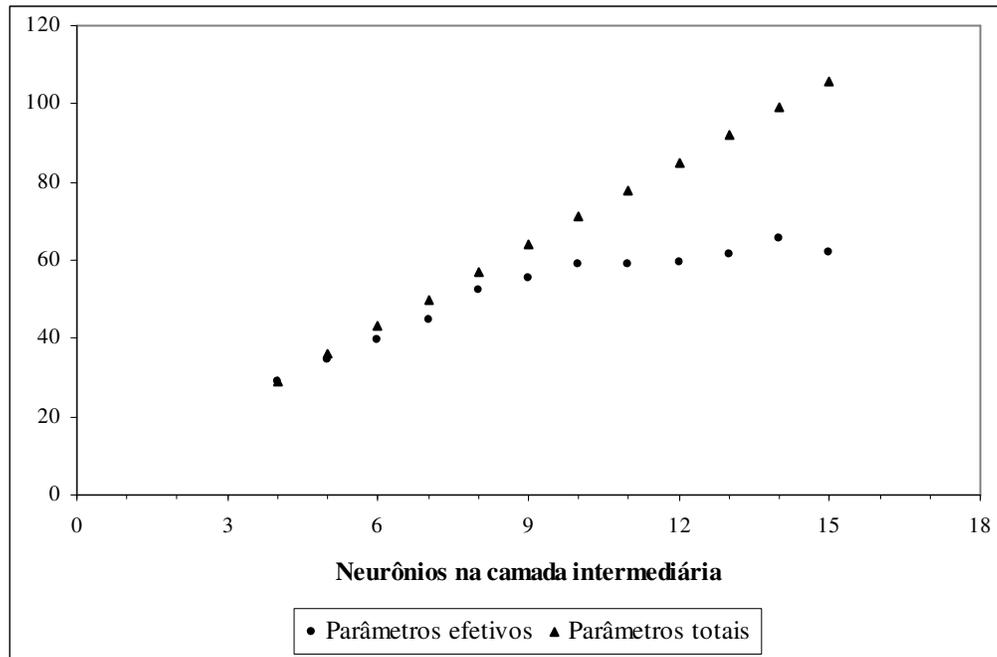
Quanto ao número de camadas intermediárias, bem como a quantidade de nodos em cada uma delas, buscou-se um arranjo capaz de proporcionar um treinamento de qualidade, com capacidade de generalização e reprodutibilidade, sem que houvesse um acréscimo exagerado no esforço computacional. A Tabela 4.1 apresenta os valores da soma dos erros quadrados (SSE) e da média dos erros quadrados (MSE) para os conjuntos de treinamento e validação, obtidos para diferentes estruturas de rede. Nota-se que estruturas superiores a 5 – 5 – 1 são equivalentes quanto aos parâmetros SSE e MSE, na fase de treinamento, assim como no teste feito com o conjunto de dados de validação. Sendo assim, uma arquitetura com duas camadas intermediárias pode ser descartada, pois esta resultaria em um acréscimo de esforço computacional, sem propiciar um aumento de desempenho significativo.

**Tabela 4.1 – SSE e MSE para diferentes estruturas de rede**

Estrutura	Conjunto de Treinamento		Conjunto de Validação	
	SSE (mol <sup>2</sup> /s <sup>2</sup> )	MSE*	SSE (mol <sup>2</sup> /s <sup>2</sup> )	MSE*
5 – 4 – 1	12,080 * 10 <sup>-7</sup>	1,4748 * 10 <sup>-9</sup>	2,0437	4,1879 * 10 <sup>-3</sup>
5 – 5 – 1	23,570 * 10 <sup>-7</sup>	2,8785 * 10 <sup>-9</sup>	2,0489	4,1985 * 10 <sup>-3</sup>
5 – 6 – 1	9,8429 * 10 <sup>-7</sup>	1,2018 * 10 <sup>-9</sup>	2,0515	4,2038 * 10 <sup>-3</sup>
5 – 7 – 1	9,9674 * 10 <sup>-7</sup>	1,2170 * 10 <sup>-9</sup>	2,0650	4,2315 * 10 <sup>-3</sup>
5 – 8 – 1	9,9988 * 10 <sup>-7</sup>	1,2208 * 10 <sup>-9</sup>	2,1112	4,3262 * 10 <sup>-3</sup>
5 – 9 – 1	9,9907 * 10 <sup>-7</sup>	1,2199 * 10 <sup>-9</sup>	2,0425	4,1854 * 10 <sup>-3</sup>
5 – 10 – 1	9,9913 * 10 <sup>-7</sup>	1,2199 * 10 <sup>-9</sup>	2,0850	4,2725 * 10 <sup>-3</sup>
5 – 11 – 1	9,9493 * 10 <sup>-7</sup>	1,2148 * 10 <sup>-9</sup>	2,1511	4,4079 * 10 <sup>-3</sup>
5 – 12 – 1	9,9971 * 10 <sup>-7</sup>	1,2206 * 10 <sup>-9</sup>	2,0380	4,1762 * 10 <sup>-3</sup>
5 – 13 – 1	9,9984 * 10 <sup>-7</sup>	1,2208 * 10 <sup>-9</sup>	2,0632	4,2278 * 10 <sup>-3</sup>
5 – 14 – 1	9,9973 * 10 <sup>-7</sup>	1,2207 * 10 <sup>-9</sup>	2,0862	4,2750 * 10 <sup>-3</sup>
5 – 15 – 1	9,9985 * 10 <sup>-7</sup>	1,2192 * 10 <sup>-9</sup>	2,1559	4,4178 * 10 <sup>-3</sup>
5 – 5 – 5 – 1	9,9898 * 10 <sup>-7</sup>	1,2197 * 10 <sup>-9</sup>	2,0544	4,2098 * 10 <sup>-3</sup>
5 – 6 – 6 – 1	9,9942 * 10 <sup>-7</sup>	1,2203 * 10 <sup>-9</sup>	2,1215	4,3473 * 10 <sup>-3</sup>
5 – 7 – 7 – 1	0,2022 * 10 <sup>-7</sup>	2,4686 * 10 <sup>-9</sup>	2,0963	4,2956 * 10 <sup>-3</sup>

\* mol<sup>2</sup>/([s<sup>2</sup>] . [nº de dados entradas-saída] . [nº de neurônios na camada de saída])

Além dos parâmetros demonstrados na Tabela 4.1, analisou-se o comportamento do número de parâmetros efetivamente utilizados pela rede ( $\gamma$ ) e do número de parâmetros totais ( $N'$ ), em relação ao número de neurônios na camada intermediária. O resultado deste teste é apresentado na Figura 4.4.

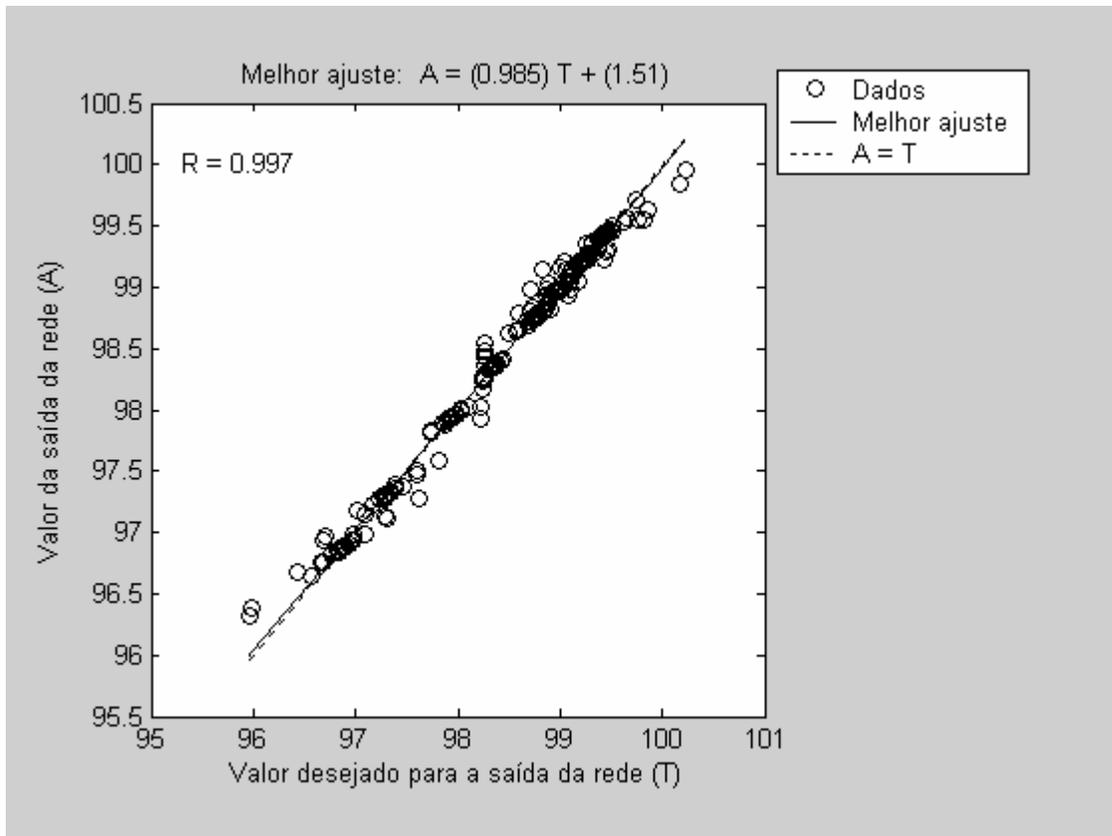


**Figura 4.4 – Número de parâmetros totais e efetivos em relação ao número de neurônios na camada intermediária**

Observando-se a Figura 4.4 pode-se perceber que a partir de 9 neurônios na camada intermediária, o número de parâmetros que são realmente utilizados pela RNA permanece praticamente constante. Sendo assim, seguindo o critério de HAGAN e FORESSE (1997) optou-se por utilizar este número de nodos na camada intermediária.

Portanto, os melhores resultados foram obtidos quando utilizada apenas uma camada intermediária, composta por nove unidades de processamento, sendo a arquitetura 5 – 9 – 1, a que melhor se adequou ao sistema estudado. A Figura 4.5 demonstra a relação obtida entre a resposta fornecida pela rede, de estrutura igual à mencionada, e a resposta desejada, sendo que para este teste, fez-se uso do banco de dados de validação. Nota-se que o desempenho da rede foi satisfatório, proporcionando um índice de correlação de 0,997.

Além disso, o ajuste efetuado para o conjunto de dados testados proporcionou um coeficiente angular de 0,985 (cujo valor ótimo seria 1) e um coeficiente linear de 1,51 (cujo valor ótimo seria 0). Desta forma, percebe-se que a rede adquiriu a capacidade de generalizar os dados apresentados a ela na etapa de treinamento, sendo evitado, portanto, o sobre-ajuste. Deve-se ressaltar ainda, que evitar tal problema é uma característica do método de treinamento utilizado.



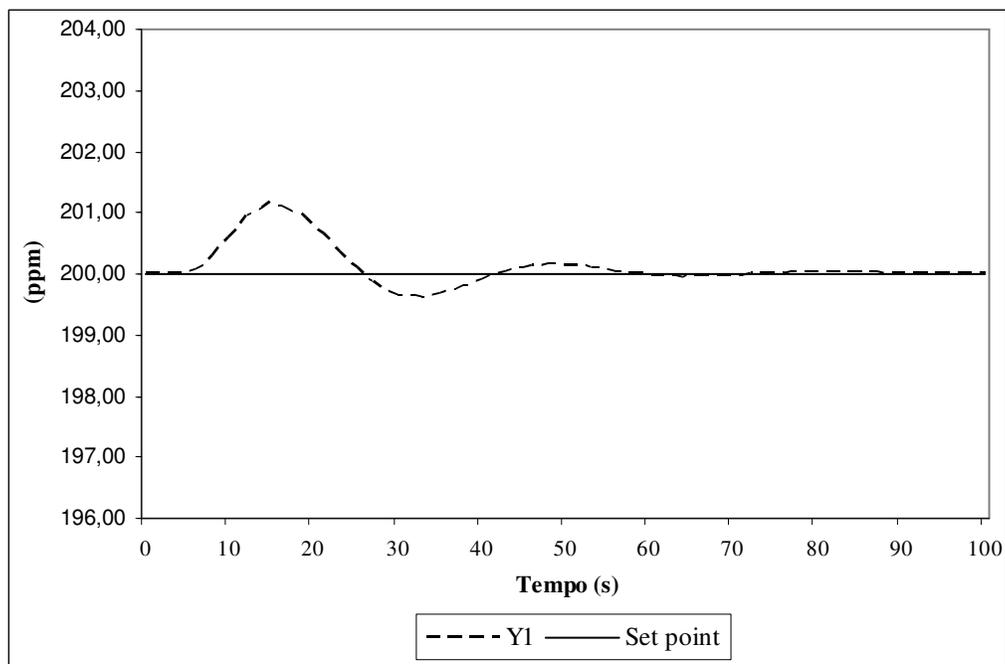
**Figura 4.5 – Relação entre a resposta fornecida pela rede e a saída desejada para o conjunto de validação**

#### **4.4 COMPARAÇÃO DE DESEMPENHO (DMC E RNA)**

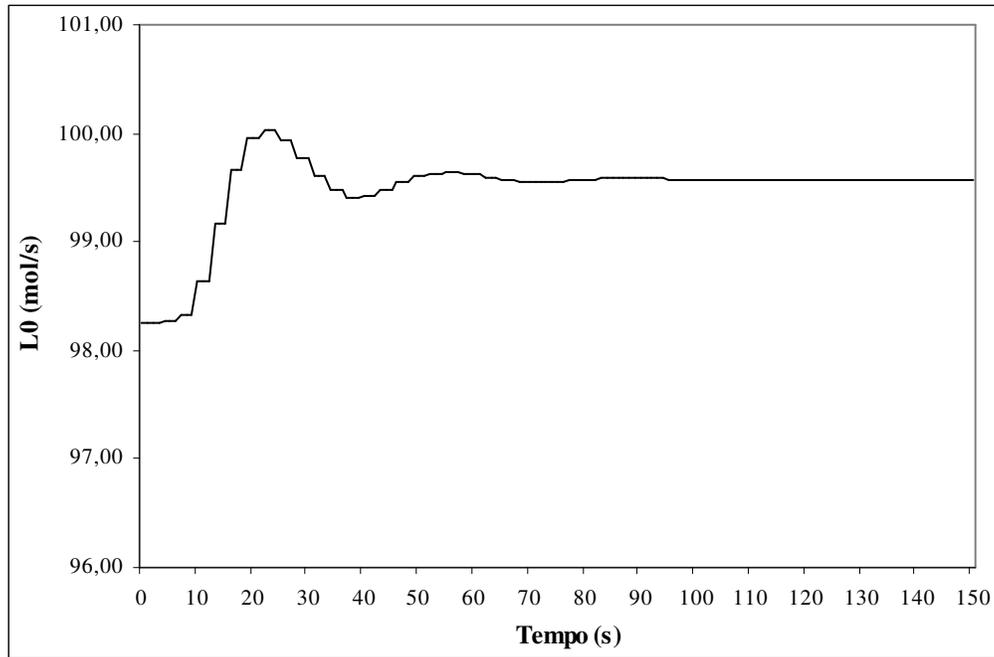
Na Figura 4.6 é apresentado um gráfico demonstrativo da resposta do processo, sob o controle via redes neurais proposto, quando submetido a uma perturbação degrau de 10% em  $Y_{N+1}$ . Nota-se que o controlador foi capaz de fazer com que a composição do gás à

saída da coluna,  $Y_1$  (variável controlada), voltasse ao ponto de ajuste, sendo necessário para que isto ocorresse, cerca de 90 segundos. O comportamento da variável manipulada (vazão de solvente  $L_0$ ), é demonstrado na Figura 4.7.

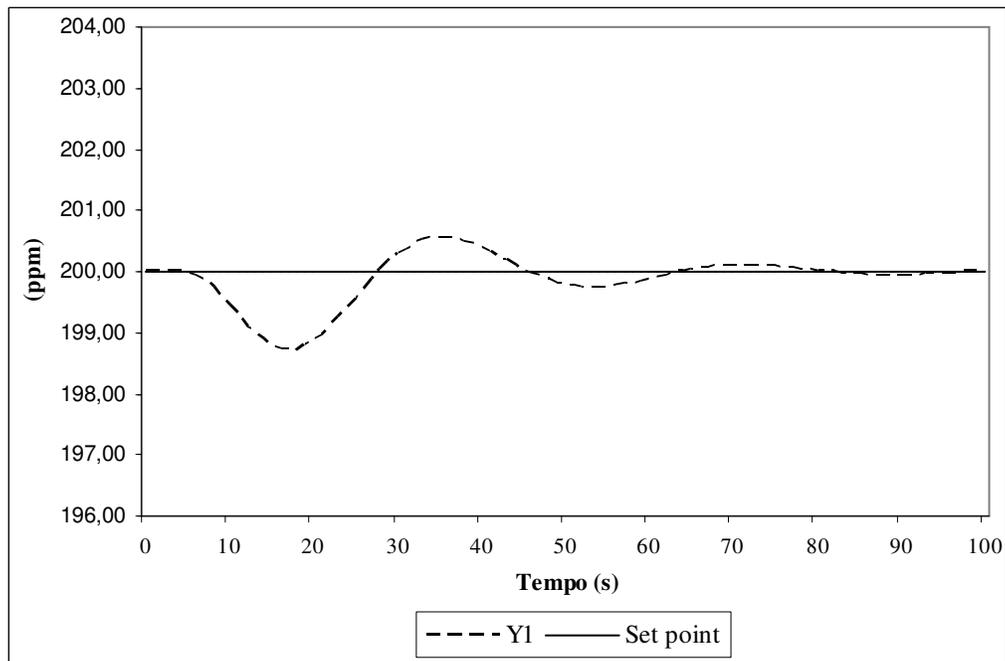
Já na Figura 4.8, ao contrário da Figura 4.6, demonstra-se o comportamento da variável controlada quando o sistema é submetido a uma perturbação de magnitude diferente das implementadas na etapa de treinamento (-8,7% em  $Y_{N+1}$ ). Ressalta-se o bom desempenho do controlador RNA, caracterizando assim, a capacidade de generalização da rede neural artificial. A manipulação realizada na vazão de solvente,  $L_0$ , é demonstrada na Figura 4.8.



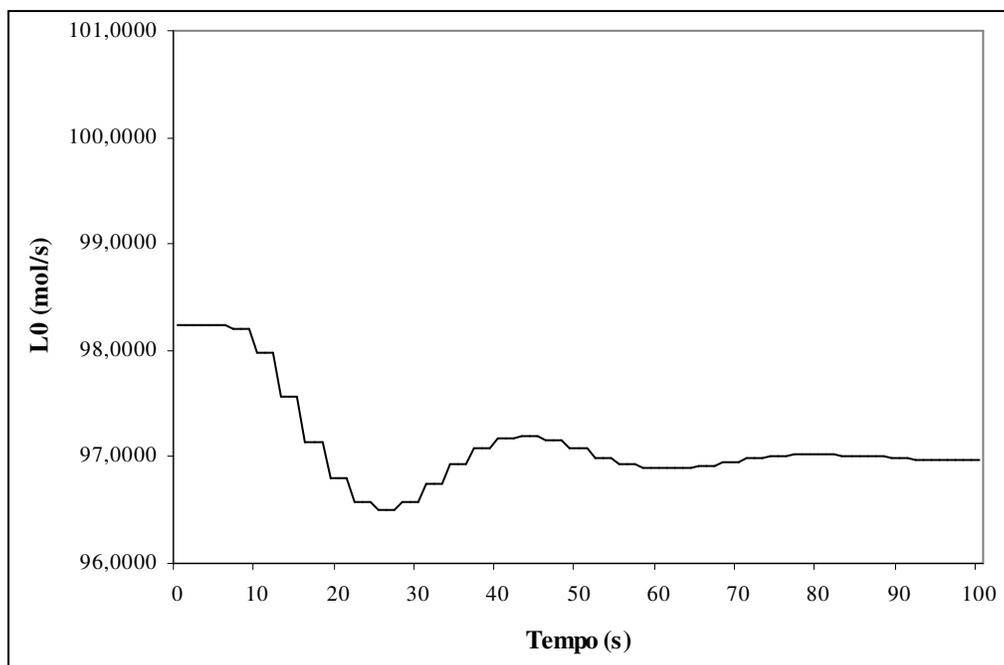
**Figura 4.6 – Resposta do processo em malha fechada com RNA frente a uma perturbação degrau, de magnitude integrante do banco de dados de treinamento da rede**



**Figura 4.7 – Comportamento da variável manipulada frente a uma perturbação degrau, de magnitude integrante do banco de dados de treinamento da rede**



**Figura 4.8 – Resposta do processo em malha fechada com RNA frente a uma perturbação degrau, de magnitude ausente no banco de dados de treinamento da rede**



**Figura 4.9 – Comportamento da variável manipulada frente a uma perturbação de degrau, de magnitude ausente no banco de dados de treinamento da rede**

Apesar de ser visível (Figuras 4.6 e 4.8) o bom desempenho do controlador RNA, faz-se necessário o confronto com estratégias de controle presentes na literatura. Neste sentido, pode-se utilizar como base de comparação, a técnica de Controle por Matriz Dinâmica (DMC), pois, em seu trabalho PALÚ (2001) comprovou a eficiência desta, quando aplicada ao sistema estudado.

Como parâmetros demonstrativos de desempenho se propôs a utilização da integral do quadrado do erro (ISE), onde o erro representa a diferença entre a resposta do processo e o ponto de ajuste, além da quantidade de solvente consumida.

Assim, o cálculo do erro e do parâmetro ISE pode ser efetuado através das Equações 4.01 e 4.02, respectivamente.

$$e = Y_1 - Y_{Set\ po\ int} \quad (4.01)$$

$$ISE = \int_0^t e^2 dt \quad (4.02)$$

A quantidade de solvente gasta pode ser obtida por meio da integral da vazão de solvente empregada, conforme Equação 4.03.

$$\text{Quantidade de solvente gasta} = \int_0^t L_0 dt \quad (4.03)$$

A Tabela 4.2 apresenta os valores destes dois parâmetros, obtidos para situações nas quais o sistema fora submetido a perturbações degrau de diferentes intensidades em  $Y_{N+1}$ .

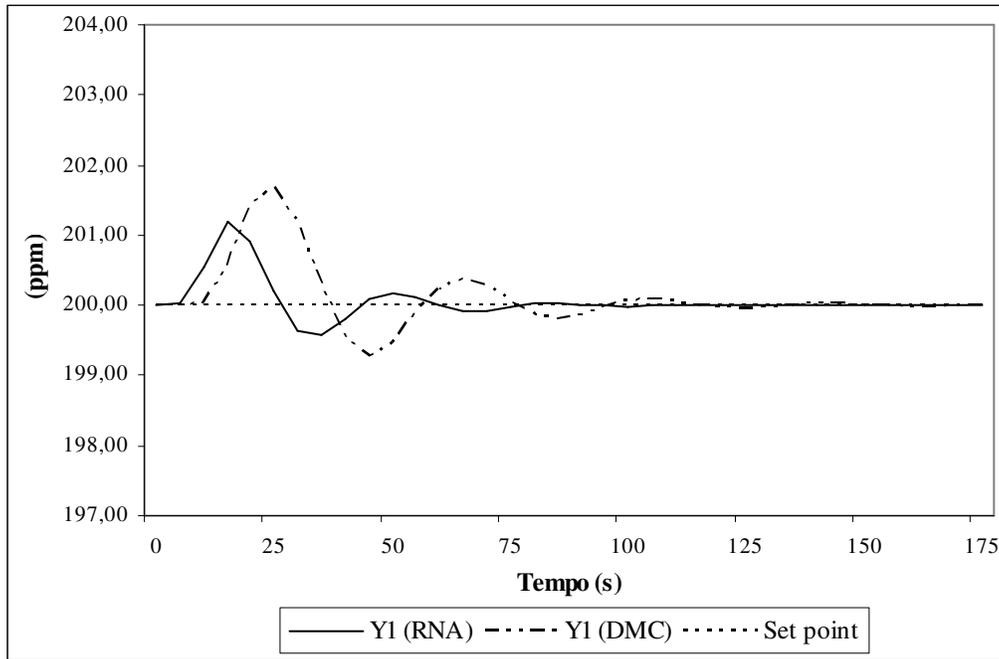
Nota-se que o controle RNA proporciona um valor menor para o parâmetro ISE, o que significa que a variável controlada permanece mais próxima do ponto de ajuste do que no caso do controle ser efetuado pelo DMC. Deve-se ressaltar ainda, que a integral do quadrado do erro possui valores pequenos por se tratar de unidade de concentração ao quadrado ( $\text{ppm}^2$ ).

Observando-se o consumo de solvente apresentado na Tabela 4.2 para cada técnica de controle, pode-se perceber que as duas se equivalem. Destaca-se a coerência dos valores obtidos para o gasto de solvente em relação a variação na composição do gás à entrada da coluna, pois, ao aumentar-se a porcentagem de álcool a ser absorvido, necessita-se de mais solvente para realizar tal tarefa, e vice-versa.

**Tabela 4.2 – Comparativo de desempenho entre a estratégia de controle RNA e DMC**

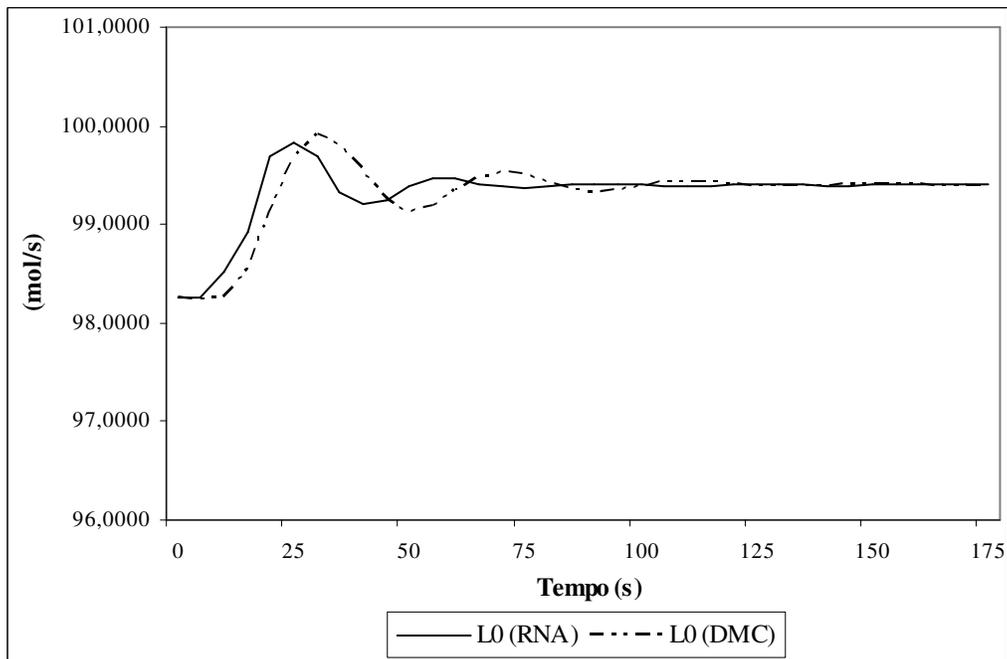
<b>Perturbação em <math>Y_{N+1}</math> (%)</b>	<b>ISE (ppm<sup>2</sup>)</b>	<b>Quantidade de solvente (mol)</b>	<b>Técnica de Controle empregada</b>
-8,3	$1,6380 * 10^{-08}$	29474,40	Malha aberta
	$3,3900 * 10^{-11}$	29133,16	Controle DMC
	$1,6600 * 10^{-11}$	29125,54	Controle RNA
-5,5	$7,1953 * 10^{-09}$	29474,40	Malha aberta
	$1,4700 * 10^{-11}$	29251,39	Controle DMC
	$6,8700 * 10^{-12}$	29246,36	Controle RNA
+2,1	$1,0448 * 10^{-09}$	29474,40	Malha aberta
	$2,1000 * 10^{-12}$	29556,37	Controle DMC
	$8,8500 * 10^{-13}$	29558,17	Controle RNA
+8,6	$1,7561 * 10^{-08}$	29474,40	Malha aberta
	$3,4000 * 10^{-11}$	29800,94	Controle DMC
	$1,5000 * 10^{-11}$	29808,09	Controle RNA

A seguir (Figura 4.10) apresenta-se um gráfico comparativo entre os desempenhos dos controladores RNA e DMC. Para tanto, inseriu-se uma perturbação degrau de 10% em  $Y_{N+1}$ .



**Figura 4.10 – Comparação entre o desempenho dos controladores RNA e DMC**

Na Figura 4.11 demonstra-se o comportamento da variável manipulada (vazão de solvente) vinculada à resposta ilustrada na Figura 4.10.



**Figura 4.11 – Vazão de solvente determinada pelos controladores RNA e DMC**

#### 4.5 SIMULAÇÃO COM O DISPOSITIVO DE SEGURANÇA (PERTURBAÇÕES NÃO PREVISTAS NO TREINAMENTO)

Para simular uma situação na qual a rede neural fosse exposta a distúrbios, pra os quais esta não teria recebido informações na fase de treinamento, implementaram-se perturbações na composição do solvente alimentado à coluna,  $X_0$ . Conforme pode ser observado na Figura 4.12 à medida que a intensidade da perturbação aumenta, cai a eficiência do controlador RNA, como previsto. Neste teste foi possível notar a presença de *off set* na resposta do processo, assim como uma sobrelevação muito maior que as usuais, obtidas nas simulações onde a variável perturbada era a composição do gás à entrada da coluna,  $Y_{N+1}$ . Na Figura 4.13 estão representados os comportamentos da variável manipulada para o mesmo teste realizado.

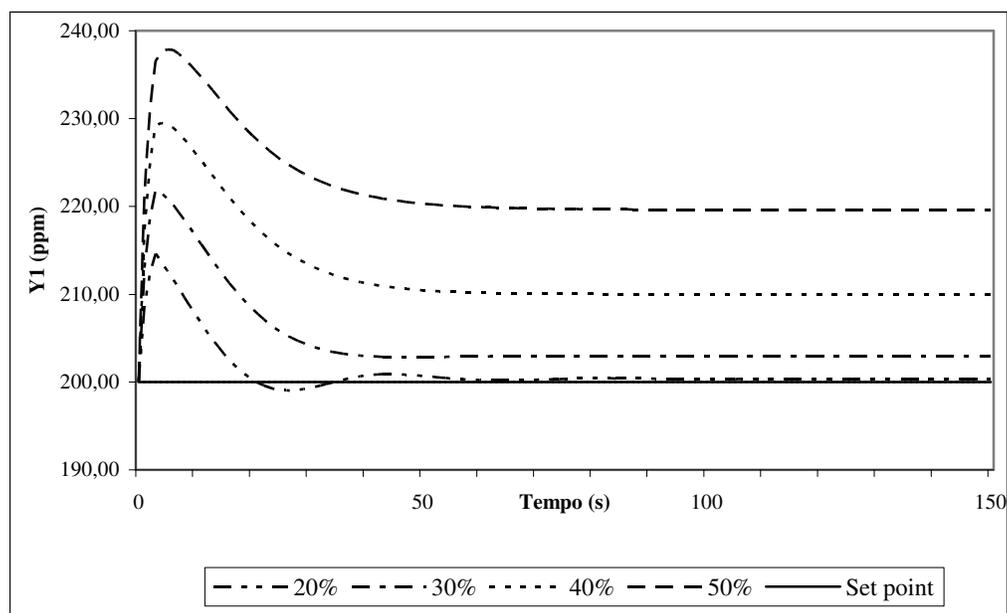


Figura 4.12 – Desempenho do controle RNA quando utilizado distúrbio diferente do apresentado à rede no treinamento

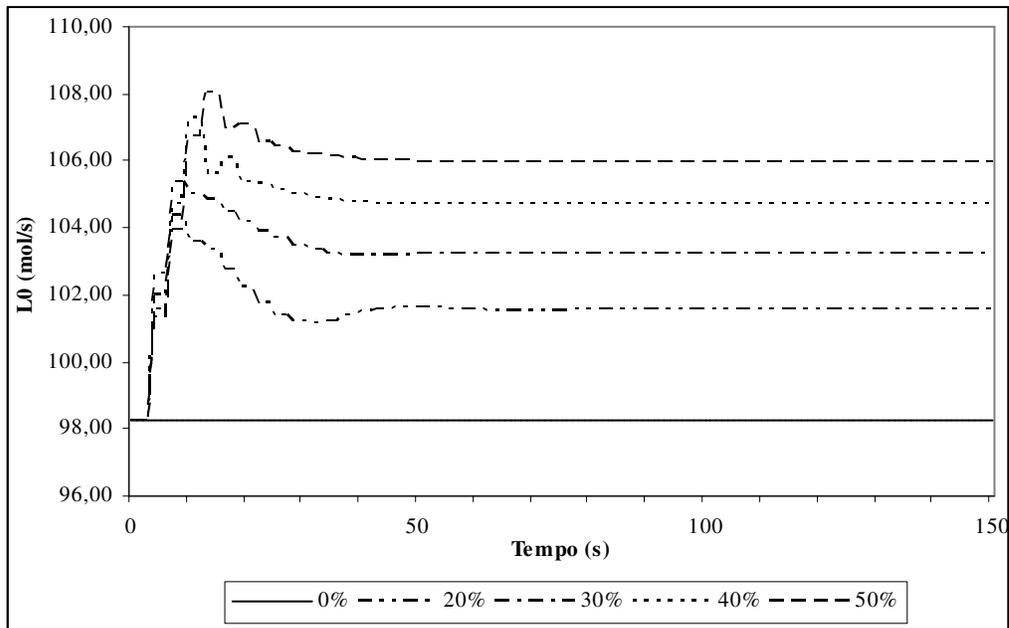
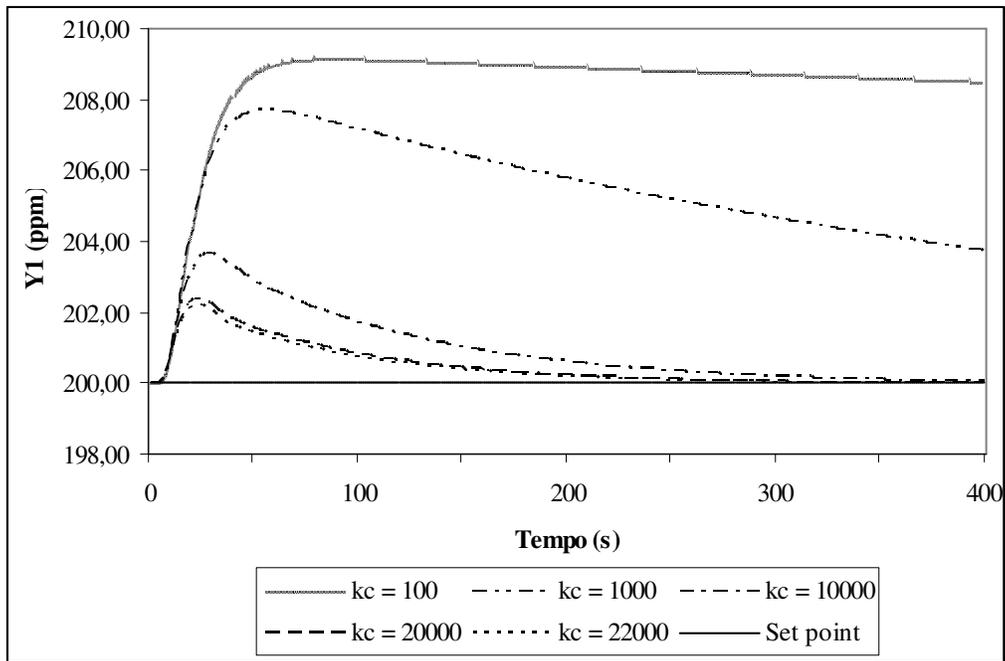


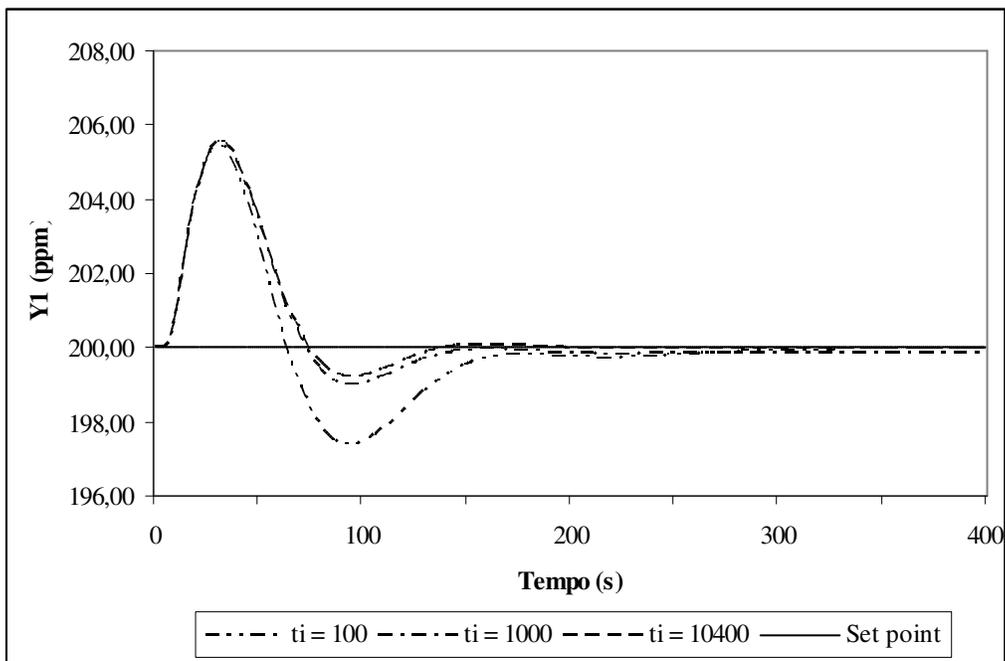
Figura 4.13 – Variável manipulada quando utilizado distúrbio diferente do apresentado à rede no treinamento

A situação simulada, não prevista na etapa de treinamento, é apenas uma, das tantas possíveis. Sendo assim, faz-se necessário o monitoramento do desempenho do sistema de controle. Neste sentido determinou-se arbitrariamente, um limite máximo de 5% para a diferença, em módulo, entre a resposta do processo e o ponto de ajuste. Caso este valor seja ultrapassado, um controlador *feedback* convencional (PI ou PID) assume a manipulação da vazão de solvente.

Primeiramente realizou-se o ajuste dos parâmetros para o controlador PI ( $k_c$  e  $\tau_I$ ). Conforme demonstrado nas Figuras 4.14 e 4.15, os melhores resultados foram obtidos quando utilizados  $k_c = 22000$  e  $\tau_I = 10400$ , ressaltando-se que em todas as simulações, uma perturbação de 10% em  $Y_{N+1}$  foi inserida no instante inicial de tempo. Quanto à magnitude do valor de  $k_c$ , esta pode ser justificada pela unidade do sinal de erro (ppm). Já o valor de  $\tau_I$ , foi obtido pelo método de tentativa e erro.



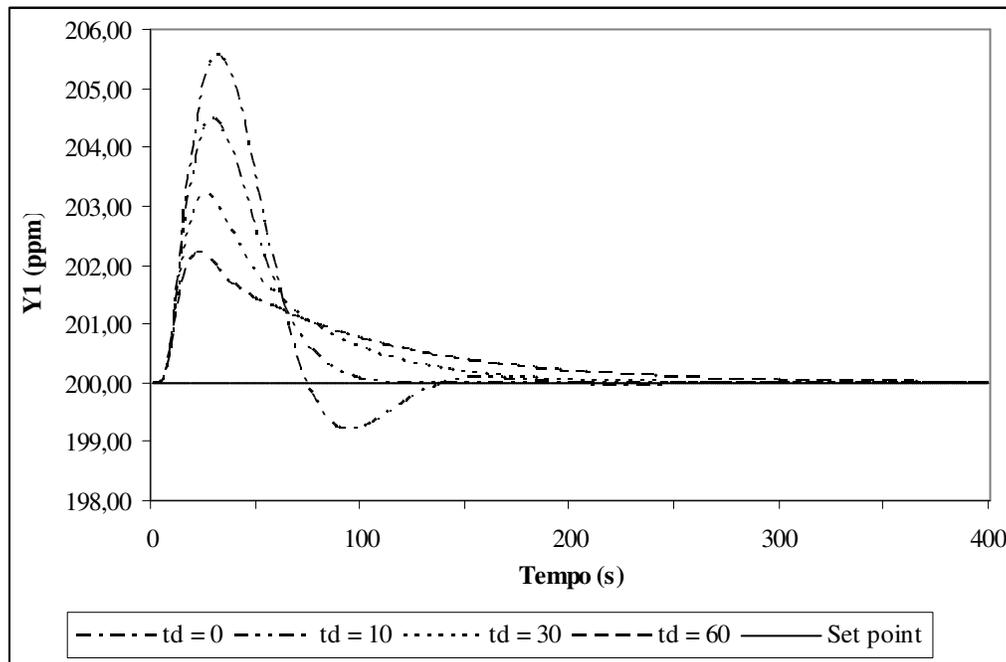
**Figura 4.14 – Ajuste do parâmetro  $k_c$**



**Figura 4.15 – Ajuste do parâmetro  $\tau_I$**

Na seqüência, fez-se uma comparação entre o desempenho do controlador PI e o PID, através do ajuste do parâmetro  $\tau_D$ , apresentado na Figura 4.16. Foi possível perceber que o PID se mostrou superior ao PI, pois quando o valor de  $\tau_D$  é aumentado progressivamente até 60, nota-se que há uma diminuição da sobre-elevação da variável

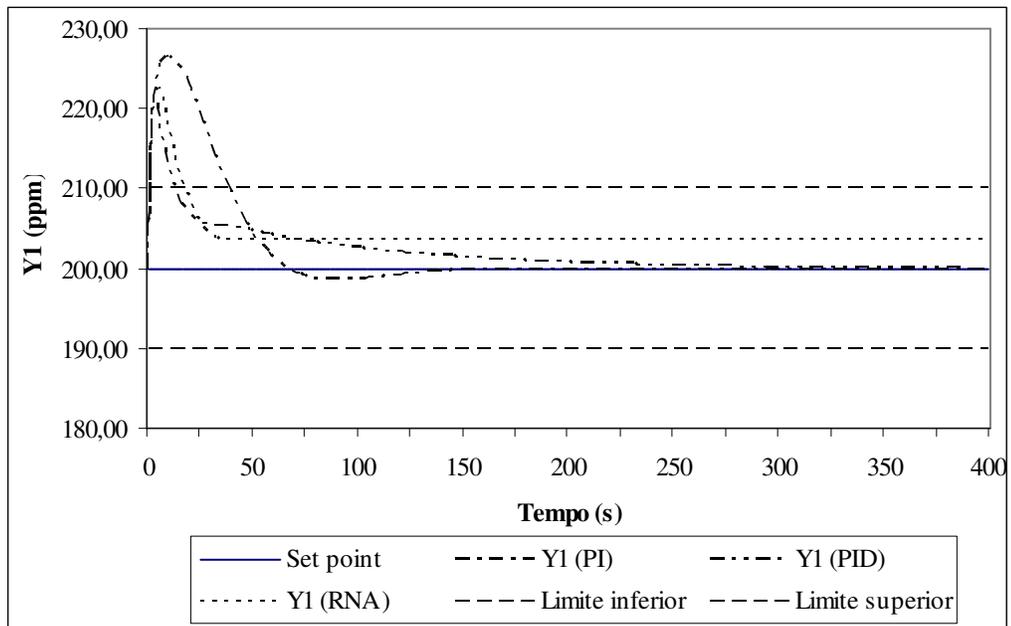
controlada. Sendo assim, para a função de dispositivo de segurança optou-se por utilizar o controlador PID com o conjunto de parâmetros composto por  $k_c = 22000$ ,  $\tau_I = 10400$  e  $\tau_D = 60$ .



**Figura 4.16 – Ajuste do parâmetro  $\tau_D$**

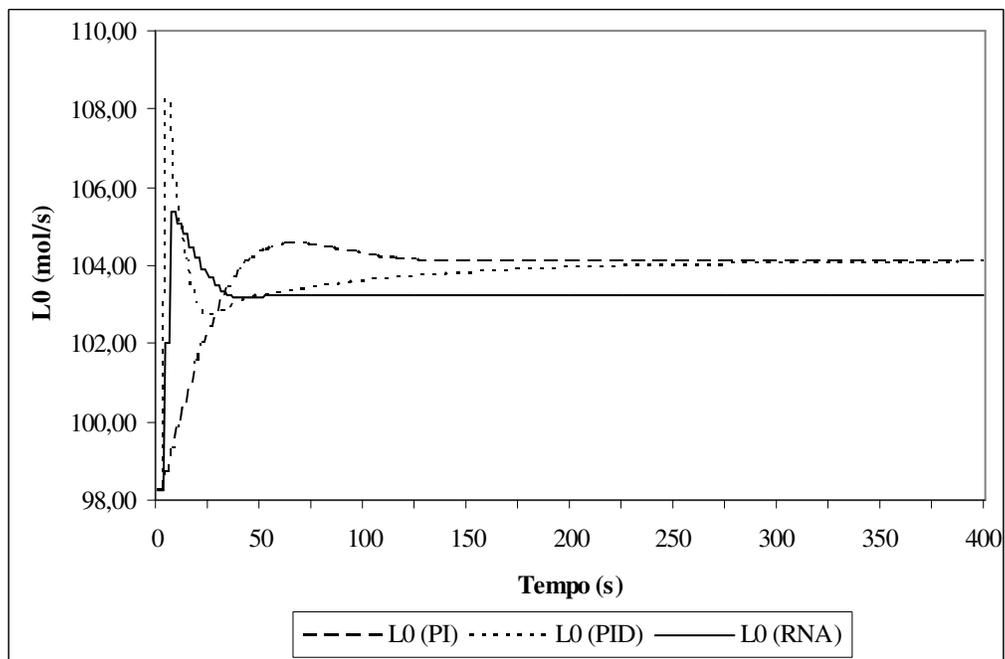
Depois de definido o tipo de controlador a ser utilizado como dispositivo de segurança, e de ter realizado o ajuste dos parâmetros deste, de modo a proporcionar uma ação de controle satisfatória, executou-se um teste no qual uma perturbação degrau de 30% foi inserida na variável  $X_0$ , no instante de tempo inicial,  $t = 0$ .

Como o erro apresentado pela resposta do processo assumiu valores superiores ao limite tolerado, o dispositivo de segurança foi acionado, permitindo que o controlador PID fosse requisitado. A Figura 4.17 faz um comparativo entre o comportamento da variável controlada sob a supervisão do dispositivo de segurança escolhido (PID), o comportamento do dispositivo de segurança rejeitado (PI) e o comportamento desta quando submetida ao controle exclusivo RNA. Nota-se que por intermédio do controlador PID o *off set* é eliminado.



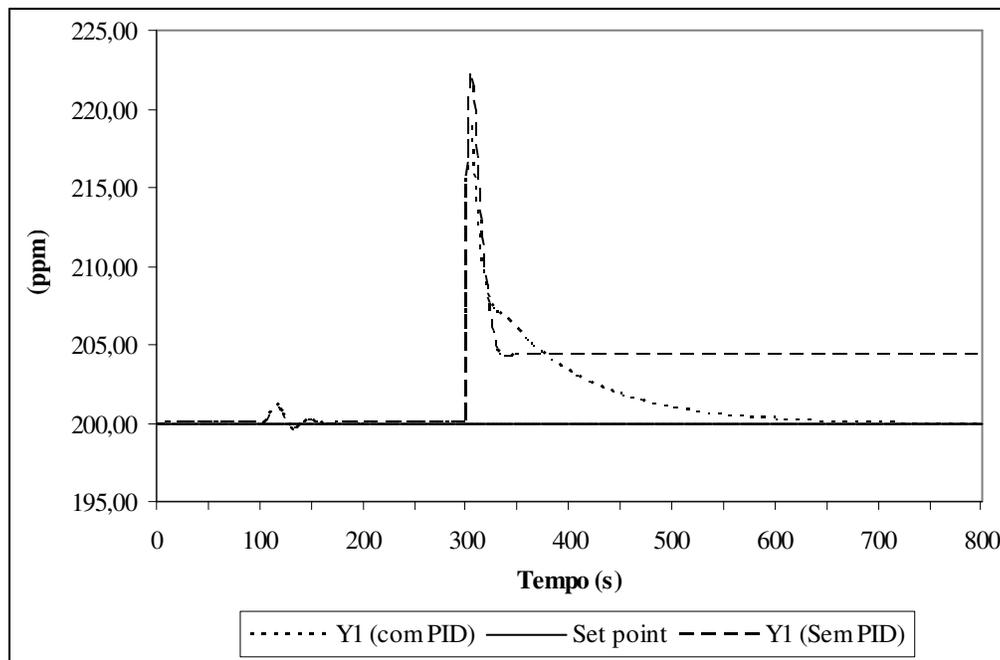
**Figura 4.17 – Comparação entre o desempenho do controle RNA, com e sem os dispositivos de segurança analisados**

Na Figura 4.18 é apresentada a manipulação da vazão de solvente para o teste descrito anteriormente, cujo resultado foi apresentado na Figura 4.17.

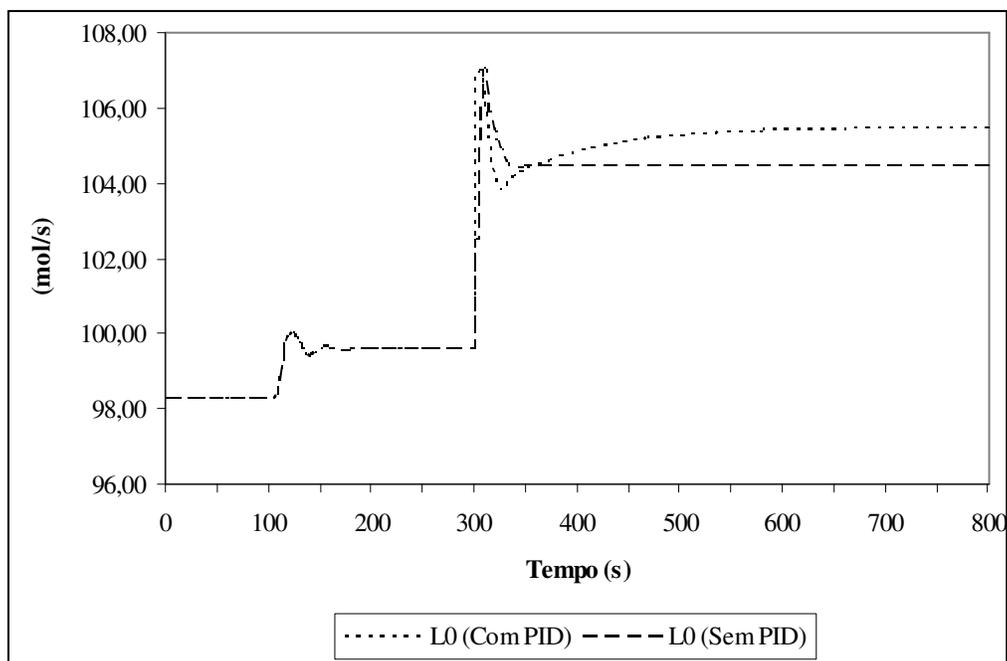


**Figura 4.18 – Comparação entre a manipulação da vazão de solvente pelo controlador exclusivo RNA e através do controle RNA/PID e RNA/PI**

Nas Figuras 4.19 e 4.20 são demonstrados, respectivamente, a resposta do processo e o comportamento da variável controlada para uma situação na qual duas perturbações degrau foram inseridas. A primeira delas, implementada no instante de tempo  $t = 100s$ , ocasionou um erro dentro da faixa tolerável, conforme o esperado, pois um distúrbio de 10% em  $Y_{N+1}$  fora previsto na etapa de treinamento da rede. Já a segunda perturbação fez com que o limite de 5% para o erro fosse excedido, uma vez que o distúrbio de 30% em  $X_0$  foi dado em  $t = 300s$ . Ressalta-se tanto o bom desempenho do controle RNA, quanto a ação eficiente do dispositivo de segurança, demonstrados frente ao primeiro e segundo distúrbios, respectivamente.



**Figura 4.19 – Resposta do processo frente a duas perturbações**



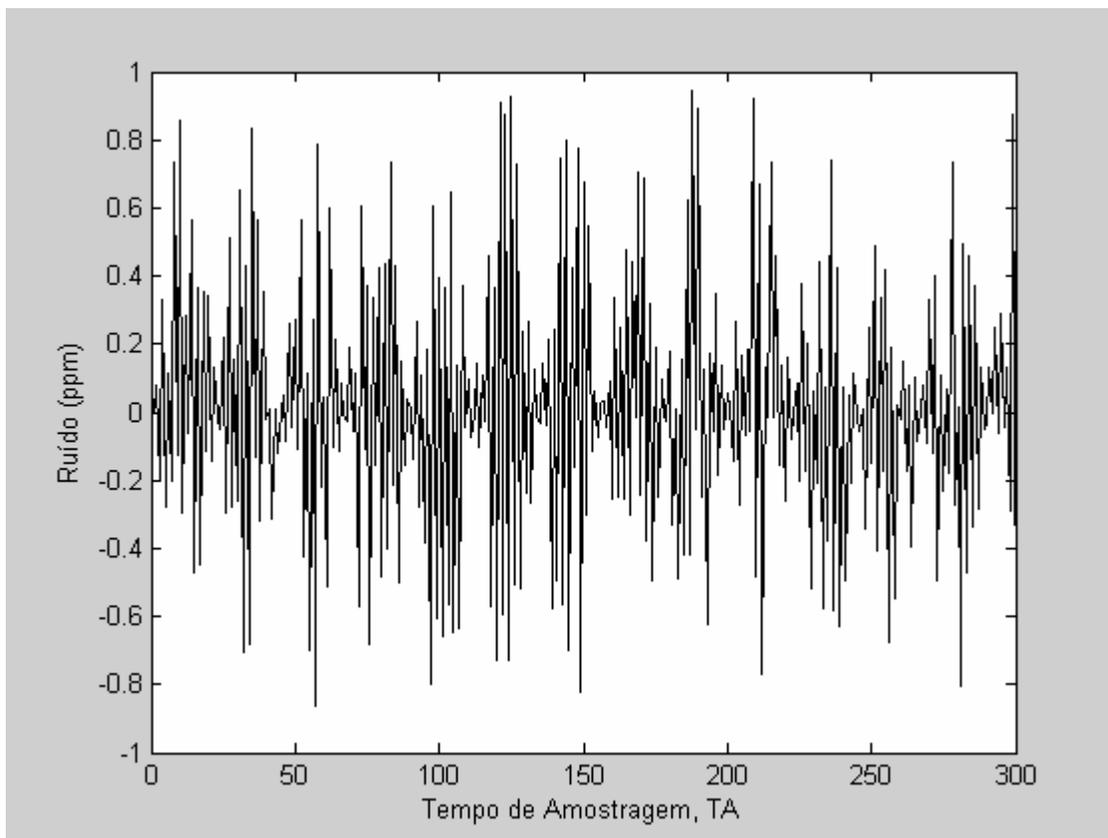
**Figura 4.20 – Comportamento da variável manipulada frente a duas perturbações**

#### **4.6 IMPLEMENTAÇÃO DO RUÍDO**

Até o momento foi considerado que os dados retirados da planta química refletiam exatamente os valores das variáveis de interesse, como por exemplo, as medidas da composição do gás à entrada e à saída da coluna. No entanto, sabe-se que a tomada destes dados sofre a interferência do meio externo, fazendo com que o valor medido seja diferente do valor real.

Visando aproximar os valores das variáveis de entrada fornecidos à rede, dos obtidos em uma situação real, acrescentou-se um ruído aos dados gerados pelo programa computacional.

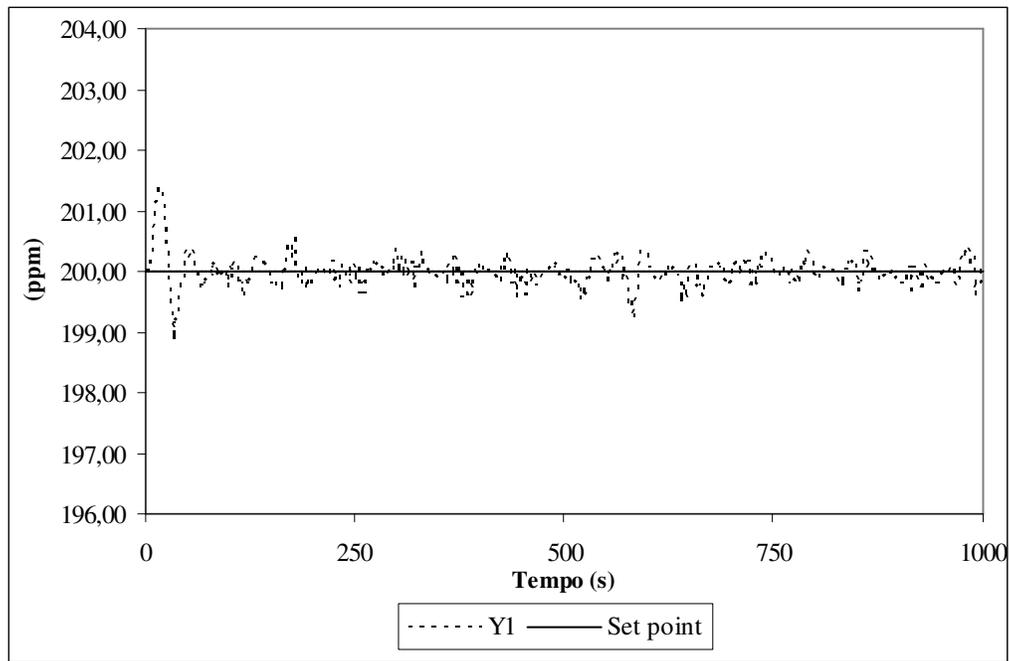
A base do ruído utilizado é uma função seno, de amplitude controlada e alta frequência. A estrutura do ruído, contém ainda, a geração de um número aleatório compreendido entre 0 e 1, que multiplicado à função senoidal fornece o valor final do ruído. A Figura 4.21 ilustra o ruído inserido nas variáveis de composição requeridas pelo sistema de controle. Nota-se que o ruído empregado varia de -1 a +1 ppm.



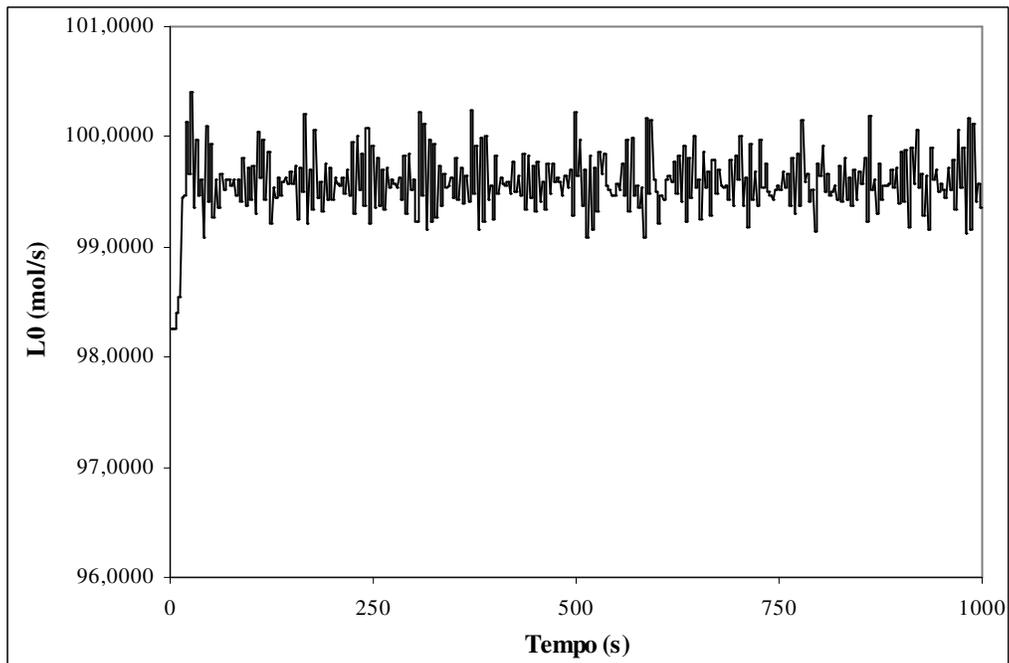
**Figura 4.21 – Ruído**

Como esperado ao distorcer os valores referentes às variáveis de entrada da rede, uma situação não prevista na fase de treinamento, o sistema de controle RNA sofreu uma queda de eficiência. Entretanto, o limite crítico de 5% para a diferença entre o valor da variável controlada e o ponto de ajuste não foi ultrapassado, sendo desnecessária a interferência do dispositivo de segurança, o que demonstra a robustez do sistema de controle via redes neurais quanto à tolerância de falhas.

Na Figura 4.22 é apresentada a resposta do processo quando o sistema é submetido a uma situação na qual as variáveis de entrada da rede  $Y_{1k}$ ,  $Y_{1k-1}$  e  $Y_{N+1k}$  são distorcidas através da inserção do ruído. Além disso, uma perturbação degrau de 10% é implementada em  $Y_{N+1}$ , no instante de tempo  $t = 0$ . A vazão de solvente empregada neste teste é apresentada na Figura 4.23.



**Figura 4.22 – Desempenho do controlador RNA para uma situação na qual os dados de entrada da rede apresentam ruído**



**Figura 4.23 – Vazão de solvente manipulada pelo controlador RNA alimentado com dados apresentando ruído**

Na ausência de ruído a resposta do processo seria estabilizada no *set point* (200 ppm), em cerca de 90s, conforme já demonstrado na Figura 4.6. Entretanto, na presença

deste, a variável controlada oscila entre 200,5 e 199,5 ppm, ou seja, a faixa de oscilação é de 1 ppm. Apesar de não conseguir a estabilização da composição do gás à saída da coluna, nota-se que pela ação do controlador RNA, a faixa de oscilação é reduzida à metade da faixa de ruído empregada, 2 ppm (de -1 a +1 ppm).

#### 4.7 IMPLEMENTAÇÃO DO FILTRO

Um dos recursos que pode ser utilizado para reduzir os efeitos do ruído, responsável pela distorção dos dados obtidos de uma planta química, é a filtração.

O filtro tem por finalidade atenuar as oscilações bruscas demonstradas pelas medidas. Constitui-se de uma soma ponderada dos valores da variável a ser filtrada, para o instante atual “k”, assim como anteriores, k-1, k-2, ..., k-NA. Onde NA representa o último instante de amostragem, anterior a k, que toma parte na soma. O cálculo do valor da variável filtrada é efetuado por meio da Equação 4.04.

$$Y|_{Filtrada} = \lambda_1 Y|_k + \lambda_2 Y|_{k-1} + \lambda_3 Y|_{k-2} + \dots + \lambda_{NA+1} Y|_{k-NA} \quad (4.04)$$

Onde:

$$Y|_{Filtrada} = \text{Variável filtrada.}$$

$$Y = \text{Variável a ser filtrada.}$$

$$\lambda_i = \text{Parâmetro do filtro, sendo que: } \sum_{i=1}^{NA+1} \lambda_i = 1$$

Os parâmetros do filtro,  $\lambda$ 's e NA, devem ser determinados de maneira que o objetivo do filtro seja alcançado, se possível. Assim, depois de testar várias configurações, verificou-se que os melhores resultados foram obtidos quando utilizados 15 instantes de amostragem anteriores a k, sendo a soma ponderada pelos seguintes coeficientes:

$\lambda_1 = 0,20$ ,  $\lambda_2 = 0,10$ ,  $\lambda_3 = 0,05$ ,  $\lambda_4 = 0,05$ ,  $\lambda_5 = 0,05$ ,  $\lambda_6 = 0,05$ ,  $\lambda_7 = 0,05$ ,  $\lambda_8 = 0,05$ ,  
 $\lambda_9 = 0,05$ ,  $\lambda_{10} = 0,05$ ,  $\lambda_{11} = 0,05$ ,  $\lambda_{12} = 0,05$ ,  $\lambda_{13} = 0,05$ ,  $\lambda_{14} = 0,05$ ,  $\lambda_{15} = 0,05$ ,  
 $\lambda_{16} = 0,05$ .

Para o sistema estudado, aplicou-se o filtro às mesmas variáveis que foram acrescidas do ruído ( $Y_{1|k}$ ,  $Y_{1|k-1}$  e  $Y_{N+1|k}$ ). Fazendo uso da Equação 4.04, bem como dos parâmetros definidos para o filtro, têm-se:

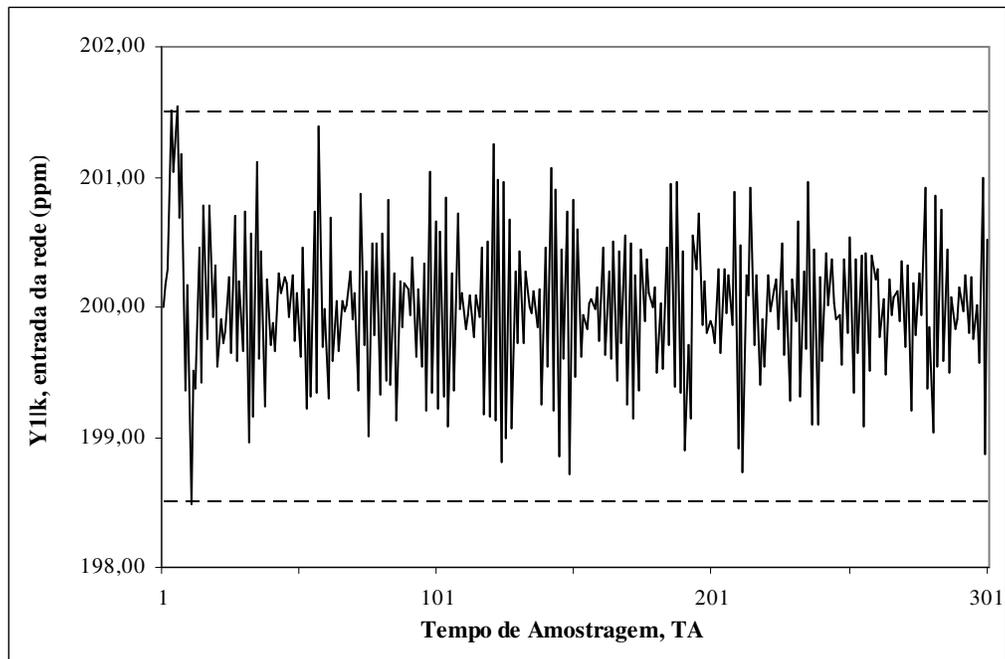
$$\left(Y_{1|k}\right)_{Filtrada} = 0,20\left(Y_{1|k}\right)_k + 0,10\left(Y_{1|k}\right)_{k-1} + 0,05\left(Y_{1|k}\right)_{k-2} + \dots + 0,05\left(Y_{1|k}\right)_{k-15} \quad (4.05)$$

$$\left(Y_{1|k-1}\right)_{Filtrada} = 0,20\left(Y_{1|k-1}\right)_k + 0,10\left(Y_{1|k-1}\right)_{k-1} + 0,05\left(Y_{1|k-1}\right)_{k-2} + \dots + 0,05\left(Y_{1|k-1}\right)_{k-15} \quad (4.06)$$

$$\left(Y_{N+1|k-1}\right)_{Filtrada} = 0,20\left(Y_{N+1|k-1}\right)_k + 0,10\left(Y_{N+1|k-1}\right)_{k-1} + 0,05\left(Y_{N+1|k-1}\right)_{k-2} + \dots + 0,05\left(Y_{N+1|k-1}\right)_{k-15} \quad (4.07)$$

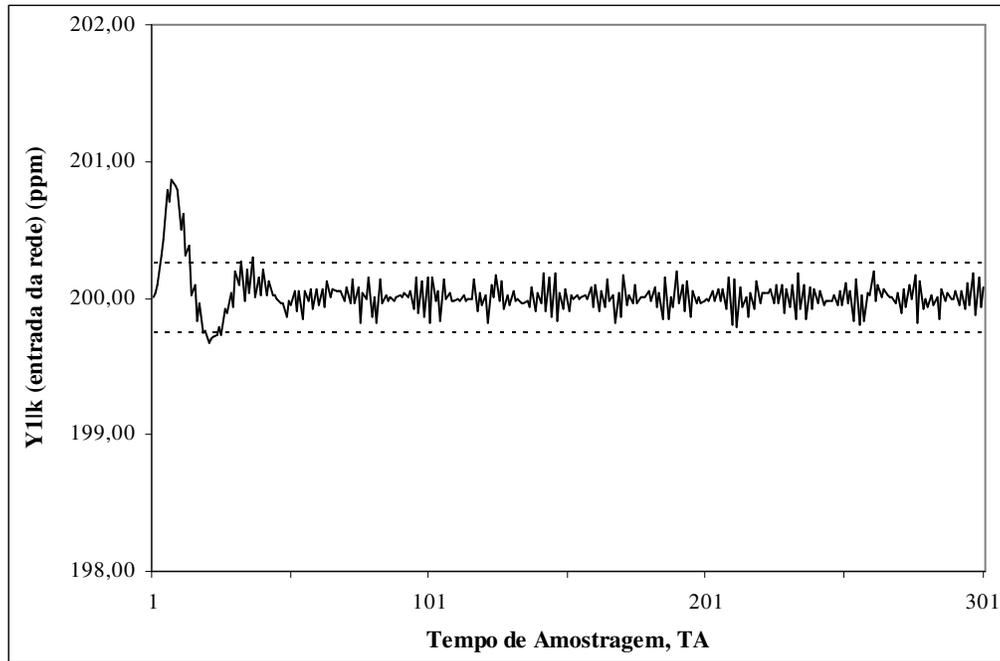
As Figuras 4.24 e 4.25 demonstram a diferença do comportamento da variável  $Y_{1|k}$  (entrada da rede neural) para uma situação sem a aplicação do filtro, e outra, depois da utilização deste, respectivamente.

Nota-se que no primeiro caso, de acordo com a intensidade do ruído empregado ( $\pm 1$  ppm), a variável mostrou-se oscilatória dentro de uma faixa de aproximadamente 3 ppm, salvo no início da simulação, onde sob o efeito de uma perturbação de grau de 10% em  $Y_{N+1}$ , registraram-se valores que ultrapassaram esta faixa.



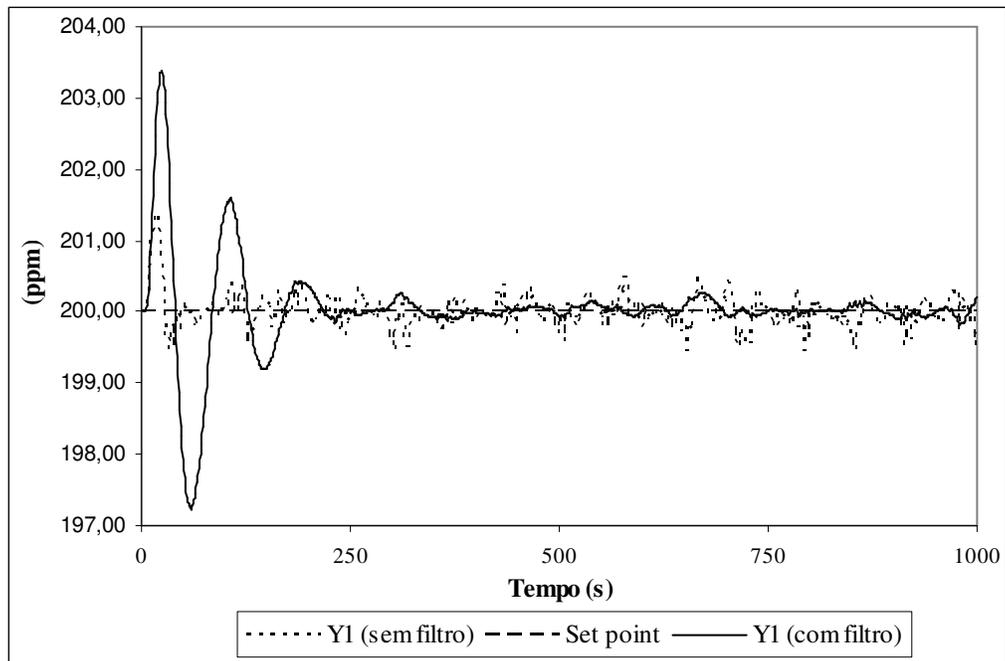
**Figura 4.24 – Variável de entrada da rede,  $Y_{1|k}$ , distorcida pela ação do ruído**

Depois de filtrada, a variável de entrada da rede  $Y_{1|k}$  teve atenuada sua oscilação, além da sobrelevação apresentada por causa da reação do sistema ao distúrbio implementado em  $t = 0$ . A faixa de oscilação caiu para 0,5 ppm, cerca de 17% da obtida na simulação sem o filtro.



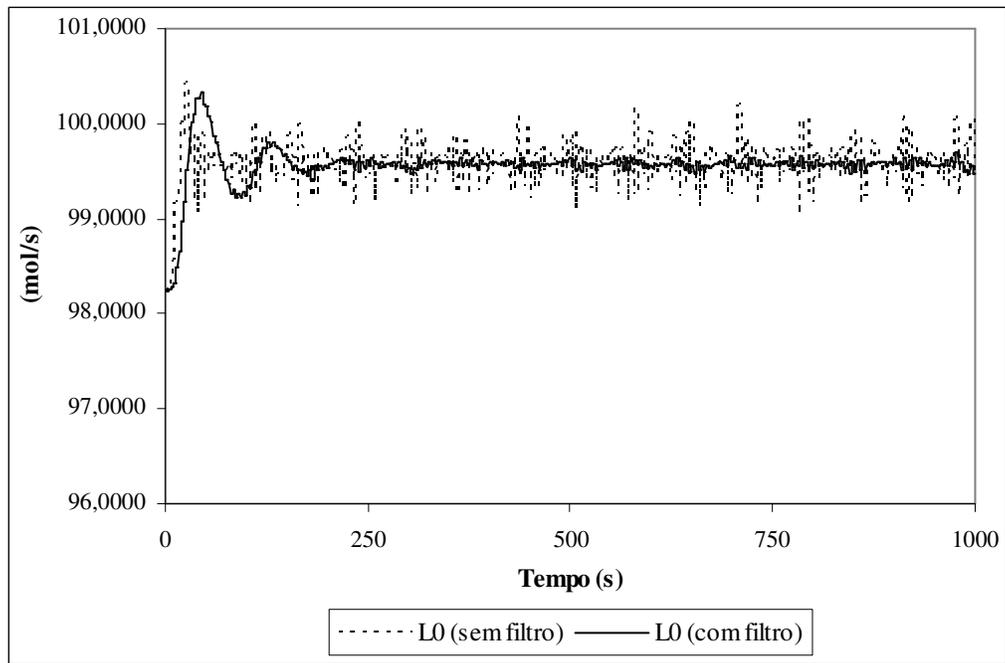
**Figura 4.25 - Variável de entrada da rede,  $Y_{1|k}$ , distorcida pela ação do ruído e atenuada pelo filtro**

A seguir (Figura 4.26), é apresentada a resposta do processo para uma simulação na qual foi inserida uma perturbação degrau de 10% em  $Y_{N+1}$  em  $t = 0$ . Além disso, as variáveis de entrada da rede  $Y_{1|k}$ ,  $Y_{1|k-1}$  e  $Y_{N+1|k}$  foram distorcidas pelo acréscimo do ruído, e na seqüência, filtradas. A curva da Figura 4.22 é incorporada à Figura 4.26 para que se possa visualizar a influência do filtro.



**Figura 4.26 – Resposta do processo para uma situação na qual as variáveis de entrada da rede são distorcidas pelo ruído e atenuadas pelo filtro**

Apesar da sobrelevação demonstrada ter sido maior que a obtida na simulação sem o filtro, verifica-se uma queda de 75% na oscilação da resposta do processo. Na Figura 4.27 demonstra-se o comportamento da variável manipulada, vinculado à resposta do processo, apresentada na Figura 4.26.



**Figura 4.27 – Comparativo entre a manipulação da vazão de solvente pelo controlador RNA alimentado com dados apresentando ruído, com e sem filtração posterior**

**CAPÍTULO 5**

**CONCLUSÕES E SUGESTÕES PARA**

**TRABALHOS FUTUROS**

## 5.1 CONCLUSÕES

O presente trabalho teve por objetivo a aplicação de uma nova estratégia de controle à coluna de absorção utilizada na recuperação do álcool etílico, perdido por evaporação durante o processo de fermentação. Este equipamento é caracterizado por sua não linearidade, o que prejudica o desempenho de técnicas de controle convencionais. Neste sentido, o controlador *feedforward* baseado em modelo inverso de redes neurais artificiais se mostrou detentor de uma grande capacidade de processamento não linear, o que foi extremamente útil na garantia de um controle satisfatório.

Durante a determinação da arquitetura da rede buscou-se dimensioná-la para que, com o menor esforço computacional possível, a rede pudesse dispor de um aprendizado de boa qualidade, agregando a esta, a capacidade de generalização. Vários arranjos foram testados, porém, os melhores resultados foram obtidos quando utilizada uma arquitetura de configuração 5 – 9 – 1. Esta verificação se deu com base no critério de HAGAN e FORESSE (1997) (conforme Figura 4.4) e foi confirmada pelos valores da soma dos erros quadrados (SSE) e média dos erros quadrados (MSE), demonstrados na Tabela 4.1. Além disso, pela Figura 4.5, nota-se que a rede foi capaz de ajustar satisfatoriamente os dados integrantes do conjunto de validação, obtendo-se um coeficiente de correlação de 0,997.

Depois de implementada a nova estratégia de controle à coluna de absorção estudada, procedeu-se testes nos quais perturbações degrau eram inseridas na composição do gás à entrada da coluna e o desempenho do sistema de controle avaliado. Verificou-se através do comportamento da variável controlada (Figuras 4.6 e 4.8) que o controlador proposto foi capaz de manipular a vazão de solvente empregada na coluna de absorção de modo a fazer com que a composição do gás à saída da coluna retornasse ao seu ponto de ajuste.

Na seqüência foram realizados testes com o intuito de comparar o desempenho demonstrado pelo controlador RNA e o apresentado pelo controlador DMC (PALÚ, 2001). Na Figura 4.10 foi possível constatar a superioridade do controle RNA frente à técnica DMC, pois a variável controlada apresentou menor sobrelevação e o ponto de ajuste foi restaurado mais rapidamente (cerca de 90 segundos, enquanto que o controle DMC propiciou um tempo de aproximadamente 175 segundos). Além disso, na Tabela 4.2, os

valores do indicador de desempenho ISE (integral do quadrado do erro) também favoreceram o controle RNA. Quanto à quantidade de solvente consumida, cujos valores também foram expostos na mesma tabela, verificou-se a equivalência entre os dois controladores.

Inserindo-se perturbações não previstas na etapa de treinamento da rede, foi possível notar que à medida que a intensidade destes distúrbios aumenta, o desempenho do controlador RNA fica prejudicado. Sendo assim, implementou-se um dispositivo de segurança (controlador PID) que assume a manipulação da vazão de solvente a partir do momento que se detecta uma diferença maior que 5% entre a variável controlada e seu ponto de ajuste. Esta medida agregou robustez ao sistema de controle, conforme demonstrado nas Figuras 4.17 e 4.19.

Em outra etapa deste trabalho fez-se a análise do comportamento do controlador RNA quando alimentado com dados distorcidos (Figura 4.22). Como esperado, houve uma queda na eficiência do sistema de controle, no entanto, a amplitude da faixa de oscilação da variável controlada foi reduzida à metade da faixa de ruído empregada. Em seguida foi testada uma situação na qual os dados distorcidos pela inserção do ruído eram filtrados. O resultado, demonstrado pela Figura 4.26, revelou uma redução ainda maior da faixa de oscilação (cerca de 75% da faixa de ruído).

Com base nos vários testes realizados neste trabalho, nos quais foi constatado o ótimo desempenho do controlador *feedforward* baseado em modelo inverso de redes neurais, é possível concluir que este representa uma opção atrativa para o controle da coluna de absorção estudada. Além de operar de maneira eficiente quando na presença de distúrbios previstos na etapa de treinamento da rede, têm agregado a ele, robustez no tratamento de situações inéditas, através do dispositivo de segurança implementado em paralelo. Deve-se destacar ainda, a tolerância a distorções nos dados de entrada, o que implica em um aumento de confiabilidade do sistema de controle.

## 5.2 SUGESTÕES PARA TRABALHOS FUTUROS

Sugere-se que os próximos trabalhos aprofundem o estudo aqui apresentado. Para tanto, pode-se levar em conta os efeitos térmicos sobre a absorção, ou ainda, verificar a

influência do tempo morto e das incertezas dos sensores sobre o desempenho do controlador proposto.

Além disso, a técnica de controle empregada na coluna de absorção responsável pela recuperação do álcool perdido por evaporação durante o processo de fermentação pode ter seu uso expandido para outra coluna, com diferentes condições de operação.

## REFERÊNCIAS BIBLIOGRÁFICAS

BRAGA, A. de P., LUDERMIR, T. B., CARVALHO, A. C. P. de L. F., *Redes Neurais Artificiais teoria e aplicações*, Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora S.A., 2000.

DEMUTH, H., BEALE, M., *Neural Network Toolbox User's Guide for Use with MATLAB®*. Version 4, Reading: Hagan, M., 2002 Cap. 5: Backpropagation. Disponível em: <http://www.mathworks.com/support/product/NN/>. Acesso em 28/10/2004.

HAGAN, M. T., FORESSE, F. D., *Gauss-Newton Approximation to Bayesian Learning*. In Proceedings of the 1997 International Joint Conference on Neural Networks, 1997, URL: <http://elec-engr.okstate.edu/mhagan/>. Acesso em 06/11/2004.

KERNANI, B. G., SCHIFFMAN, S. S., NAGLE, H. T., *Performance of the Levenberg-Marquardt neural network training method in electronic nose applications*. Sensors and Actuators B, Elsevier B. V., 2005.

LUYBEN, W. L., *Process Modeling, Simulation and Control for Chemical Engineers*. McGraw – Hill Book Company, 1990.

MAIA, M. L. O. *Controle preditivo de Colunas de Absorção*. Campinas: UNICAMP, 1994. Tese (Mestrado)

MCCULLOCH, W., PITTS, W. *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Math. Bio., n.5, p. 115-133, 1943.

NAJIM, K., RUIZ, V. *Long-range predictive control of an absorption packed column*. Appl. Math. Modelling. v. 19, p. 39-45, 1995.

NASTARI, M. P. *Tendências mundiais para o uso de etanol*. 2005. Disponível em: [www.nipeunicamp.org.br/proalcool/Palestras/17/Nastari%20NIPE%20Unicamp%2017nov05%20presentation.ppt](http://www.nipeunicamp.org.br/proalcool/Palestras/17/Nastari%20NIPE%20Unicamp%2017nov05%20presentation.ppt). Acesso em 03/02/2006.

NUNES, G. C., KINCAL, S., CRISALLE, O. *A polynomial perspective on the stability of multivariable predictive controllers*. Computers and Chemical Engineering. v. 27, p. 1097-1111, 2003.

OGUNNAYKE, B. A., RAY, W. H., *Process Dynamics, Modeling, and Control*. Oxford University Press, 1994.

PACIANOTTO, T. A., *Aplicação de Redes Neurais para o ajuste operacional do período final de sopro de um processo de aciaria a oxigênio*. Campinas: UNICAMP, 2003. Tese (Mestrado).

PALÚ, F. *Controle preditivo de colunas de absorção com o método de controle por matriz dinâmica*. Campinas: UNICAMP, 2001. Tese (Doutorado)

PARTHASARATHY, S., GOWAN H., INDHAR, P. *Prediction of Flooding in an Absorption Column using Neural Networks*. International Conference on Control Applications, 1999.

RÓG, E., WILK, M. *Simulation of an absorption column performance using feed-forward neural networks in nitric acid production*. Computers Chemical Engineering. v. 22, p. 909-912, 1998.

RUMMELHART, D., MCCLELLAND, J. *Parallel Distributed Processing*. MIT Press, Cambridge, 1986.

SEBORG, D. E., EDGAR, T. F., MELLICHAMP, D. A., *Process Dynamics and Control*. Hamilton Printing Company, 1989.

SHERWOOD, T. K., PIGFORD, R. L., WILKE, C. R. *Mass Transfer*. McGraw-Hill, 1975.

SILVA, J. E. L. e, *Simulação e Controle Preditivo Linear (com Modelo de Convolução) e Não-Linear (Com Modelo baseado em Redes Neurais Artificiais) de*

*Colunas Recheadas de Absorção com Reação Química*. Campinas: UNICAMP, 1997. Tese (Mestrado).

STEPHANOPOULOS, G., *Chemical Process Control An Introduction to Theory and Practice*. PTR Prentice Hall, 1984.

TREYBAL, R. E., *Mass-Transfer Operations*. McGraw – Hill Book Company, 1981.

# **ANEXOS**

**ANEXO 01 - PROGRAMA COMPUTACIONAL EM MATLAB® -  
MODELO\_COLUNA\_DE\_ABSORCAO.M**

global N L X Y M MX G L0 X0 MM LW AP HV C DT ELN1 ELN EXN1 EXN EYN1  
 EYN DerM DerMX Rmax Rmin rad Ruido\_Y\_defasado Y\_atual

%#####

%Etapa 1 - Entrada de dados

N=9; % N = Numero de pratos  
 G=61.9; % G = Vazao de gas  
 X0=0.0001; % X0 = Fração molar do componente A no liquido de entrada  
 YN=0.01; % YN = Fração molar do componente A no gas de entrada  
 Y1=0.0002; % Y1 = Fração molar do componente A no gas de saida  
 MM=1.0682; % MM = Constante de equilibrio  
 LW=8.3; % LW = Comprimento do vertedouro  
 C=0.14322; % C = Constante  
 HV=8; % HV = Altura do vertedouro  
 AP=210; % AP = Area do prato  
 DT=0.05; % DT = Passo de integração  
 TEMPOFINAL=400; % TEMPOFINAL = Ultimo valor do tempo desejado  
 L0=98.248; % L0 = Vazao de solvente inicial (vazao otima)  
 Perturb\_Y=1; % Determina se ha perturbação em Y(N+1)/ 0-nao 1-sim  
 ValorP\_Y=[10]; % ValorP\_Y = Valor percentual da perturbação em Y(N+1)  
 tperturb\_Y=0; % Instante no qual ha a perturbação em Y(N+1)  
 int\_perturb\_Y=110; % Intervalo entre cada perturbação em Y(N+1)  
 Perturb\_X=0; % Determina se ha perturbação em X0/ 0-nao 1-sim  
 ValorP\_X=[30]; % ValorP\_Y = Valor percentual da perturbação em X0  
 tperturb\_X=0; % Instante no qual ha a perturbação em X0  
 int\_perturb\_X=110; % Intervalo entre cada perturbação em X0  
 Perturb\_L=0; % Determina se ha perturbação em L0/ 0-nao 1-sim  
 ValorP\_L=[-10]; % ValorP\_L = Valor percentual da perturbação em L0  
 tperturb\_L=0; % Instante no qual ha a perturbação em L0  
 int\_perturb\_L=110; % Intervalo entre cada perturbação em L0  
 Y1sp=200; % Valor desejado para Y1 (set point)(\*1e6)  
 erro\_percentual\_max=5; % Porcentagem maxima de erro toleravel (valor absoluto)  
 f\_PI=0; % Valor inicial da fração de L0 proveniente do controlador PI

```

kc=22000;           % Parametro kc do controlador PI
tal_i=10400;       % Parametro tal_i do controlador PI
%tal_d=60;         % Parametro tal_d do controlador PID
R=0;               % Determina se ha ruido / 0-nao 1-sim
F=0;               % Determina se ha Filtro / 0-nao 1-sim
Control_cont=1;    % Determina se o sistema sera controlado (1) ou nao (0)
NA=15;             % Numero de instantes de amostragem anteriores a k

% Vetor de parametros do filtro
alfa=[0.20 0.10 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05];

%#####
% Etapa 02 – Dados para normalização das variáveis
minp=[197.9679; 197.9679; 197.9679; 96.1991; 0.009];
maxp=[201.9447; 201.9447; 201.9447; 100.1769; 0.011];
minL0 = 96.1991;
maxL0 = 100.1769;

%#####
% Etapa 03 – Dados para a implementação do Ruído
Rmin=[-1 -1*(1e-6)]; % Ruído minimo
Rmax=[1 1*(1e-6)];  % Ruído maximo
rad=0;               % Início da função seno, base do ruído (radianos)

%#####
% Etapa 04 – Calculo dos valores iniciais de M(I), X(I), MX(I) e Y(I)
Y(1)=Y1;
X(1)=Y(1)/MM;
for i=2:N
    Y(i)=4.12791E-5 + 1.5872*X(i-1);
    X(i)=Y(i)/MM;
end
Y(N+1)=4.12791E-5 + 1.5872*X(N);

```

```

for i=1:N
    L(i)=L0;                % L(i) = Vazao do liquido em cada prato
    RO=0.01713*X(i) + 0.05551*(1-X(i));
                            % RO = Massa especifica molar media da mistura
    M(i)=AP*RO*(((L(i)/(RO*LW))^0.666667)*C+HV);
end
for i=1:N                  % Valores iniciais de MX
    MX(i)=M(i)*X(i);
end

%#####

% Etapa 05 – Dados pré-laço while
TEMPO=0;                  % Tempo inicial
t_print=0;                % Valor inicial do tempo para impressao em arquivo ascii
TA = 0;                   % Valor inicial do tempo de amostragem
ISE=0;                   % Valor inicial da integral do quadrado do erro
Q_solvente=0;            % Valor inicial da quantidade gasta de solvente
k=1;                     % Contador do vetor de dados para arquivo em ascii
Y_PRINT(k)= Y(1)*1e6;    % Impressao de Y1 inicial para arquivo em ascii (valor
                          inicial)
L0_PRINT(k)=L0;          % Impressao de L0 inicial para arquivo em ascii
T_PRINT(k)=TEMPO;       % Impressao de TEMPO inicial para arquivo em ascii (valor
                          inicial)
L0_atual = L0;          % Valor inicial da variavel de entrada da rede L0|k
SP(k)=200;              % Set point para arquivo em ascii
k=k+1;                  % Atualizacao do contador do vetor de dados
kk=1;                   % Valor inicial do contador do vetor dos dados filtrados
TA_print(1)=1;          % Valor inicial do Tempo de Amostragem para impressao
pt_Y=1;                 % Contador do vetor pertubacao ValorP_Y
pt_X=1;                 % Contador do vetor pertubacao ValorP_X
pt_L=1;                 % Contador do vetor pertubacao ValorP_L
Controle=0;             % Inicialmente o sistema esta desativado, somente e ativado
                          quando TA e atingido

```

```

erro=0; % Valor Inicial do erro
erro_antigo=0; % Valor Inicial do erro em um instante anterior
int_e =0; % Valor inicial da Integral do erro
der_e=0; % Valor inicial da Derivada do erro
L0_PI=L0; % Valor inicial da vazao de solvente proveniente do
controlador PI

PI_cont = 0; % Contador do controlador PI
Y_atual=Y(1)*1e6; % Valor inicial da variavel de entrada da rede Y1|k
YN_atual=Y(N+1); % Valor inicial da variavel de entrada da rede Y(N+1)|k
for i=1:NA
    Y_defasado(i)=Y_atual;
    YN_defasado(i)=YN_atual;
end

while (TEMPO <= TEMPOFINAL)
    % Inicio do laço While

%#####
% Etapa 06 – Calculo dos valores de Y(i)
for i=1:N
    Y(i)=MM*X(i);
end
if TA >= 3
    for i=NA:-1:2
        Y_defasado(i)=Y_defasado(i-1);
        YN_defasado(i)=YN_defasado(i-1);
    end
    Y_defasado(1) = Y_atual;
    YN_defasado(1) = YN_atual;
    Y_atual = Y(1)* 1e6; L0_atual=L0; YN_atual=Y(N+1);
    % Atualização das demais variaveis de entrada

    if Control_cont == 1
        Controle = 1;
    end
end

```

```

End

%#####
% Etapa 07 – Implementação do ruído
    if R == 1
        Ruído;
    end

%#####
% Etapa 08 – Implementação do filtro
    if F == 1
        Filtro;
    end
Y_atualprint(kk)=Y_atual;
YN_atualprint(kk)=YN_atual;
    if kk >= 2
        TA_print(kk)=TA_print(kk-1)+1;
    end
kk=kk+1;

%#####
% Etapa 09 – Atualização dos dados para o controlador PID
if R == 0                % Caso R=0, então F=0
    Y1_PID = Y(1);
end
erro_antigo= erro;
erro= Y1_PID-(Y1sp*1e-6);
erro_percentual = ((erro)/(Y1sp*1e-6))*100;
                    % Porcentagem de erro na variavel controlada
int_e= int_e + erro*TA;
der_e = (erro-erro_antigo)/TA;
var_e = (erro-erro_antigo);
TA=0;

```

end

*#####*

*% Etapa 10 – Calculo dos valores de L(i)*

for i=1:N

RO=0.01713\*X(i) + 0.05551\*(1-X(i));

L(i)=RO\*LW\*(((M(i)/(AP\*RO))-HV)/C)^1.5);

end

*#####*

*% Etapa 11 – Impressao dos Resultados*

if t\_print >= 1 *% Dados para impressao em arquivo ascii*

Y\_PRINT(k)= Y(1)\*1e6; T\_PRINT(k)=TEMPO; SP(k)=200; L0\_PRINT(k)=L0;

k=k+1;

t\_print=0;

end

*#####*

*% Etapa 12 – Controle RNA*

if Controle == 1

Z=[Y\_atual; Y\_defasado(1); Y1sp; L0\_atual; YN\_atual];

*% Normalização das variaveis*

for i= 1:5

Zn(i,1)= 2\*(Z(i,1)-minp(i,1))/(maxp(i,1)-minp(i,1)) - 1;

end

*% Rede\_Coluna\_de\_Absorcao\_Simulacao*

SR= sim(net,Zn); *%Simulação da rede com os novos dados de entrada*

*% Desnormalização das variaveis*

*% Definição da vazao de solvente proveniente da RNA*

L0\_RNA = (((SR+1)\*(maxL0-minL0))/2)+minL0;

```

%#####
% Etapa 13 – Controlador PID
if abs(erro_percentual) >= erro_percentual_max
    f_PI=1;          % Ateracao da fração de L0 proveniente do controlador PI
end

                                % Definição da vazao de solvente proveniente do
                                controlador PID
L0_PI= L0_PI + kc*erro + (kc/tal_i)*int_e + (kc*tal_d)*der_e;

%#####
% Etapa 14 – Definição da ação de controle
L0 = (L0_RNA*(1-f_PI))+(L0_PI*f_PI);
Controle=0;
end

%#####
% Etapa 14 – Euller
Euller;

%#####
% Etapa 15 – Inserção das Perturbações em Y(N+1) e/ou X0 e/ou L0
if Perturb_Y == 1
    if TEMPO >= tperturb_Y
        Y(N+1)=Y(N+1)*(1+(ValorP_Y(pt_Y)/100));
        tperturb_Y= tperturb_Y + int_perturb_Y;
        pt_Y=pt_Y + 1;
        if pt_Y == 2
            tperturb_Y=TEMPOFINAL+10*DT;
        end
    end
end
end
if Perturb_X == 1

```

```

if TEMPO >= tperturb_X
    X0 = X0*(1+(ValorP_X(pt_X)/100));
    tperturb_X= tperturb_X + int_perturb_X;
    pt_X=pt_X + 1;
    if pt_X == 2
        tperturb_X=TEMPOFINAL+10*DT;
    end
end
end
if Perturb_L == 1
    if TEMPO >= tperturb_L
        L0=L0*(1+(ValorP_L(pt_L)/100));
        tperturb_L= tperturb_L + int_perturb_L;
        pt_L=pt_L + 1;
        if pt_L == 2
            tperturb_L=TEMPOFINAL+10*DT;
        end
    end
end
end

%#####
% Etapa 16 – Impressão do erro
erro_print= Y(1)-(Y1sp*1e-6);
erro_percentual_print = ((erro_print)/(Y1sp*1e-6))*100;

%#####
% Etapa 17 – Cálculo dos parâmetros de comparação
ISE=ISE + (erro_print^2)*DT;
Q_solvente= Q_solvente + L0*DT;

TEMPO = TEMPO + DT
t_print = t_print + DT;
TA = TA + DT;

```

```
end                                     % Fim do laço WHILE
```

```
%#####
```

```
% Etapa 18 – Impressão dos resultados
```

```
T_PRINT= T_PRINT'; Y_PRINT= Y_PRINT'; SP = SP'; L0_PRINT = L0_PRINT';
```

```
Y_atualprint=Y_atualprint' ; YN_atualprint=YN_atualprint' ; TA_print = TA_print' ;
```

```
W=[T_PRINT Y_PRINT SP L0_PRINT];
```

```
WW=[TA_print Y_atualprint YN_atualprint];
```

```
save SaidaW.dat W /ascii; % Impressao em arquivo ascii
```

```
save SaidaWW.dat WW /ascii;
```

**ANEXO 02 – PROGRAMA COMPUTACIONAL EM MATLAB® -  
REDE\_COLUNA\_DE\_ABSORCAO\_TREINAMENTO.M**

```

%#####
% Etapa 01 – Dados para treinamento da rede
DTR=load('BCD_Treinamento5.dat');
entrada=DTR(:,1:5);      %Dados de entrada da rede
saida=DTR(:,6);         %Dados de saida da rede
entrada = entrada';
saida = saida';

%#####
% Etapa 02 – Normalização das variáveis
[entradan, minentrada, maxentrada, saidan, minsaida, maxsaida] = premnmx(entrada,
saida);

%#####
% Etapa 03 – Arquitetura da rede
net.numinputs = size(entrada,5);
                                %Numero de variáveis de entrada
net.numLayers = 2;              %Numero de camadas, sem considerar a camada de entrada
net = newff(minmax(entradan),[9,1],{'tansig','purelin'},'trainbr');

%#####
% Etapa 04 – Treinamento da rede
net.trainParam.epochs = 5000;
                                %Numero de passos
net.trainParam.show = 25;      %Atualiza no grafico o numero de passos
net.trainParam.goal = 1e-6;    %Convergencia desejada
net.initFcn = 'initlay';       %Funcao que inicia os pesos e bias
net.performFcn = 'sse';        %Funcao objetivo a ser minimizada
net.trainParam.min_grad= 1e-100;
                                %Minimo gradiente
net.trainParam.mu_max = 1e+100;
                                %Max MU
net = init(net);                %Inicialização da rede

```

```

[net,tr] = train(net, entradan, saidan);
                %Realiza o treinamento (determina pesos)

%#####

% Etapa 05 – Simulação utilizando-se a rede
% DVR = Dados para Validação da Rede
DVR= load('BCD_Validacao.dat');
                %Leitura dos dados de teste
in= DVR(:,1:5);    %Dados de entrada
out= DVR(:,6);    %Dados de saída
in= in';
out= out';

%#####

% Etapa 06 – Normalização das variáveis
minp=[197.9679; 197.9679; 197.9679; 96.1991; 0.009];
maxp=[201.9447; 201.9447; 201.9447; 100.1769; 0.011];
minL0 = 96.1991;
maxL0 = 100.1769;
[liin,colin]=size(in);
[liout,colout]=size(out);
for j=1:colin
for i= 1:liin
    inn(i,j)= 2*(in(i,j)-minp(i,1))/(maxp(i,1)-minp(i,1)) - 1;
end
end
for j=1:colout
for i= 1:liout
    outn(i,j)= 2*(out(i,j)-minL0)/(maxL0-minL0) - 1;
end
end
% SR= Simulacao da Rede
% DSR= Dados provenientes da Simulação da Rede

```

```

SR= sim(net,inn);          %Simulação da rede com os novos dados de entrada
DSR = (((SR+1)*(maxL0-minL0))/2)+minL0;%Desnormalização
figure(1);
plot(out,':')            % plota a variavel de saida.
hold on
plot(DSR,'r');          % plota a variavel de saida fornecida pela rede.
xlabel('eixo x')
ylabel('eixo y')
hold off
figure(2);
[m,b,r]= postreg(out,DSR);
for i=1:488
    erro_sse(i) = out(1,i)-DSR(1,i);
end
sse_validacao=sse(erro_sse)

```

## **ANEXO 03 – ARQUIVOS .M SECUNDÁRIOS EM MATLAB®**

### **% Arquivo Derivadas.m**

```
% Calculo das derivadas para a utilização em Euler
global ELN1 ELN EXN1 EXN EYN1 EYN G DerM DerMX
DerM(i)= ELN1 - ELN;
DerMX(i)= (ELN1*EXN1) - (ELN*EXN) + (G*(EYN1 - EYN));
```

### **% Arquivo Definicao.m**

```
%Definição dos valores de X e Y utilizados em Runge_Kutta
global N L0 X0 ELN1 ELN EXN1 EXN EYN1 EYN X Y L
if i==1
    ELN1=L0; ELN=L(i); EXN1=X0;
    EXN=X(i); EYN1=Y(i+1); EYN=Y(i);
else
    ELN1=L(i-1); ELN=L(i); EXN1=X(i-1);
    EXN=X(i); EYN1=Y(i+1); EYN=Y(i);
end
```

### **% Arquivo Euler.m**

```
% Aplicação do metodo de Euler
global N L0 X0 ELN1 ELN EXN1 EXN EYN1 EYN X Y L DerM DerMX G MX M
for i=1:N
    Definicao;           %Define os valores de ELN1 ELN EXN1 EXN EYN1 EYN
    Derivadas;          %Calcula as derivadas DerM e DerMX
    M(i)=M(i) + (DerM(i)*DT);
    MX(i)=MX(i) + (DerMX(i)*DT);
    X(i)=MX(i)/M(i);
End
```

### **% Arquivo Ruido.m**

```
global rad TEMPO DT Rmax Rmin Y Ruido_Y_defasado Y1_PI
R1=rand(1,2);
for i=1:2
    Rn(1,i)=R1(1,i)*sin(rad);
```

```

Rd(1,i) = (((Rn(1,i)+1)*(Rmax(1,i)-Rmin(1,i)))/2)+Rmin(1,i);
end
Rd_print(kk)=Rd(1,1);
Y_atual=Y_atual+(Rd(1,1));
YN_atual=YN_atual+(Rd(1,2));
Y1_PI=Y_atual*(1e-6);      %Y(1)+(Rd(1,1)*1e-6);
rad = rad + 60*DT;

```

### **% Arquivo Filtro.m**

```

global Y_atual YN_atual Y1_PI Y_defasado YN_defasado alfa
Y_defasado_incremento=0;
YN_defasado_incremento=0;
Y_atual= alfa(1)*Y_atual;
YN_atual= alfa(1)*YN_atual;
for i=1:NA
Y_defasado_incremento = Y_defasado_incremento + alfa(i+1)*Y_defasado(i);
YN_defasado_incremento = YN_defasado_incremento + alfa(i+1)*YN_defasado(i);
end
Y_atual= Y_atual + Y_defasado_incremento;
YN_atual= YN_atual + YN_defasado_incremento;
Y1_PID=Y_atual*(1e-6);

```