

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA QUÍMICA
ÁREA DE CONCENTRAÇÃO: SISTEMAS DE PROCESSOS
QUÍMICOS E INFORMÁTICA

**CONTROLE EM DESTILAÇÃO BATELADA:
CONTROLE ADAPTATIVO E CONTROLE PREDITIVO COM
MODELO BASEADO EM REDES NEURAS ARTIFICIAIS.**

Ana Maria Frattini Fileti

Orientador : Prof. Dr. João Alexandre F. Rocha Pereira

Tese apresentada à Faculdade de Engenharia Química
como parte dos requisitos exigidos para obtenção
do título de Doutor em Engenharia Química.

Campinas - Outubro/1995

0.m.00078982-6

UNIDADE	BC
N.º CHAVE	7/UNICAMP
	FUNCO
V	
TÍTULO	26 023
PREÇO	433,95
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 13,00
DATA	02/11/95
N.º CPD	

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

F474c

Fileti, Ana Maria Frattini

Controle em destilação batelada: controle adaptativo e controle preditivo com modelo baseado em redes neurais artificiais. / Ana Maria Frattini Fileti.--Campinas, SP: [s.n.], 1995.

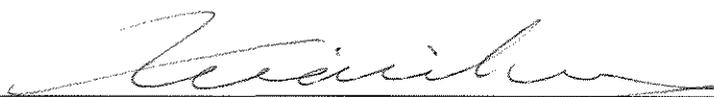
Orientador: João Alexandre F. Rocha Pereira.
Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Química.

1. Destilação. 2. Controle de processo. 3. Inteligência artificial. 4. * Identificação de processos. I. Pereira, João Alexandre F. Rocha. II. Universidade Estadual de Campinas. Faculdade de Engenharia Química. III. Título.

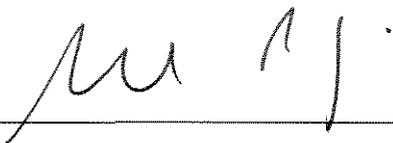
Tese aprovada em 18 de outubro de 1995 pela Banca Examinadora, constituída pelos seguintes professores:



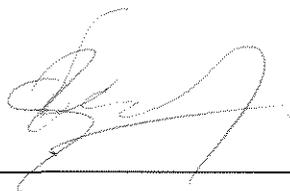
Prof. Dr. João Alexandre Ferreira da Rocha Pereira - Professor Titular
(Orientador)



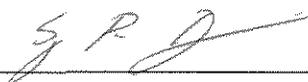
Prof. Dr. Mário de Jesus Mendes - Professor Titular



Prof. Dr. Fernando Augusto Pinto Garcia - Professor Associado



Prof. Dr. Mauro Antônio da Silva Sá Ravagnani - Professor Doutor



Prof. Dr. Sergio Persio Ravagnani - Professor Livre Docente

Esta versão corresponde à redação final da Tese de Doutorado defendida pela aluna Ana Maria Frattini Fileti, e aprovada pela Banca Examinadora em 18 de outubro de 1995.



Prof. Dr. João Alexandre Ferreira da Rocha Pereira

Ao Maurílio,

pela pessoa especial que é e pela
importância de sua presença ao meu lado.

AGRADECIMENTOS

Ao Prof. Dr. João Alexandre F. R. Pereira, pela confiança, pela orientação e por tantas lições de vida transmitidas nestes anos de convivência.

Aos meus pais, Amaury e Vitória, pelos constantes incentivos na busca de meus objetivos profissionais.

Ao Prof. Dr. Mário de Jesus Mendes pelas críticas construtivas realizadas durante o desenvolvimento deste trabalho.

À Profa. Dra. Sandra Lúcia da Cruz pela atenção dispensada na orientação do Estágio de Capacitação Docente, realizado no segundo semestre de 1993, em paralelo à Tese de Doutorado.

Ao Prof. Dr. Roger Zemp e aos colegas da FEQ, em especial à Maria de Lourdes, à Mônica e ao Frede, pelas discussões técnicas constantes nas áreas de estudo envolvidas neste trabalho.

Ao CNPq, pelo auxílio financeiro que permitiu a realização deste trabalho.

*“Por mais que na batalha se vença um ou mais inimigos,
a vitória sobre si mesmo é a maior de todas as vitórias.”*

Siddhartha Gautama.

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO	1
1.1 - A IMPORTÂNCIA DA DESTILAÇÃO EM BATELADA	2
1.2 - MOTIVAÇÃO PELO ESTUDO DO TEMA	2
1.3 - ORGANIZAÇÃO DA TESE	4
CAPÍTULO 2 - COLUNAS DE DESTILAÇÃO EM BATELADA. ESTADO DO CONHECIMENTO EM CONTROLE	6
2.1 - INTRODUÇÃO	7
2.2 - APRESENTAÇÃO DO EQUIPAMENTO	7
2.3 - OPERAÇÃO	9
2.4 - MODELAGEM E SIMULAÇÃO	11
2.5 - ESTADO DO CONHECIMENTO EM CONTROLE	15
2.5.1 - Operação a Refluxo Constante	15
2.5.2 - Operação a Composição Constante	15
2.5.3 - Operação com Planejamento Ótimo	19
2.6 - ANÁLISE DA LITERATURA E PROPOSTA DA TESE	24
CAPÍTULO 3 - CONTROLE ADAPTATIVO	29
3.1 - INTRODUÇÃO	30
3.2 - HISTÓRICO	30
3.2.1 - Controlador Adaptativo Programável (PAC)	32
3.2.2 - Controlador Auto-Ajustável (STR)	34
3.3 - IMPLEMENTAÇÃO DO PAC	36
3.4 - IMPLEMENTAÇÃO DO STR	37
3.4.1 - Identificação em Linha do Processo	39
3.4.2 - Projeto do Controlador	41

3.5 - RESULTADOS	46
3.5.1 - Caso 1	49
3.5.2 - Caso 2	55
3.5.3 - Caso 3	58
3.5.4 - Caso 4	61
3.6 - ANÁLISE	66

CAPÍTULO 4 - CONTROLE PREDITIVO COM MODELO BASEADO EM REDES NEURAS ARTIFICIAIS

68

4.1 - INTRODUÇÃO	69
4.2 - REDES NEURAS	69
4.2.1 - A Rede Neural Biológica	69
4.2.2 - A Essência da Rede Neural Artificial (RNA)	71
4.2.3 - A Computação Neural	74
4.2.4 - Topologia da Rede Neural Artificial	76
4.2.5 - Treinamento da RNA	77
4.2.5.1 - Algoritmo de Treinamento Regra de Delta Generalizada (GDR)	79
4.3 - RNA'S EM CONTROLE DE PROCESSOS - UM BREVE HISTÓRICO	85
4.4 - CONTROLE PREDITIVO COM MODELO	91
4.4.1 - Conceito	91
4.4.2 - Histórico	93
4.5 - CONTROLE PREDITIVO COM MODELO RNA	95
4.5.1 - Implementação	102
4.5.1.1 - Método de Powell	106
4.6 - RESULTADOS E DISCUSSÃO	106
4.6.1 - Exemplo 1	115
4.6.2 - Exemplo 2	126
4.7 - ANÁLISE	129

CAPÍTULO 5 - CONCLUSÕES E SUGESTÕES	130
5.1 - CONCLUSÕES	130
5.2 - SUGESTÕES	133
ANEXO 1 - SIMULADOR DA COLUNA DE DESTILAÇÃO MULTICOMPONENTE EM BATELADA	135
ANEXO 2 - OBTENÇÃO DO MODELO DISCRETO DO PROCESSO VIA TRANSFORMADA-Z	155
ANEXO 3 - OBTENÇÃO ANALÍTICA DO GRADIENTE DA FUNÇÃO-OBJETIVO	159
ANEXO 4 - OBTENÇÃO ANALÍTICA DO JACOBIANO DA FUNÇÃO-OBJETIVO	163
ANEXO 5 - IMPLEMENTAÇÃO DA IDENTIFICAÇÃO DE PROCESSOS POR ANÁLISE REGRESSIVA	168
ANEXO 6 - IMPLEMENTAÇÃO DO ALGORITMO DE TREINAMENTO GDR	178
ANEXO 7 - IMPLEMENTAÇÃO DO MÉTODO DE OTIMIZAÇÃO DE POWELL	184
REFERÊNCIAS BIBLIOGRÁFICAS	192
ABSTRACT	201

NOMENCLATURA

A_p = Área do prato;

A = Vetor-entrada do neurônio artificial;

a_1, b_1 e b_2 = Coeficientes ajustáveis da equação polinomial discreta do processo (Cap. 3);

a_i = Sinal de entrada i do neurônio artificial (Cap. 4);

b_j = Saída calculada do neurônio j (Cap. 4);

c_k = Saída calculada da Rede Neural Artificial;

E = Erro entre variável controlada e valor de referência;

E_j = Eficiência de Murphree no estágio j ;

e = Constante da Equação de Francis;

D = Fluxo de Destilado;

$D(z)$ = Função transferência do controlador (Cap. 3);

d_k = Saída conhecida da Rede Neural Artificial;

dic = Discrepância entre valores amostrado e predito da variável controlada;

f_k = Gradiente da função-objetivo P (Cap. 3);

$f()$ = Forma funcional de cálculo interno do neurônio artificial (Cap. 4);

$f(x)$ = Função minimizada pelo Método de Powell (Cap. 4);

$f_i^0 = \phi_i^s P_i^s$ = Fugacidade de referência do componente i puro à temperatura e pressão de equilíbrio (Capítulo 2);

$G_p(s)$ = Função transferência do processo;

$G_H(s)$ = Função transferência do “data-hold”;

h_v = Altura do Vertedouro;

h_j = Entalpia total molar da fase líquida no estágio j ;

H_j = Entalpia total molar da fase vapor no estágio j ;

$(\Delta H)_o$ = Calor Latente de Vaporização da mistura presente em V_1 ;

$(\Delta H)_{N+1}$ = Calor Latente de Vaporização da mistura presente no refeededor;

I = Vetor-entrada da Rede Neural Artificial (Cap. 4);

I = Número de amostragens passadas usadas na Análise Regressiva (Cap. 3);

J = Função-objetivo utilizada no MPC;

K_c = Ganho da Lei de Controle PI;

K_p = Ganho do processo;

L_w = Altura do Vertedouro;

L_j = Fluxo de Líquido que deixa o estágio j ;

L_n = Número de neurônios artificiais da camada de entrada da RNA;

M_j = Acúmulo Molar de Líquido no estágio j ;

M_n = Número de neurônios artificiais da camada oculta da RNA;

N_c = Horizonte de controle;

N_p = Horizonte de predição;

N_n = Número de neurônios da camada de saída da RNA;

P = Pressão de Operação (Cap. 2);

P = Função-objetivo (Cap. 3);

P = Número de pares de vetores entrada/saída usados para treinamento da RNA (Cap. 4);

P_i^s = Pressão de saturação calculada pela Equação de Antoine;

Q_c = Fluxo de calor retirado no condensador;

Q_R = Fluxo de calor fornecido ao refeedor;

q_i = Fator restritivo das ações de controle no MPC;

sp = Valor de referência da variável controlada;

s_n^k = Direção de busca do Método de Powell;

t = Tempo de operação;

t_d = Tempo morto do processo;

T = Intervalo de amostragem;

T_f = Temperatura de fundo da coluna;

Th = Valor residual de atividade do neurônio artificial ("Threshold");

u = Variável controlada;

u' = Valor em otimização da variável manipulada;

V_j = Vetor fator ponderal do neurônio j (Cap. 4);

$v_{i,j}$ = Peso da conexão do neurônio i sobre o neurônio j ;

V_j = Fluxo de Vapor que deixa o estágio j (Cap. 2);

W_k = Vetor fator ponderal do neurônio k ;

$w_{j,k}$ = Peso da conexão do neurônio j sobre o neurônio k ;

x = Somatório ponderado de entrada do neurônio acrescido de seu valor residual de atividade (Cap. 4);

x_n^k = Ponto de busca no Método de Powell;

X = Vetor de parâmetros do processo;

x_j = Raízes da função gradiente (Cap. 3);

$x_{i,j}$ = Fração Molar do componente i na fase líquida do estágio j (Cap. 2);

x_B = Fração Molar de fundo;

$y_{i,j}$ = Fração Molar do componente i na fase vapor do estágio j (Cap. 2);

$y_{i,j}^*$ = Fração Molar de Equilíbrio do componente i na fase vapor do estágio j (Cap. 2);

y = Variável controlada (composição de topo);

\tilde{y} = Variável controlada amostrada;

\hat{y} = Variável controlada predita (Cap. 4).

Letras Gregas

α = Coeficiente de “momentum” no GDR;

δ = Termo de declínio do gradiente no GDR;

ε = Erro quadrático total entre os valores conhecidos e calculados de saída da RNA;

ϕ_i = Coeficiente de Fugacidade do componente i (Eqs. Termodinâmica Básica + Eq. Virial).

γ_i = Coeficiente de Atividade do componente i (UNIQUAC);

η = Taxa de aprendizado no GDR;

λ = Passo ótimo do Método de Powell;

μ = Constante de tempo no Algoritmo de Dahlin;

θ = Atraso de transporte no Algoritmo de Dahlin;

ρ_j = Massa Específica Molar da mistura líquida no prato j ;

τ_I = Constante de tempo integral da Lei de Controle PI;

τ_p = Constante de tempo do processo.

SIMBOLOGIA

- Ⓐ Analisador Cromatográfico
- Ⓒ Inferenciador de Composição
- ⒸⒸ Controlador de Composição
- ⒹⒹ Medição de Diferencial de Pressão
- ⒹⒸⒸ Controlador de Diferencial de Pressão
- ⒻⒸ Controlador de Fluxo
- ⒻⒹ Transmissor de Fluxo
- ⒻⒸ Controlador de Nível
- ⒻⒹ Transmissor de Pressão
- Ⓓ Ganho do Controlador de Razão
- ⒹⒹ Transmissor de Temperatura

RESUMO

Processos de separação são de relevante importância na indústria química, sendo a destilação em batelada um destes processos mais antigos. Apesar de ter sido considerado ultrapassado pelo surgimento das colunas de destilação contínuas, o processo em batelada tem recebido atenção especial nos últimos anos em função do interesse crescente na área que se destina à produção de substâncias químicas ou bioquímicas de alta tecnologia e alto valor agregado, e em problemas ambientais de redução de volumes residuais.

Apesar de envolverem princípios fenomenológicos similares que lhes conferem características de processos não-lineares, a falta de alimentação contínua no equipamento em batelada torna os perfis de temperatura e composição instantâneos ao longo da coluna, exigindo sistemas de controle substancialmente diferentes. Diante das constantes variações nas características do processo (regime transiente), desenvolveram-se projetos de sistemas de controle digitais que se desviassem da sintonia e “design” fixos dos algoritmos convencionais de controle por retroalimentação, para controle de composição de topo através da manipulação da razão de refluxo. Visando economia de energia, uma malha independente, de controle de aquecimento na base da coluna, opera através de medições de composição de fundo ou diferencial de pressão ao longo da coluna.

Inicialmente estudou-se uma das maneiras mais simples de se efetuarem adaptações nos parâmetros ajustáveis do sistema de controle digital: as adaptações programadas (“Programmable Adaptive Control”). Estas são programadas previamente em função do comportamento de uma variável auxiliar que represente bem o estado atual processo, dependendo fortemente do conhecimento das propriedades do material a ser destilado e dos aspectos fenomenológicos que descrevem o processo.

A complexidade das misturas normalmente processadas na prática exige, entretanto, o projeto e a implementação de controladores que possam efetuar a identificação do processo “on-line”, independente das informações referidas anteriormente. Com esta finalidade e ainda na classe dos Controladores Adaptativos, utilizou-se um Controlador Auto-ajustável (“Self-tuning Regulator”), onde a equação polinomial discreta que descreve o processo tem seus coeficientes ajustados através de Análise Regressiva, e diferentes leis de controle puderam ser implementadas, como PI (Forma Posição ou Velocidade) e a derivada do Algoritmo de Dahlin.

Procurando por uma representação também não-fenomenológica, porém não-linear, implementou-se ainda um Controlador Preditivo com Modelo baseado em Redes Neurais Artificiais, que, sendo treinadas a cada intervalo de amostragem, fornecem previsões de comportamento do processo em determinado horizonte de tempo. A seleção das variáveis de estado mais importantes do processo define o número de neurônios artificiais utilizados nas camadas de entrada e saída da rede (“Perceptron”). Este último tipo de controlador, além de possuir as características genéricas anteriores, possibilita a operação não somente à composição constante, mas também sujeita à trajetória ótima preestabelecida de composição de topo.

Através de simulações do processo em malha fechada para diferentes misturas ternárias, fez-se uma análise dos principais resultados obtidos com os controladores propostos, onde se constatou a superioridade de representação do processo através da aproximação por Rede Neural Artificial, treinada por Retropropagação, e mais efetiva ação de controle proveniente do Controlador Preditivo na manutenção de composição constante no topo e economia de energia.

CAPÍTULO 1 - INTRODUÇÃO.

1.1 A IMPORTÂNCIA DA DESTILAÇÃO EM BATELADA

Processos de separação são de relevante importância na indústria química, sendo a destilação em batelada um destes processos mais antigos. Apesar de ter sido considerado ultrapassado pelo surgimento das colunas de destilação contínuas, o processo em batelada tem recebido atenção especial nos últimos anos em função do interesse crescente na área que se destina à produção de substâncias químicas ou bioquímicas de alta tecnologia e alto valor agregado.

Além destas aplicações em Química Fina, justificadas pelas pequenas quantidades manipuladas de material, a mesma técnica da destilação em batelada tem sido utilizada em problemas ambientais de redução de volumes residuais e recuperação de materiais. Como exemplo, verifica-se a larga utilização do referido processo na recuperação de solventes, sendo normalmente aplicado a resíduos líquidos industriais com concentrações de substâncias orgânicas voláteis maiores que 10% em peso. Ainda se o material a ser processado contém sólidos ou substâncias que possam se tornar altamente viscosas quando concentradas, a destilação em batelada pode ser normalmente utilizada já que estes sólidos ou fluidos viscosos permanecem no recipiente de carregamento no fundo da coluna e podem ser removidos sem problemas ao final da destilação.

Além destas, outras vantagens da destilação em batelada sobre a destilação contínua podem ser enumeradas. A flexibilidade oferecida é uma das principais, ou seja, uma única coluna em batelada pode ser usada no processamento de diversas misturas sob diferentes condições operacionais. Mais importante ainda é o número de produtos que podem ser obtidos em uma única operação em batelada. Enquanto a operação em batelada permite que “n” componentes sejam obtidos como produtos, para a separação contínua desta mesma mistura de “n” componentes seriam necessárias “n-1” colunas interconectadas.

1.2 MOTIVAÇÃO PELO ESTUDO DO TEMA

Apesar de envolverem princípios fenomenológicos similares que lhes conferem características de processos não-lineares, as colunas contínuas e em batelada

operam em regimes diferentes. A falta de uma alimentação contínua no equipamento em batelada torna os perfis de temperatura e composição instantâneos ao longo da coluna, caracterizando regime transiente de operação.

Sendo objeto de estudo freqüentemente relatado em literatura, o projeto de sistemas de controle digital para colunas contínuas visa superar as dificuldades inerentes às não-linearidades do processo e procura pela retirada fixa de produtos de topo e também de fundo. Já o projeto de controle para colunas em batelada deve ser substancialmente diferente. Neste caso, além das mesmas não-linearidades, o regime transiente e o objetivo de retirada de diferentes produtos, com pureza especificada, no topo, exigem tratamento bastante diferenciado, com implementação de sistemas de controle no mínimo adaptativos.

Trazendo inúmeros trabalhos relacionados à simulação dinâmica e otimização de projeto e operação de colunas em batelada, ao contrário da destilação contínua, a literatura falha no estudo de projeto e implementação de sistemas de controle digitais essenciais para a manutenção da operação nas condições desejadas, otimizadas ou não.

Deste modo e acoplado à intenção de implementação futura em equipamento real pertencente ao Departamento de Engenharia de Sistemas Químicos da Faculdade de Engenharia Química da UNICAMP, a realização de um estudo detalhado da aplicação de técnicas de controle avançado ao equipamento em questão mostrou-se ser um objetivo bem definido e justificado.

Quanto à escolha das técnicas de controle a serem utilizadas (Controle Adaptativo ou Preditivo baseado em Redes Neurais Artificiais), esta justifica-se no item de análise de literatura e proposta da tese presente no capítulo dois. Esclarece-se apenas aqui que a evolução tecnológica atual dos computadores digitais permite que algoritmos de controle cada vez mais complexos sejam implementados praticamente sem restrições de tempo de amostragem a ser utilizado. Envolvendo dois algoritmos de otimização “on-line” a cada amostragem - um para treinamento da Rede Neural utilizada e outro para otimização das ações de controle - o Controle Preditivo proposto, sendo o mais complexo de todos implementados no presente trabalho, pode ser perfeitamente implementado para os propósitos definidos em computador do tipo PC486 DX2 de 66 MHz.

1.3 ORGANIZAÇÃO DA TESE

No presente capítulo, procurou-se mostrar a relevância dos processos de destilação em batelada na indústria química e quais as principais motivações envolvidas no estudo do tema proposto.

No capítulo dois, o equipamento em estudo será apresentado esquematicamente e seus diversos tipos de operação detalhados com o objetivo de clarear a problemática envolvida no projeto de sistemas de controle. Para que esta melhor se estabeleça no contexto literário existente, uma revisão bibliográfica será apresentada, onde serão discutidos os esquemas de controle implementados pelos autores citados. A revisão de literatura para os controladores propostos no presente trabalho é deixada para os capítulos subsequentes, já que nenhuma aplicação destes foi encontrada para colunas de destilação em batelada. Finalmente, a análise dos trabalhos existentes, juntamente com o desejo de desenvolvimento de sistemas de controle mais genéricos justificam, no último item do capítulo dois, a proposta da tese.

No terceiro capítulo, em face ao problema de controle a ser resolvido, onde se verificam variações constantes nas características do processo devido à descontinuidade e às não-linearidades intrínsecas, propõem-se dois tipos de Controladores Adaptativos: o Adaptativo Programável (PAC) e o Auto-ajustável (STR). Neste segundo tipo, de implementação mais genérica que o primeiro, serão utilizadas leis diferentes de controle, como a PI digital e a resultante do Algoritmo de Dahlin. O referido capítulo traz um histórico literário sobre os controladores adaptativos, assim como a descrição detalhada do projeto de cada um deles juntamente com os métodos matemáticos necessários e os resultados obtidos para diferentes sistemas binários e ternários sendo processados.

O quarto capítulo trata da implementação do Controle Preditivo Não-linear com Modelo (MPC) baseado em Redes Neurais Artificiais (RNA's). A crescente popularidade do MPC e o aumento do número de publicações na área de Controle de Processos que fazem uso desta ferramenta computacional que imita o comportamento inteligente do cérebro humano - as RNA's - são relatados em históricos independentes. Ainda no capítulo quatro são apresentados detalhadamente a computação neural, as

diferentes topologias existentes de RNA's, o procedimento de treinamento das mesmas, além da implementação do controlador por último referido e os resultados obtidos. Uma análise final dos resultados encontrados relaciona alguns conselhos práticos para implementação do mesmo.

No capítulo cinco, são apresentadas as conclusões finais do estudo e colocadas algumas sugestões para futura investigação.

**CAPÍTULO 2 - COLUNAS DE DESTILAÇÃO EM BATELADA.
ESTADO DO CONHECIMENTO EM CONTROLE.**

2.1 INTRODUÇÃO

No presente capítulo, o equipamento em estudo será apresentado esquematicamente e seus diversos tipos de operação detalhados com o objetivo de clarear a problemática envolvida no projeto de sistemas de controle.

Apresentam-se também as principais equações utilizadas na modelagem rigorosa da destilação em batelada pois, para a realização de um estudo detalhado da viabilidade de aplicação das técnicas de controle propostas, utilizou-se um programa de simulação do processo, com vistas à implementação futura em equipamento real.

Para melhor estabelecer o problema a ser resolvido no contexto literário existente, uma revisão bibliográfica é apresentada, onde são discutidos os esquemas de controle implementados pelos autores citados. A revisão de literatura para os controladores propostos no presente trabalho é deixada para os capítulos subsequentes já que nenhuma aplicação destes foi encontrada para colunas de destilação em batelada.

A análise dos trabalhos existentes, juntamente com o desejo de desenvolvimento de sistemas de controle mais genéricos justificam, no último item deste capítulo, a proposta da tese.

2.2 APRESENTAÇÃO DO EQUIPAMENTO

A destilação em batelada é governada pelos mesmos princípios de equilíbrio líquido-vapor, hidráulica dos pratos e dinâmica das colunas contínuas, diferenciando-se apenas pela falta de continuidade na alimentação da mesma, gerando um comportamento intrinsecamente transitório.

Destilações comerciais deste tipo normalmente sofrem carregamentos de misturas líquidas binárias ou multicomponentes no refeedor, que podem ou não ser continuamente coletados no topo após se submeterem aos estágios de equilíbrio presentes no corpo da coluna.

Uma coluna convencional de destilação em batelada é esquematizada na figura 2.1, onde verifica-se ser constituída de:

- refulvedor, onde normalmente a mistura a ser separada é colocada e onde se fornece o calor necessário à separação através de serpentinas de vapor ou resistências elétricas;
- corpo de uma coluna de retificação, contendo pratos perfurados ou recheio para promover contato íntimo entre as duas fases (líquida e vapor);
- condensador, total ou parcial, do vapor de topo da coluna;
- tanques de estocagem para coleta das diferentes frações obtidas como produto.

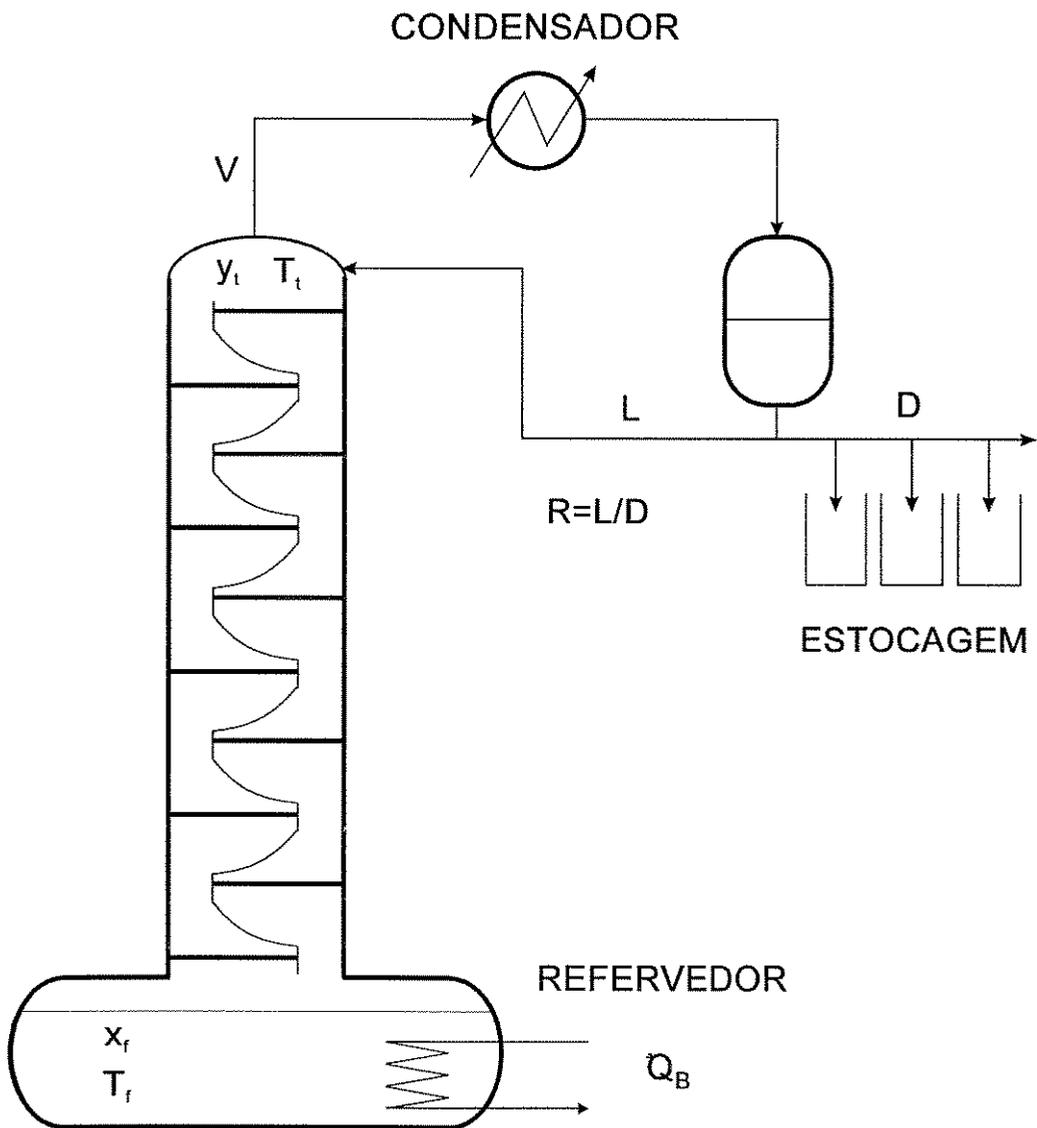


Figura 2.1 - Representação esquemática de uma coluna de destilação em batelada.

2.3 OPERAÇÃO

O equipamento detalhado no item anterior tem duas etapas diferentes de funcionamento, embora ambas de características transientes: a partida (“start-up”) e a produção propriamente dita.

Durante a partida, o material a ser processado é carregado no refeedor onde é aquecido (calor sensível) até atingir a temperatura de bolha da mistura. Continuando a receber calor pelo sistema de aquecimento, frações da mistura são evaporadas (calor latente), em fluxo ascendente ao corpo da coluna. Para a formação dos acúmulos (“hold up”) nos pratos, a coluna deve operar em condições de refluxo total até atingir um estado estacionário onde sejam estabelecidos os perfis de temperatura e composição. Como alternativa a este procedimento, pode-se alimentar pelo topo a mesma mistura líquida do refeedor em ponto de bolha ou operar a coluna com refluxo nulo até que se acumule líquido suficiente no condensador para estabelecimento dos acúmulos nos pratos e então passa-se a refluxo total para atingir o equilíbrio.

Sendo, em geral, o equipamento de destilação em batelada um equipamento multiproduto, o procedimento de partida deve ser escolhido de acordo com a estrutura disponível e o sistema a ser destilado. Esta etapa não constitui objeto de estudo no presente trabalho.

Após o estabelecimento dos perfis iniciais de temperatura e composição sob condição de refluxo total, altera-se o mesmo de forma a iniciar a retirada de produto. Este, rico no componente mais volátil da mistura alimentada, começa então a ser coletado à saída do condensador e sua qualidade depende fortemente da quantidade de material condensado que retorna à coluna. A composição deste produto de topo coletado tende a acompanhar as variações de composição do material presente na coluna graças à descontinuidade de alimentação no equipamento.

Na etapa de produção - que é o verdadeiro objeto de estudo neste trabalho - a operação pode ser conduzida de três maneiras distintas:

- Refluxo constante, que é o modo operacional mais simples. Neste, a fração molar do componente de pureza especificada pode começar mais elevada que a desejada e

terminar abaixo deste valor, porém o produto final acumulado deverá estar dentro de especificação;

- Composição constante, onde a razão de refluxo deverá variar de modo a enriquecer o produto à medida que o componente desejado for se esgotando. Razão de refluxo elevada sempre eleva a concentração do componente mais volátil presente, mas este pode não ser o componente de maior prioridade;

- Combinação das duas maneira convencionais de operação, onde um planejamento de refluxo ótimo deve ser seguido de forma a satisfazer um critério ótimo de desempenho: produção máxima dentro de especificação em tempo mínimo, tempo mínimo para produção de determinada quantidade de produto dentro de especificação, ou lucro máximo.

Para colunas em que nenhuma instrumentação é disponível, existe ainda a possibilidade da operação cíclica, onde são alternados períodos com refluxo total e refluxo nulo.

Em todos os modos operacionais citados acima, pode-se ainda optar por variação ou não do calor fornecido à mistura presente no refeedor. Se a configuração do refeedor não permitir que sejam efetuadas variações na quantidade de calor injetada na coluna, pode-se operar a mesma com taxa de injeção de calor fixa em valor que supere a entalpia de vaporização da mistura do refeedor mesmo quando estiverem presentes somente os componentes menos voláteis. Deste modo, o fluxo de vapor que deixa o refeedor começa alto (componentes mais voláteis) e decresce à medida em que a temperatura do ponto de bolha da mistura multicomponente se eleva no refeedor. Com fluxo de vapor alto pode ser necessária elevada razão de refluxo para purificação do destilado, caracterizando desperdício de energia.

Havendo possibilidade de variação na potência de aquecimento, consegue-se manter um fluxo fixo de vapor que deixa o refeedor desde que se eleve a taxa de calor injetada para acompanhar a variação de composição sofrida no refeedor.

Acoplando as vantagens da destilação em batelada e da destilação extrativa, Lang *et al.* (1994) propõem a destilação extrativa em batelada sob condições de refluxo constante. Misturas azeotrópicas ou de baixa volatilidade relativa são separadas pela introdução de um solvente no sistema, da seguinte maneira:

1. Operação sob refluxo total sem alimentação de solvente;
2. Operação sob refluxo total com alimentação de solvente para diminuir a concentração do componente menos volátil no destilado;
3. Operação sob razão de refluxo finita fixa com alimentação de solvente para produção de componente mais volátil;
4. Operação sob razão de refluxo finita fixa sem alimentação de solvente para separação do componente menos volátil do solvente.

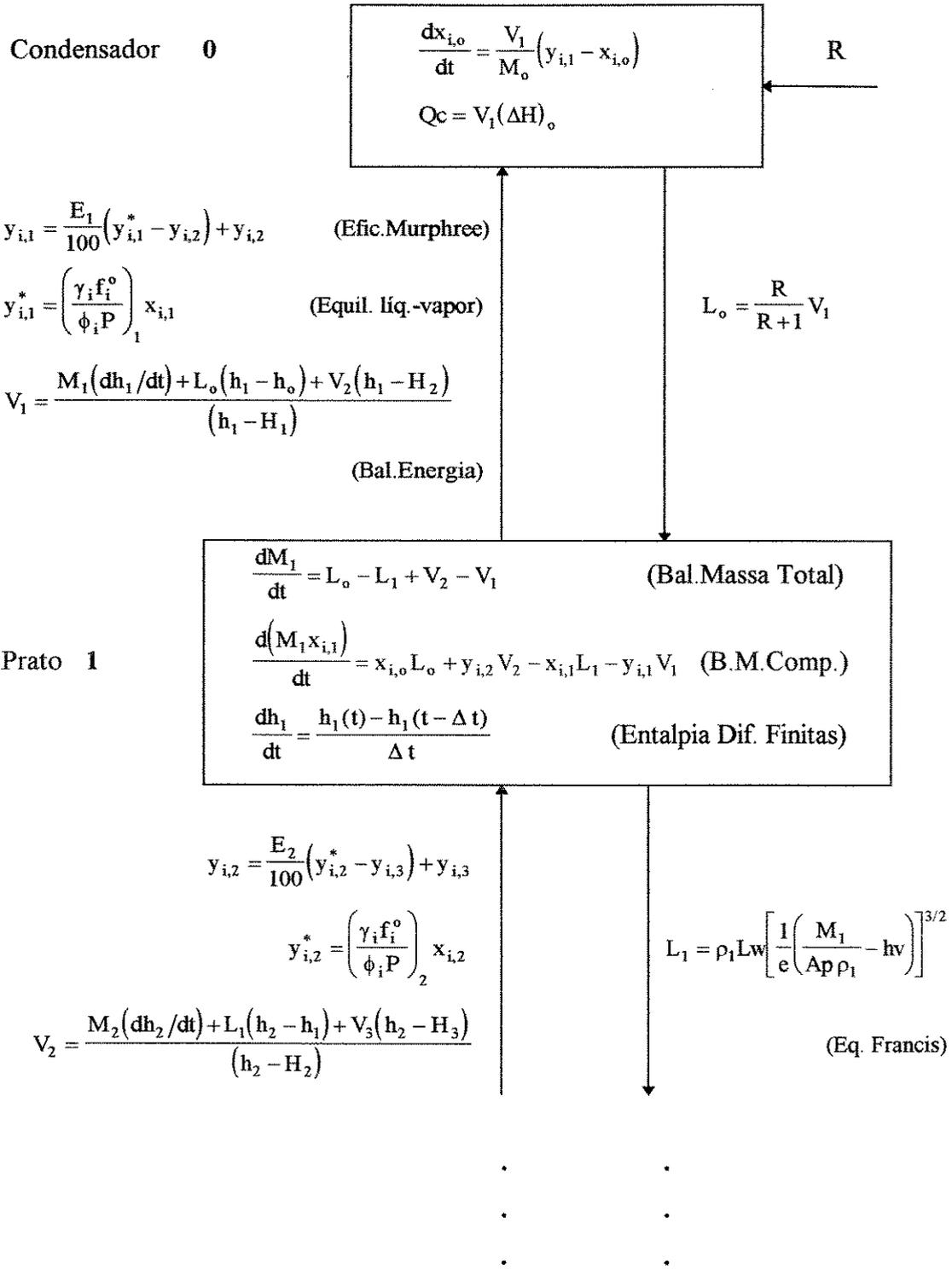
Outro uso industrial da destilação em batelada relatado em literatura (Reuter *et al.*, 1989; Cuille *et al.*, 1986) é a montagem de uma coluna de retificação sobre um reator tanque, ou destilação batelada com reação química. Este tipo de operação é utilizado quando o produto principal da reação é mais volátil que os outros produtos e os próprios reagentes. Quando a reação é reversível, a remoção deste produto (conseqüente diminuição na concentração do mesmo na fase líquida) facilita a reação direta. Se for irreversível, a remoção do produto principal (de temperatura de ebulição menor) promove o aumento de temperatura no reator, podendo aumentar a velocidade de reação.

2.4 MODELAGEM E SIMULAÇÃO

A realização de um estudo detalhado da aplicação de técnicas de controle avançado no equipamento em estudo é mais facilmente feita através de um simulador do processo. Objetivando a análise dos sistemas de controle em estudo no presente trabalho para futura implementação em equipamento real pertencente ao Departamento de Engenharia de Sistemas Químicos da Faculdade de Engenharia Química da UNICAMP, utilizou-se o programa computacional de simulação COLBAT, por nós desenvolvido como parte de pesquisa realizada em tese de mestrado (Fileti, 1992), onde se encontram descrição detalhada e validação do modelo por comparação com dados experimentais obtidos por Domenech *et al.* (1974).

Apresentam-se aqui apenas as principais suposições feitas e o equacionamento básico (fig. 2.2) do programa computacional COLBAT que se encontra no ANEXO 1^o deste trabalho:

- acúmulo de vapor desprezível;
- perda de carga desprezível ao longo da coluna;
- eficiência de separação constante nos pratos;
- condensador total;
- refeedor considerado estágio ideal.



(continua)

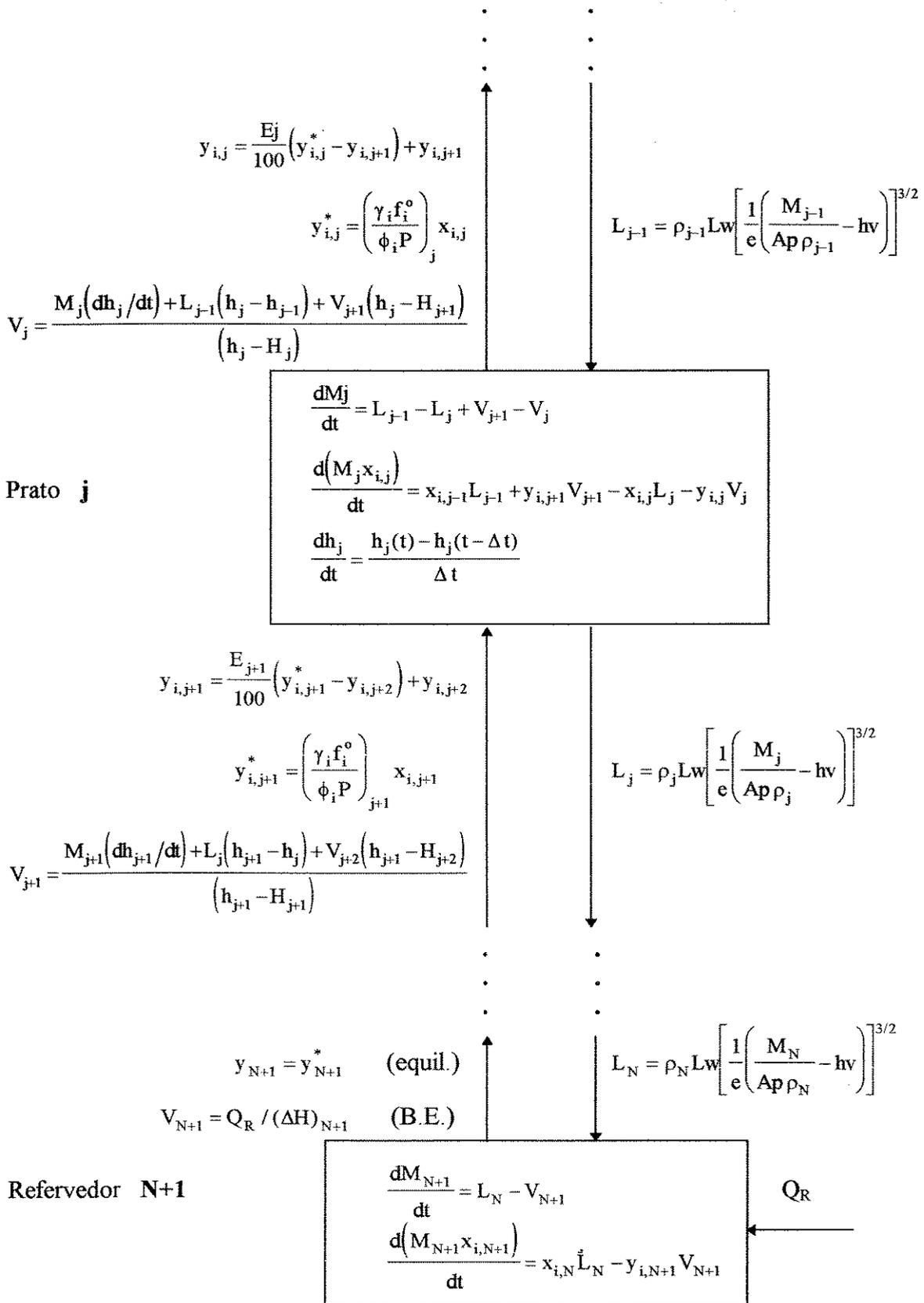


Fig. 2.2 - Modelo rigoroso de destilação multicomponente em batelada.

2.5 ESTADO DO CONHECIMENTO EM CONTROLE

Segundo Abrams *et al.* (1987), o maior problema relacionado com o processo de destilação em batelada é a dificuldade de operar o equipamento de modo a alcançar objetivos preestabelecidos ou controlar o processo.

Sendo o desenvolvimento de sistemas de controle para automação de colunas de destilação multicomponente em batelada o principal objetivo do presente estudo, resumem-se aqui os principais trabalhos encontrados em literatura com o mesmo propósito e para os diferentes modos operacionais citados anteriormente.

2.5.1 Operação a Refluxo Constante

O modo de operação convencional mais simples - refluxo constante - é o que necessita sistema de controle menos elaborado, porém não-trivial já que o processo é transiente e portanto mesmo um PI ou PID, para controle de composição de topo, têm que ter seus parâmetros ajustados durante a operação. A configuração de controle e instrumentação utilizados para manter o refluxo constante estão representados na figura 2.3, presente entre outras mais complexas no artigo de Bozenhardt (1987).

As desvantagens da operação a refluxo constante são relacionadas por Shinsky (1979) e constatadas em nossa pesquisa de mestrado (Fileti, 1992):

- se a relação D/V for relativamente alta, a separação será pobre e a retirada do destilado deverá ser interrompida mesmo com uma concentração grande do componente de interesse no refeedor;
- se a relação D/V for reduzida para melhorar a separação, a destilação pode consumir uma quantidade absurda de energia.

2.5.2 Operação a Composição Constante

Neste modo de operação mais eficiente, onde o refluxo deve variar constantemente para acompanhar as variações de composição de fundo da coluna, o sistema de controle torna-se mais complexo.

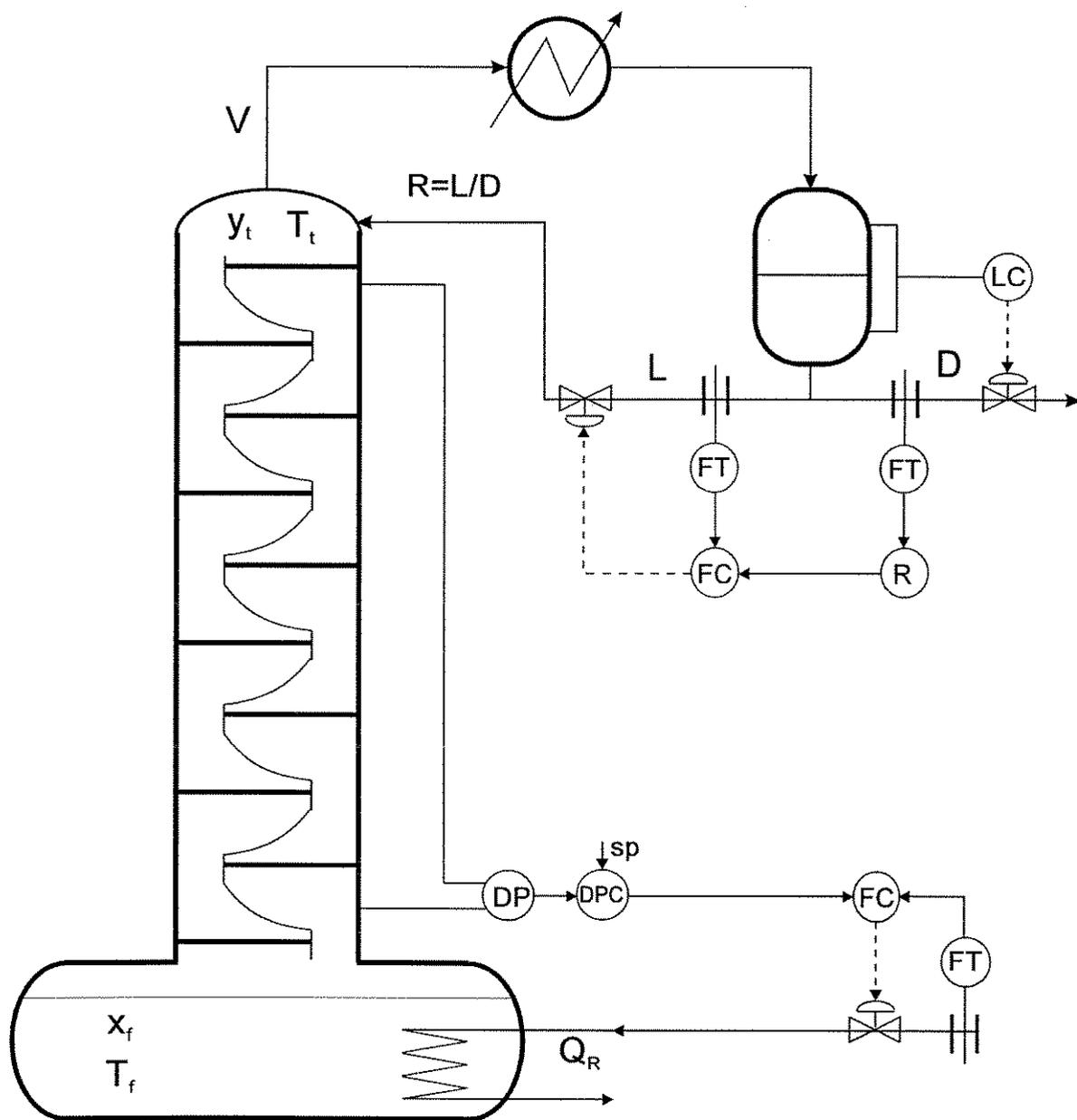


Fig. 2.3 - Configuração de controle e instrumentação no modo de operação a refluxo constante.

Se o componente prioritário for o mais volátil, a razão de refluxo (L/D) deve permanecer baixa no começo da batelada e ser gradualmente aumentada pelo sistema de controle à medida em que o componente for se esgotando na coluna.

Separações multicomponentes são também efetuadas sem dificuldades no equipamento de destilação em batelada desde que recipientes diferentes para coleta de destilado sejam acionados automaticamente ou manualmente após um aviso do sistema de controle. Neste instante, o valor de referência ("set point") de composição de topo também é reajustado, assim como os parâmetros ajustáveis do controlador utilizado.

A figura 2.4 (Robinson, 1985; Shinskey, 1979) apresenta a configuração de controle e instrumentação mínimas necessárias para o modo de operação a refluxo variável. Neste esquema, as duas referências citadas assumem os controladores como sendo do tipo convencional por retroalimentação (P, PI ou PID), sendo que a primeira faz uso da aproximação digital dos mesmos para manutenção de composição de topo constante, além de chamar a atenção para a constante possibilidade de saturação do controlador já que seus parâmetros são ajustados para operação em regime estacionário.

Em todos os modos de operação citados, uma segunda malha de controle é representada no fundo da coluna. Conforme descrito no item 2.3, que trata da operação da coluna, à medida em que os componentes mais voláteis se esgotam a temperatura do ponto de bolha do líquido presente no refeedor aumenta e uma maior quantidade de calor deve ser suprida ao mesmo. Em implementações práticas, onde normalmente não existe cromatógrafo em linha para medições de composição de fundo, medições do diferencial de pressão ao longo da coluna constituem guias sensíveis de indicação do fluxo de vapor ascendente e conseqüentemente da taxa de calor requerida no refeedor.

Em relação à não existência de cromatógrafo em linha para medição de composição de topo, medições de temperatura e pressão, juntamente com dados de equilíbrio líquido-vapor, geralmente são suficientes para inferência da mesma e ainda com a vantagem de não apresentar atrasos inerentes às análises cromatográficas. Num dos modelos de inferência propostos por Quintero-Marmol e Luyben (1992), considerou-se que na etapa de produção os perfis de composição na parte superior da coluna são essencialmente binários, independente do número de componentes da mistura inicial. Assim sendo, este tipo de inferência não se restringe apenas a separações binárias.

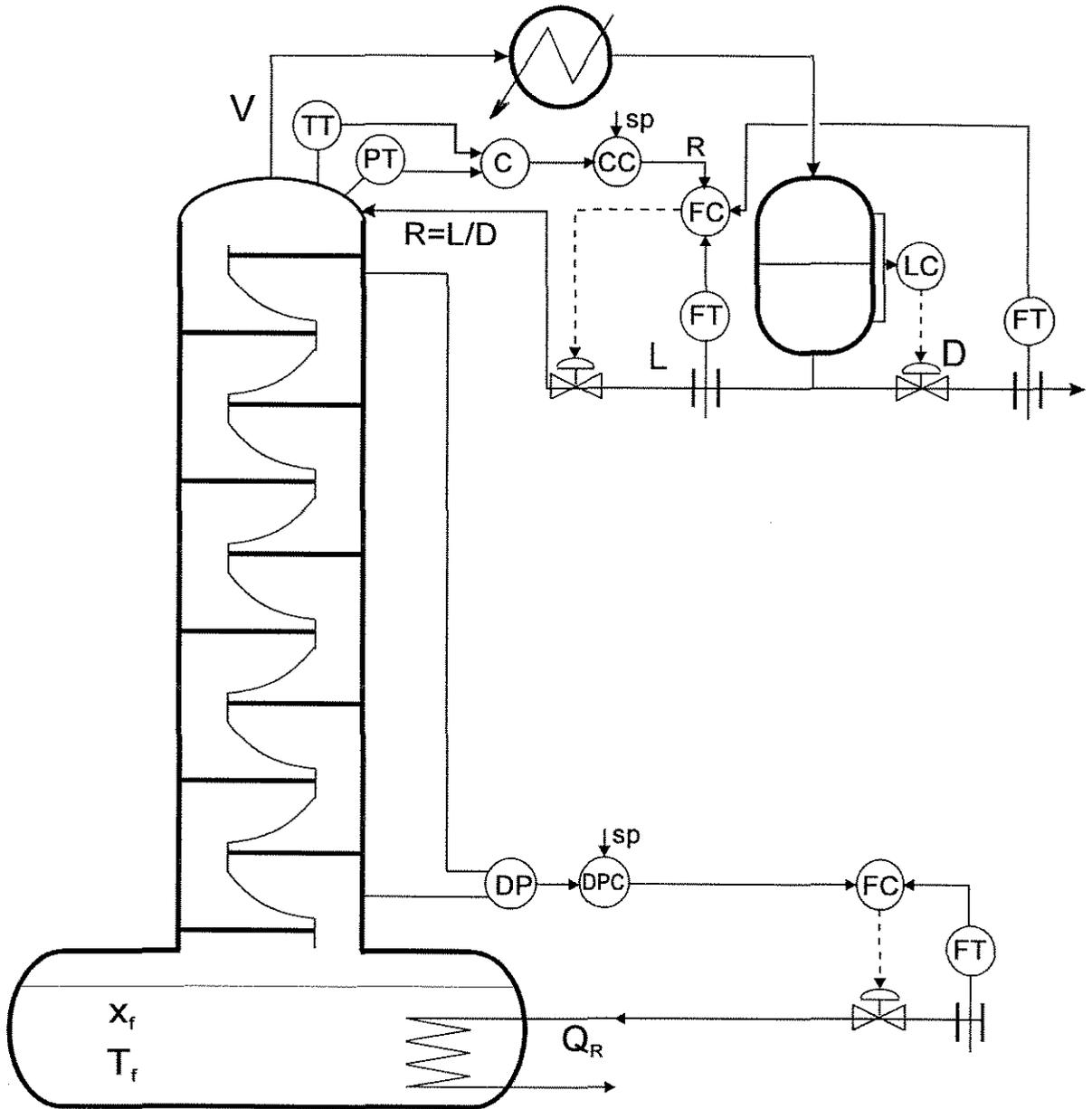


Fig. 2.4 - Configuração mínima de controle e instrumentação para operação a refluxo variável.

Como alternativa não-convencional de inferenciação de variáveis controladas a partir de medições de variáveis secundárias, Willis et al. (1992) sugerem a utilização de uma aproximação não-linear através de uma Rede Neural Artificial devidamente treinada com dados de análise prévia de amostras retiradas do processo em malha aberta.

Deixando estas duas propostas de inferenciação como sugestões para o próximo passo a ser tomado no sentido da implementação prática, considera-se no presente trabalho uma inferenciação perfeita de composição de topo e portanto sem tempo morto associado à medição da mesma.

Ainda no modo de operação à composição constante, encontra-se no trabalho de Lueke *et al.* (1985) uma aplicação isolada de sistema de Controle Preditivo Linear do tipo DMC (Controle por Matriz Dinâmica), ao invés dos algoritmos convencionais, a uma coluna real de recheio em batelada. O objetivo seria a obtenção de etanol com 85 a 90% em peso no topo a partir de uma mistura inicial de 10% de etanol e 90% de metanol. Segundo os autores, a operação à composição constante apresentaria, no caso estudado, aproximadamente ótimo consumo de energia. A composição de topo calculava-se através de medições de temperatura e pressão no topo e correlação empírica tirada de dados de operação em malha aberta. Por ser o processo não-linear e o modelo do controlador linear, a matriz de ganhos era ajustada manualmente quando se percebiam oscilações nas variáveis manipulada ou controlada. Tipicamente eram realizadas de 5 a 8 ajustes de ganhos durante 7 horas de operação. Levando-se em conta que este caso era o de recuperação de apenas um componente de uma mistura binária (caso simples), provavelmente seria exigida atenção redobrada do operador durante uma separação multicomponente.

2.5.3 Operação com Planejamento Ótimo

Como citado anteriormente no item 2.3, um terceiro modo operacional existe de modo a satisfazer um critério ótimo de desempenho. O objetivo é obter um planejamento ótimo para a razão de refluxo através da otimização de uma função objetivo sujeita a restrições do processo, usando-se normalmente para este fim um modelo dinâmico

do processo para simulação do mesmo. Esta otimização é geralmente realizada “off-line”, mas tendo-se um modelo reduzido que agilize os cálculos, pode também ser utilizada em paralelo à planta. Os critérios ótimos mais utilizados são: produção máxima dentro de especificação em tempo mínimo, tempo mínimo para produção de determinada quantidade de produto dentro de especificação, e lucro máximo. Quanto às restrições, estas podem ser de natureza operacional - como, por exemplo, limitações reais de fluxos máximos e mínimos manipulados, limitações reais dos elementos finais de controle quanto ao tamanho das variações das variáveis manipuladas - ou imposições, como quantidade ou pureza do produto em determinado momento do processo.

Converse e Gross (1963) estudaram a maximização da quantidade total de destilado de composição especificada, recolhido em tempo estipulado, através da aplicação do Princípio de Máximo Contínuo de Pontryagin e de programação dinâmica aplicados ao modelo dinâmico do processo em estudo.

Murty *et al.* (1980) usaram o mesmo modelo simplificado utilizado por Converse e Gross e compararam os resultados obtidos através de diferentes técnicas de otimização, sendo este trabalho restrito a sistemas binários e com pequeno número de pratos.

Ainda usando o mesmo modelo simplificado anterior, Coward (1967a,b), Robinson (1969, 1970, 1971) e Mayur e Jackson (1971) ao invés de maximizarem a produção, minimizaram o tempo de destilação necessário para a produção de determinada quantidade de destilado de composição especificada.

Robinson (1969) impôs, além da razão de refluxo ótima, um aumento linear do fluxo de vapor que deixa o refeedor (“boil up”). O tempo de batelada necessário foi bastante reduzido em relação à operação a refluxo constante. Para solução deste problema de tempo mínimo foi utilizado um método de procura inclinação acima. Misturas multicomponentes também foram estudadas neste trabalho, porém tratadas efetivamente como sistemas binários.

A maioria dos trabalhos de otimização citados anteriormente consideraram o acúmulo molar de líquido no refeedor como sendo dominante, desprezando assim o acúmulo de líquido nos pratos.

Robinson (1970) concluiu que acúmulos nos pratos, mesmo que pequenos, não podem ser desprezados na modelagem da destilação em batelada, mostrando seu efeito numa coluna industrial utilizada para separação de terpenos. Neste último trabalho e no de Mayur e Jackson (1971), sistemas ternários foram estudados. Entretanto, os próprios autores expõem as dificuldades de extensão do método de otimização a sistemas que envolvam mais de três componentes, já que o aumento deste número implica na solução de um número maior de equações. Outro problema associado com a formulação, discutem os autores, é que o número de equações também aumenta com o número de pratos, requisitando mais memória e tempo computacional.

Diwekar *et al.* (1987) propuseram, como solução para o aumento de tempo computacional pelas razões acima citadas, a utilização da extensão de um método “short-cut” usado para colunas contínuas. Ao invés de utilizar o tempo como variável explícita, os autores rescreveram as equações diferenciais com o componente 1 como componente de referência. Esta formulação, onde a variação de qualquer outro componente pode ser expressa em termos do componente de referência (equações com tempo implícito), tem a vantagem de reduzir a dimensionalidade do problema. O problema de destilado máximo foi então resolvido pelo Princípio de Pontryagin para colunas multicomponentes em batelada, nas quais os acúmulos nos pratos seriam desprezíveis. Porém, em comparação feita pelos autores com dados do artigo de Murty *et al.* (1980), para colunas de pequeno número de pratos (comum neste tipo de equipamento), houve desvios significantes entre os dados obtidos da otimização com os modelos rigoroso e reduzido do processo.

Hansen e Jorgensen (1986) resolveram o problema de tempo mínimo para colunas binárias não só de pratos como de recheio. Nas colunas de estágios, o acúmulo nos pratos era desprezado. As equações do modelo foram discretizadas no tempo utilizando-se para isto técnicas de colocação ortogonal. Segundo os autores, para o problema de tempo mínimo, o fluxo de vapor que deixa o refeedor deveria permanecer em seu valor máximo (otimizando-se apenas a razão de refluxo) em colunas de pratos, enquanto em colunas de recheio as duas variáveis deveriam ser utilizadas para otimização, já que o fluxo ascendente de vapor influencia o coeficiente total de transferência de massa entre as fases.

Usando uma combinação entre os problemas de destilado máximo e tempo mínimo, Kerkhof e Vissers (1978) encontraram um planejamento de refluxo ótimo

maximizando uma função lucro que levava em conta as quantidades produzidas, o tempo de partida, o custo dos produtos e o tempo de batelada.

Trabalhos mais recentes têm se concentrado na otimização e projeto conjuntos de colunas em batelada para processamento de uma determinada mistura particular. Diwekar *et al.* (1989), utilizando a aproximação “short-cut” proposta em Diwekar *et al.* (1987), investigaram o projeto de uma coluna com base numa função objetivo econômica que considerava custos globais, analisando em seqüência seu planejamento de controle ótimo.

No artigo de Logsdon (1990), juntamente com grandes estudiosos da destilação em batelada e da otimização de processos, que são respectivamente Diwekar e Biegler, explorou-se uma estratégia para a otimização simultânea do projeto e operação de colunas de destilação em batelada. Através da aplicação de métodos implícitos de ordem superior (como colocação ortogonal em elementos finitos), obteve-se uma representação algébrica bastante precisa e estável das equações diferenciais, que puderam ser escritas como restrições dentro do problema de otimização. Deste modo, solução e otimização do modelo da coluna ocorrem simultaneamente. Esta aproximação simultânea permite, à formulação do problema, lidar com o projeto e operação integrados, já que os parâmetros de projeto e a política de operação são otimizados ao mesmo tempo através de um algoritmo de programação não-linear. A função-objetivo utilizada foi formulada de modo a maximizar o lucro, definido como o valor do produto menos os custos principais da unidade de destilação em batelada. Além da aplicação fundamental a somente uma mistura particular a ser destilada, foi considerado também o problema de otimização de projeto e operação multiperifódico, onde uma única coluna foi utilizada para separação de diferentes misturas.

Ainda nesta linha de formulação multiperifódica, Mujtaba e Macchietto (1993) trazem um apanhado dos principais trabalhos para otimização da operação do equipamento em estudo. Os autores criticam esforços concentrados apenas na otimização de retirada de produtos individuais, lembrando que numa separação multicomponente produtos diferentes são coletados e portanto a operação tem que ser otimizada como um todo. Assim sendo, propõem um método para determinação da política ótima de operação multiperifódica de uma determinada mistura, com vários produtos de interesse e vários

cortes intermediários fora de especificação. A solução do problema determina a quantidade ótima a ser obtida de cada produto e dos cortes intermediários (malha externa de otimização), e a duração e o planejamento de refluxo ótimo para cada um deles (malha interna de otimização).

Para tornar possível a otimização em paralelo, a maioria dos trabalhos citados utiliza modelo reduzido do processo, normalmente negligenciando os acúmulos molares de líquido nos pratos, para agilizar e diminuir a complexidade de cálculos. Porém, se os acúmulos na prática existirem e forem significantes frente à vazão de refluxo, as constantes de tempo dos pratos assumem valores não desprezíveis, associando característica dinâmica dominante às variáveis de estado em cada estágio. Diante da possibilidade de obtenção de planejamentos que na realidade não sejam ótimos, Robinson (1985) propôs que o planejamento ótimo fosse alcançado a partir de combinações de períodos alternados a refluxo constante e a composição constante, estudados criteriosamente através de simulações prévias. Shinskey (1979) já propusera que o planejamento de refluxo ótimo fosse encontrado por uma simples equação linear relacionando o valor de referência de composição de topo, y_{sp} , com o valor atualizado do fluxo de destilado, D :

$$y_{sp} = kD + y_o \quad (2.1)$$

sendo k (inclinação da curva) e y_o (valor inicial do valor de referência) previamente estabelecidos.

Somente nestas duas últimas citações, que trabalham com planejamento sub-ótimo, verifica-se uma preocupação com os problemas de controle propriamente ditos. Em todos os outros trabalhos de otimização citados, as políticas ótimas não eram implementadas de modo a se obter uma malha de controle de composição de topo, mas sim um sistema bastante simples de controle para que a razão de refluxo seguisse a curva ótima encontrada. Já nos trabalhos de Robinson (1985) e Shinskey (1979), uma instrumentação do tipo da figura 2.4, utilizada no modo operacional à composição constante, é proposta de modo a tornar o problema de controle também supervisorio e não somente regulatório. Neste sistema de controle o planejamento ótimo é imposto sob forma

de trajetória de “set-point” de composição e o algoritmo de controle continua sendo do tipo PI ou PID.

Num dos trabalhos mais recentes referentes à operação de colunas de destilação em batelada, Leversund *et al.* (1994) criticam justamente a falta de estudos em temas associados ao controle e operação em linha do referido processo, enquanto que a modelagem dinâmica e a otimização receberam atenção especial de inúmeros pesquisadores. Neste último trabalho, os autores utilizam mais uma vez o algoritmo convencional PI de controle, constituindo sistema regulatório e supervisor, para implementação em destilação batelada com reação.

2.6 ANÁLISE DA LITERATURA E PROPOSTA DA TESE

Seguindo a mesma linha de crítica, citada no trabalho de Leversund *et al.* (1994), quanto à falta de trabalhos com enfoque à implementação do sistema de controle propriamente dito ao equipamento em estudo, tem-se a impressão de que o ambiente propício da destilação em batelada para estudos complexos de otimização tornaram-na simplesmente um caso-estudo em que se medem a validade e a precisão dos mais variados métodos computacionais desenvolvidos para otimização de processos.

Esquecem-se, entretanto, os verdadeiros problemas que surgem quando se trata a destilação em batelada como problema prático real das indústrias químicas que a utilizam. Em primeiro lugar, verifica-se que com exceção da indústria de química fina, o equipamento de destilação em batelada é escolhido justamente pela flexibilidade oferecida em termos de manipulação das mais variadas e complexas misturas que surgem de um processo químico, sejam como produto principal, subproduto ou resíduo industrial. Tendo esta característica de equipamento multiproduto, não se justifica um projeto otimizado do mesmo e a melhor solução para projeto, se possível, é de construção do equipamento em módulos agrupados de maneira específica para cada mistura, ficando longe de uma especificação otimizada.

Em segundo lugar, pode ser preciso mantusear misturas complexas das quais se conhece apenas a presença de um determinado componente de interesse, sendo os outros desconhecidos. Num estudo de recuperação de compostos orgânicos presentes em resíduos

industriais (Hassan e Timberlake, 1992), os autores enfrentam estes dois problemas relacionados acima. Neste caso, além de um projeto ótimo não poder ser efetuado, não se conhecem devidamente as propriedades das substâncias envolvidas no processo. Assim sendo, a otimização da operação do processo se torna também de difícil execução já que um modelo fenomenológico do processo é necessário para se atingir tal objetivo.

Por último, nos poucos trabalhos em que existe a preocupação da verdadeira implementação do sistema de controle para que este atue de modo regulatório (composição constante) e/ou supervisório (trajetória ótima de composição), verifica-se sempre a utilização dos algoritmos convencionais de controle por retroalimentação: PI ou PID. Apesar da alternativa de utilização destes algoritmos em sua forma velocidade (ver cap.3) para prevenção contra a saturação do termo integral, como sugere Robinson (1985), os parâmetros ajustáveis necessários dificilmente assumem valores constantes aceitáveis. Nenhum trabalho, exceto o de Luecke *et al.* (1985), cita um ajuste sistemático dos parâmetros do controlador. Neste último, o ajuste seria feito manualmente após constatação de oscilações na variável controlada, ou seja, a adaptação era realizada de maneira corretiva e não preventiva. Sendo única exceção também quanto ao algoritmo utilizado (DMC), a utilização deste é severamente criticada por Lundström *et al.* (1995) por ser baseada na suposição de que a planta possa ser representada por um modelo estável obtido da reação a uma perturbação em degrau. Para um bom desempenho, mesmo em processos estacionários, estes últimos autores sugerem que um número excessivo de coeficientes de convolução devam ser utilizados. Estendendo para nosso estudo não estacionário, certamente estes inúmeros coeficientes teriam que ser atualizados em paralelo à planta, tarefa não facilmente realizável.

Diante do que se apresenta em literatura e das limitações discutidas acima, a proposta do presente trabalho é o projeto de sistemas de controle ainda não utilizados em destilação batelada e que fujam da sintonia e “design” fixos dos algoritmos convencionais por retroalimentação, como fazem os controladores avançados do tipo adaptativo e/ou preditivo.

De início propõe-se estudar uma das maneiras mais simples de se efetuarem adaptações nos parâmetros ajustáveis do sistema de controle digital: as adaptações programadas (“Programmable Adaptive Control”). Estas são programadas previamente em

função do comportamento de uma variável auxiliar que represente bem o estado atual processo. Assim como no planejamento de refluxo ótimo, o projeto depende fortemente do conhecimento do material a ser processado e dos aspectos fenomenológicos que descrevem o processo em estudo, já que são necessárias várias simulações do mesmo.

Visando tornar o sistema de controle independente do conhecimento das propriedades do material a ser processado e compensar falhas de planejamento por possível imprecisão do modelo no projeto proposto anteriormente, estudar-se-á em seguida a utilização de controladores auto-ajustáveis (“Self-tuning Regulators”). Nestes, todo trabalho de identificação do processo é realizado recursivamente em tempo real. A planta passa então a ser representada por determinada função de transferência discretizada via transformada-z, tornando o controlador genérico também em relação ao equipamento a ser controlado.

Procurando por uma representação também não-fenomenológica, porém não-linear e auto-ajustável do processo, propõe-se ainda o desenvolvimento de um controlador baseado em técnica de Inteligência Artificial, cuja utilização em controle de processos vem crescendo exponencialmente nos últimos anos: Redes Neurais Artificiais (RNA's). Utilizando as RNA's para fornecerem predições de comportamento da planta em determinado horizonte de tempo, um sistema de Controle Preditivo Não-linear Baseado em Modelo (“Non-linear Model Predictive Control”) se tornou viável para otimização das ações de controle a serem implementadas. Com este último tipo de controlador, além de apresentar as mesmas características genéricas anteriores, também haveria a possibilidade de se operar não somente à composição constante, mas também sujeito à trajetória ótima preestabelecida de composição de topo.

Juntamente com a malha de controle de topo e procurando fornecer somente energia necessária para possibilitar uma taxa de evaporação fixa, acrescenta-se um controlador, por alimentação direta, de potência de aquecimento na base da coluna. Este se baseia em medições de composição de fundo ou diferencial de pressão ao longo da coluna.

Sendo várias as tarefas a serem realizadas por um computador acoplado ao processo para controle em tempo real, enfatiza-se que o objetivo do presente trabalho restringe-se ao desenvolvimento das partes de identificação e controle do processo

propriamente ditas (fig. 2.5) e que os resultados a serem apresentados são baseados exclusivamente em simulações do processo, conforme esclarecido no item 2.4.

Finalmente, para possibilitar a implementação de qualquer um dos sistemas de controle digitais a serem estudados, a instrumentação básica proposta para suportar os três modos operacionais é representada esquematicamente na figura 2.6, considerando-se que o computador digital já seja dotado de uma placa de conversão de sinal analógico-digital-analógico (placa ADA).

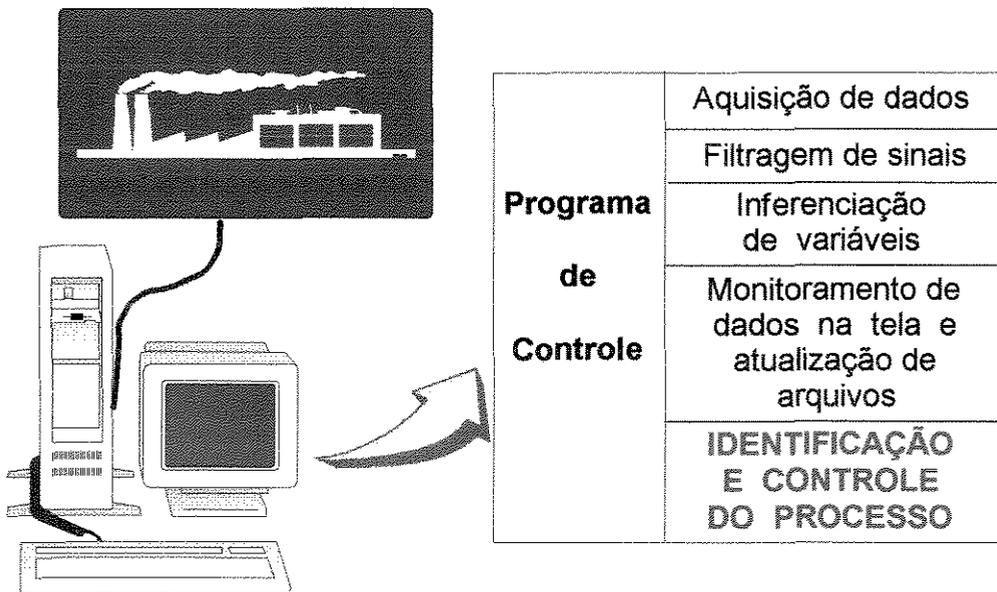


Fig. 2.5 - Tarefas de um computador digital “on-line”. Em destaque a parte a ser desenvolvida no presente trabalho.

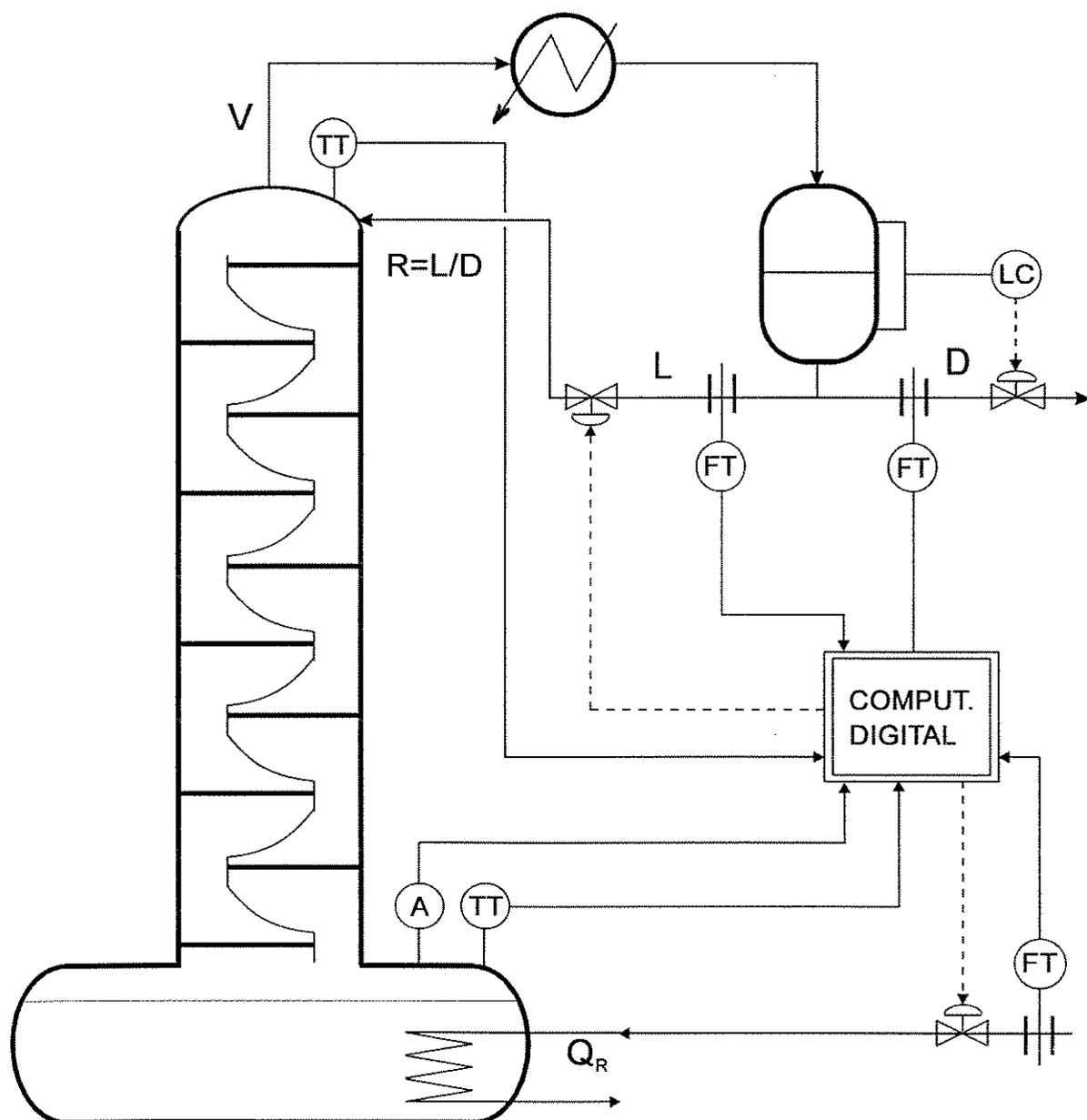


Fig. 2.6 - Configuração básica de instrumentação para os sistemas de controle digital propostos no presente trabalho.

CAPÍTULO 3 - CONTROLE ADAPTATIVO.

3.1 INTRODUÇÃO

De acordo com Aström (1983), o Dicionário Webster traz o significado do verbo adaptar como sendo o equivalente em inglês a “mudar de modo que seu comportamento se ajuste a circunstâncias novas ou modificadas”.

Em face ao problema de controle a ser resolvido, onde se verificam variações constantes nas características do processo devido à descontinuidade e às não-linearidades do mesmo, pode-se observar que o controlador terá de sofrer modificações de modo que se ajuste a estas novas circunstâncias, ou seja, o controlador deve ser adaptativo.

Várias maneiras de se efetuar o ajuste do controlador são sugeridas em literatura como sendo diferentes tipos de controladores adaptativos. Neste capítulo são propostos dois tipos: o Controlador Adaptativo Programável (“Programmable Adaptive Controller” ou PAC) e o Controlador Auto-Ajustável (“Self-tuning Regulator” ou STR). Neste segundo tipo, de implementação mais flexível que o primeiro, procurou-se ainda utilizar diferentes leis de controle, como PI e a resultante do algoritmo de Dahlin, devidamente detalhados no item que trata de suas implementações. Um método de otimização é ainda detalhado junto com a descrição da fundamental parte de identificação do processo.

Por último, são apresentados alguns resultados da implementação das configurações de controle adaptativo propostas ao processo de destilação simulado para diferentes sistemas multicomponentes a serem separados.

3.2 HISTÓRICO

Segundo Aström (1983), as palavras “controle adaptativo” têm sido usadas no mínimo desde o começo da década de 1950. É citada como exemplo uma patente de Caldwell (1950) sobre um regulador adaptativo. No decorrer dos anos, surgiram várias tentativas de se definir o Controle Adaptativo (Truxal, 1964; Sadiris *et al.*, 1973). Intuitivamente, um regulador adaptativo é aquele que pode mudar seu comportamento em resposta a mudanças na dinâmica do processo e suas perturbações. Em seu artigo, Aström

adota a suposição pragmática de que controlador adaptativo é simplesmente um tipo especial de controladores não-lineares por retroalimentação.

Muitas pesquisas na área de controladores adaptativos foram realizadas, no início da década de 50, motivadas pelo objetivo de se projetar pilotos automáticos para aviões de alta performance já que estes operam normalmente em uma faixa ampla de velocidades e altitudes. Sendo de conhecimento que um controlador linear por retroalimentação, com ganho constante, somente funcionaria de maneira adequada ao redor de uma determinada condição de operação, um regulador mais sofisticado, que se adequasse a diversas condições de velocidade e altitude, seria então necessário. Ainda segundo Aström, o trabalho com controladores adaptativos para vôos foi caracterizado por muito entusiasmo, "hardware" ruim e teoria escassa. Uma apresentação de resultados foi apresentada por Gregory (1959) e Mishkin e Braun (1961). Porém, o interesse na área diminuiu devido ao não surgimento de novas idéias e principalmente por causa de um desastre ocorrido num teste de vôo (Taylor e Adkins, 1965).

Na década de 60, várias contribuições à teoria de controle foram desenvolvidas, as quais de grande importância ao desenvolvimento do controle adaptativo. Teorias de estado-espço e de estabilidade foram introduzidas. Surgiram também muitos resultados importantes na teoria do controle estocástico. Contribuindo para o entendimento de processos adaptativos, Bellman (1957, 1961) introduziu a Programação Dinâmica e Feldbaum (1960a,b,1961a,b,1965) a Teoria do Controle Dual. Contribuições fundamentais foram feitas também por Tsytkin (1971) e Aström e Eykhoff (1971) na parte que trata da identificação de sistemas e estimação de parâmetros.

O interesse em controle adaptativo foi renovado na década de 70. O progresso na teoria de controle da década anterior contribuíra para um melhor entendimento do controle adaptativo. O rápido e revolucionário progresso na microeletrônica tornou mais simples e barata a implementação de controladores adaptativos, resultando num vigoroso desenvolvimento no assunto, tanto nas universidades como nas indústrias.

Muitas revisões existem em literatura e uma bibliografia extensa, cobrindo mais de 700 artigos é dada por Asher *et al.* (1976). Alguns livros do início da década de 80 contêm coleção representativa de artigos que mostram diversas aplicações do controle

adaptativo, como Narendra e Monopoli (1980), Unbehauen (1980) e Harris e Billings (1981).

Os diferentes esquemas existentes de controle adaptativo partem sempre de uma malha simples de controle por retroalimentação, com um processo e um regulador com parâmetros ajustáveis. A chave do problema é encontrar uma maneira conveniente de mudar os parâmetros do regulador em resposta a variações no processo. Portanto, os vários esquemas existentes diferem entre si somente na maneira como os parâmetros do controlador são ajustados.

3.2.1 Controlador Adaptativo Programável (PAC)

Uma das maneiras mais simples de se efetuarem as adaptações do controlador é programá-las. Para tanto, em alguns casos é possível encontrar variáveis que correlacionem bem seus valores com as variações da dinâmica do processo. Deste modo, é possível reduzir os efeitos das variações dos parâmetros do processo variando-se os parâmetros do controlador como funções das variáveis auxiliares medidas (figura 3.1).

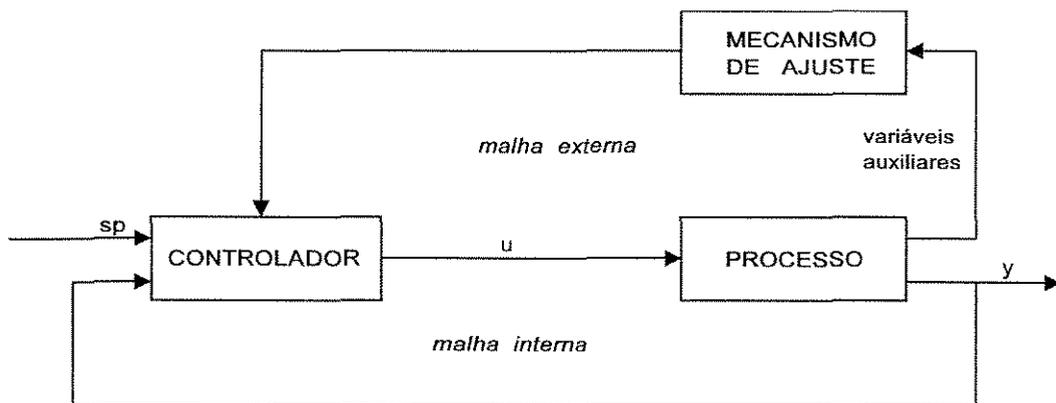


Fig. 3.1 - Diagrama de blocos do Controlador Adaptativo Programável (PAC).

Este esquema é também chamado de Planejamento de Ganho (“Gain Scheduling”) por ser originalmente usado para acomodar variações somente no ganho do processo.

Neste esquema, a chave do problema de projeto do controlador é encontrar variáveis auxiliares adequadas. Isto é normalmente feito baseado no conhecimento físico do sistema. Quando estas forem escolhidas, os parâmetros do controlador poderão ser determinados, em qualquer condição de operação, usando-se método de sintonização adequado. Os estudos de estabilidade e desempenho do sistema são normalmente avaliados através de simulações.

Stephanopoulos (1984) apresenta dois exemplos de aplicação do PAC. No primeiro apresenta um problema somente de planejamento de ganho, onde, frente a possíveis não-linearidades da válvula de controle de uma malha de controle por retroalimentação, mantém-se fixo o ganho total da malha através da modificação do ganho do controlador baseada em medição do posicionamento da haste da válvula (variável auxiliar). No segundo, através de dados experimentais conhecidos e medição de temperatura do ar, o controlador programável ajusta a razão entre combustível e ar de modo a maximizar a eficiência de combustão de uma caldeira submetida a um controlador de razão (“Ratio Control”).

Um dos pontos negativos do PAC é ser uma compensação em malha aberta, ou seja, não há retroalimentação para compensar um possível planejamento inadequado. Portanto, o PAC pode ser visto como um sistema de controle por retroalimentação onde os parâmetros do controlador são ajustados por compensação “feedforward”. Outro ponto negativo é o tempo consumido para projeto do sistema, que tem de ter seus parâmetros determinados para diversas condições de operação do processo, e ter seu desempenho testado através de simulações exaustivas. Em contrapartida, PAC tem a vantagem de modificar os parâmetros do controlador rapidamente em resposta às variações do processo. Os fatores limitantes dependem somente da rapidez com que as variáveis auxiliares respondem às mudanças no processo.

Apesar da controvérsia em nomenclatura em torno de o Controlador Adaptativo Programável ser considerado como um sistema adaptativo ou não, em função

de seus parâmetros serem ajustados em malha aberta, PAC é uma técnica bastante útil de se reduzirem os efeitos das variações de parâmetros de processo.

3.2.2 Controlador Auto-Ajustável (STR)

O segundo método de ajuste dos parâmetros utilizado é o auto-ajustável. Este sistema é mostrado na figura 3.2.

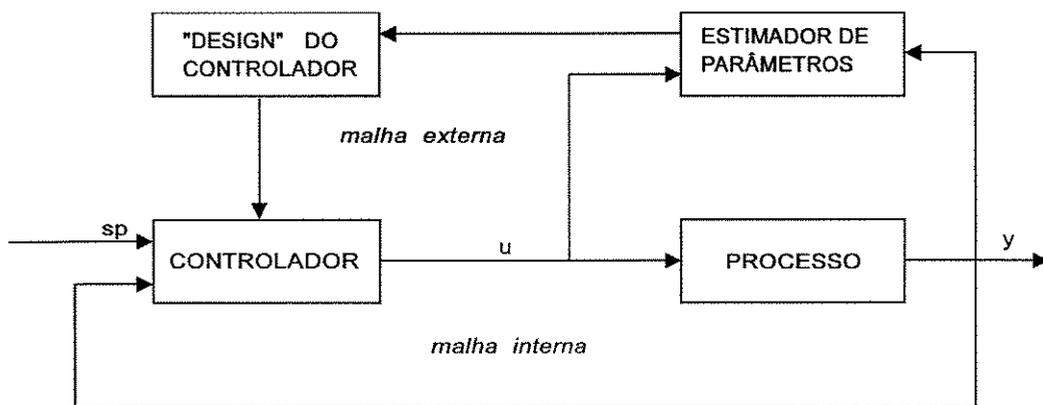


Fig. 3.2 - Diagrama de blocos do Controlador Auto-ajustável (STR).

O regulador pode ser visto como composto de duas malhas. A malha interna consiste do processo e de um regulador ordinário linear por retroalimentação. Os parâmetros do regulador são ajustados pela malha externa que é composta de um estimador de parâmetros recursivo e um mecanismo de ajuste dos parâmetros do controlador. Para se obter boas estimativas dos parâmetros pode ser necessária a introdução de sinais de perturbações no processo.

O STR foi originalmente proposto por Kalman (1958), que construiu um computador específico para implementação do controlador e, mais tarde, clareado por Aström e Witenmark (1973).

Sendo o controlador muito flexível com respeito ao método de ajuste dos parâmetros, extensões diferentes surgiram. Reguladores auto-ajustáveis baseados em margens de fase e amplitude são discutidos em Aström (1982) e os auto-ajustáveis

baseados em deslocamento de pólos têm sido investigados por diversos autores (ver Aström, 1983).

Assim como na adaptação de parâmetros, diferentes esquemas de estimação também têm sido utilizados, como por exemplo: Aproximação Estocástica, Mínimos Quadrados, Filtro Kalman Estendido.

Muitas aplicações experimentais surgiram com a maior disponibilidade dos computadores digitais. Dahlqvist (1981) aplicou o STR no controle multivariável de uma coluna de destilação piloto, onde os produtos de topo e fundo eram simultaneamente controlados. Devido à natureza não-linear do processo, os parâmetros do processo eram representativos somente no ponto operacional em que foram estimados. Se fosse decidido operar a coluna em outro ponto, seria necessário encontrar novos valores dos parâmetros do controlador e, para a realização deste ajuste em linha, foi implementado o STR.

Em seu artigo, Dahlqvist cita algumas aplicações do STR em processos industriais como em máquinas de papel, digestores de polpa e extração de minérios, relatados em Borisson e Wittenmark (1974), Borisson e Syding (1976) e Cegrell e Hedqvist (1974, 1975).

Kiparissides e Shah (1983) mostram um estudo de simulação de um reator de polimerização em batelada onde um STR foi implementado. O reator é altamente não-linear, instável e opera em regime transiente. Explicando o porquê da utilização deste tipo de controlador, os autores afirmam que muitas vezes a descrição estado-espaco de processos não-lineares com tempo morto é difícil e inadequada, além do que para vários casos pode não haver disponível um modelo dinâmico do processo. Portanto, muitas técnicas de controle não podem ser aplicadas diretamente e a aplicação do controle adaptativo aparece como uma solução promissora para muitos destes problemas de controle.

Segundo Aström (1983), a atenção considerável dispensada ao STR se deve ao fato dele ser flexível, de fácil entendimento e de fácil implementação com microprocessadores.

Em nosso estudo particular, onde o processo é não-linear e transiente, o STR mostra-se como uma ferramenta atraente para a obtenção de composição fixa no topo da coluna.

3.3 IMPLEMENTAÇÃO DO PAC

Visando encontrar um planejamento para a razão (ou vazão) de refluxo que levasse a destilação descontínua a operar com composição de topo constante, o referido processo foi dividido em uma seqüência de estados pseudo-estacionários, nos quais poderia ser estudado como um processo contínuo.

Como variável auxiliar, usada no mecanismo de ajuste (fig. 3.1), utilizou-se a composição (ou temperatura) de fundo da coluna por ser um forte indicativo das futuras composições de topo no equipamento.

Portanto, através de simulações “off-line” de colunas contínuas, em diversos estados pseudo-estacionários, a dependência da razão de refluxo com a variável auxiliar pode ser encontrada por regressão não-linear dos dados obtidos (eq. 3.1). Esta relação é então utilizada para o termo de estado estacionário, u_{ss} , na equação do controlador PI.

$$u_{ss} = u_{ss} \text{ (variável auxiliar)} \quad (3.1)$$

Para a implementação do mecanismo de ajuste dos parâmetros do controlador, é necessária a caracterização prévia do processo estudado. Para este propósito, utilizaram-se programas desenvolvidos para simulação de colunas contínuas e em batelada. Considerando-se, mais uma vez, a destilação em batelada como uma seqüência de destilações contínuas, o Método da Curva de Reação do Processo foi aplicado “off-line”, em cada intervalo pseudo-estacionário, para identificação dos parâmetros do processo aproximado por uma função de transferência de primeira ordem com atraso de transporte (eq. 3.2).

$$G_p(s) = \frac{K_p e^{-t_d s}}{\tau_p s + 1} \quad (3.2)$$

onde K_p , τ_p e t_d são, respectivamente, o ganho, a constante de tempo e o tempo morto do processo, que não possuem valores constantes devido às não-linearidades e ao regime

transiente do processo, e que dependem fortemente das propriedades da mistura a ser destilada.

Com os resultados obtidos para cada intervalo pseudo-estacionário, a adaptação dos parâmetros do controlador PI pode ser programada através da aplicação das equações de ajuste semiempíricas propostas por Cohen-Coon (eqs. 3.3):

$$K_c = \frac{1}{K_p} \cdot \frac{\tau_p}{t_d} \left(0,9 + \frac{t_d}{12\tau_p} \right) \quad (3.3a)$$

$$\tau_I = t_d \left(\frac{30 + 3t_d/\tau_p}{9 + 20t_d/\tau_p} \right) \quad (3.3b)$$

onde K_c e τ_I são, respectivamente, o ganho e a constante de tempo integral do controlador.

Efetuando-se, então, um ajuste de curvas dos parâmetros do controlador em função da variável auxiliar, para o conjunto de intervalos utilizado, obtém-se o planejamento da adaptação do controlador dado pelas equações 3.4:

$$K_c = K_c(\text{variável auxiliar}) \quad (3.4a)$$

$$\tau_I = \tau_I(\text{variável auxiliar}) \quad (3.4b)$$

Com a implementação das equações 3.4 no bloco de “mecanismo de ajuste” da figura 3.1, verifica-se que somente com o valor atual da variável auxiliar, medida ou inferenciada, atualizam-se os parâmetros do controlador.

3.4 IMPLEMENTAÇÃO DO STR

Ao contrário do esquema de adaptação programada (PAC), o STR é uma classe de controlador adaptativo em que o projeto e a implementação não necessitam do conhecimento das propriedades dos componentes envolvidos no processo a ser controlado.

No STR, o processo deve ser representado por um modelo reduzido discreto do tipo polinomial ou obtido de função de transferência de ordem n com ou sem atraso de transporte associado.

Como o desenvolvimento do PAC fundamentou-se na representação do processo por uma função de transferência de primeira ordem com tempo morto e o controlador mostrou-se robusto, adotou-se a mesma função transferência (eq. 3.2) para o desenvolvimento do STR.

Tendo em vista que a maioria dos elementos finais de controle (ex: válvulas) são ativados por sinais contínuos (ex: ar comprimido) e os comandos de controle produzidos pelo programa de controle são discretos, deve-se utilizar um elemento que converta este sinal discreto em contínuo. Para tanto, utilizou-se um “data-hold” de ordem zero representado pela seguinte função transferência no domínio de Laplace:

$$G_H(s) = \frac{1 - e^{-sT}}{s} \quad (3.5)$$

onde T é o intervalo de amostragem.

Portanto, visando a discretização do modelo escolhido para o processo, a função transferência de pulso entre as variáveis controlada e manipulada pode ser encontrada com o auxílio da transformada-z:

$$\frac{\hat{y}(z)}{\hat{u}(z)} = G_H G_p(z) = Z \left\{ G_H(s) G_p(s) \right\} = Z \left\{ \frac{1 - e^{-sT}}{s} \cdot \frac{K_p e^{-t_d s}}{\tau_p s + 1} \right\} \quad (3.6)$$

Deixando a descrição detalhada da obtenção do modelo discreto via transformada-z no ANEXO 2, apresenta-se neste capítulo somente a equação discreta final que descreve o processo:

$$y_i = a_1 y_{i-1} + b_1 u_{i-1} + b_2 u_{i-2} \quad (3.7)$$

$$\text{onde } b_1 = K_p \left[1 - e^{-\frac{(T-t_d)}{\tau_p}} \right] \quad (3.8a)$$

$$b_2 = K_p \left[e^{-\frac{(T-t_d)}{\tau_p}} \cdot \frac{-T}{\tau_p} - e^{-\frac{T}{\tau_p}} \right] \quad (3.8b)$$

$$a_1 = e^{-\frac{T}{\tau_p}} \quad (3.8c)$$

3.4.1 Identificação em Linha do Processo

Sabendo-se da forte dependência de um bom desempenho do controlador digital com o nível de conhecimento da dinâmica do processo a ser controlado, comprova-se a fragilidade do sistema de controle se o processo for pouco conhecido ou mal representado.

Portanto, um processo de identificação mais sofisticado em relação ao PAC (fora de linha) será implementado no STR, apesar da utilização do mesmo modelo reduzido, agora discretizado (eq. 3.7).

Esta tarefa pode ser realizada através de qualquer procedimento padrão de estimação recursiva. Sugerido por Stephanopoulos (1984), no capítulo que trata da Identificação de Processos e Controle Adaptativo, o Método Recursivo dos Mínimos Quadrados foi o escolhido para o STR em desenvolvimento.

Para a determinação dos parâmetros desconhecidos do conjunto de equações (3.8), ou seja, K_p , τ_p e t_d , o processo real deve ser perturbado por uma variação em pulso ou degrau na variável manipulada, vazão de refluxo (u), e efetuar-se a amostragem da variável controlada, composição de topo (y). Os valores otimizados dos parâmetros desconhecidos serão, então, aqueles que resultarem num erro quadrático médio mínimo (eq. 3.9) entre o valor amostrado da variável controlada (\tilde{y}_i) e o calculado pelo modelo discreto da equação (3.7), em I instantes de amostragens passados. Deste modo, a solução deve ser dada pelo seguinte problema de mínimos quadrados:

$$P = \frac{1}{I} \sum_{i=1}^I \text{erro}^2 = \sum_{i=1}^I (\tilde{y}_i - y_i)^2 \quad (3.9)$$

$$f_1 = \frac{\partial P}{\partial K_p} = 0 \quad (3.10a)$$

$$f_2 = \frac{\partial P}{\partial \tau_p} = 0 \quad (3.10b)$$

$$f_3 = \frac{\partial P}{\partial t_d} = 0 \quad (3.10c)$$

Através das equações (3.7), (3.8), (3.9) e (3.10), três equações algébricas não-lineares (f_1 , f_2 e f_3), funções dos três parâmetros desconhecidos, foram obtidas analiticamente (ANEXO 3).

O problema se transforma então na localização das raízes (K_p , τ_p e t_d), do conjunto de equações 3.10, que passarão a ser chamadas de x_1 , x_2 e x_3 . Segundo Press *et al.* (1986), no capítulo que trata da localização de raízes e conjuntos de equações não-lineares, o método numérico mais indicado para a solução é o de Newton-Raphson, que se adequa facilmente a múltiplas dimensões, sendo um recurso bastante eficiente para se atingir a convergência, desde que as raízes existam e que sejam disponíveis estimativas iniciais adequadas.

Chamando de X o vetor de valores das raízes, então, na vizinhança de X , cada uma das funções f_k pode ser expandida em Séries de Taylor, e, desprezando-se os termos de segunda ordem e superiores, obtém-se:

$$f_k(X + \delta X) = f_k(X) + \sum_{j=1}^3 \frac{\partial f_k}{\partial x_j} \delta x_j \quad k=1,2,3. \quad (3.11)$$

Sendo o objetivo do método matemático mover simultaneamente em direção ao zero as funções $f_k(X + \delta X)$, podem-se obter as correções dos valores das raízes, δX , através da solução do sistema linear de equações abaixo, utilizando-se o Método de Decomposição LU.

$$\sum_{j=1}^3 \frac{\partial f_k}{\partial x_j} \delta x_j = -f_k(X) \quad k=1,2,3. \quad (3.12)$$

Para tanto é necessário que sejam obtidas as derivadas parciais de cada uma das funções f_k obtidas no ANEXO 3. Visando a rapidez dos cálculos que deverão ser efetuados a cada instante de amostragem, mais uma vez estas derivadas foram obtidas analiticamente e estão descritas no ANEXO 4.

Com os valores de δX determinados, obtêm-se os novos valores das raízes através da seguinte equação:

$$x_j^{\text{novo}} = x_j^{\text{velho}} + \delta x_j \quad j=1,2,3. \quad (3.13)$$

Se este novo vetor X fizer com que as funções f_k se anulem ou se o vetor δX for muito pequeno, as raízes (parâmetros otimizados do processo) estarão determinadas. Caso contrário, retorna-se à solução do sistema de equações (3.12) acima.

O algoritmo simplificado para identificação de processos em tempo real é apresentado na figura 3.3 e sua implementação computacional no ANEXO 5.

Para que se faça a perturbação do processo, é necessário que se respeitem os valores limites da variável a ser manipulada e que esta perturbação não seja grande a ponto de desestabilizar o sistema. Se o processo estiver em malha fechada, a perturbação pode ser a própria ação de controle sobre a variável manipulada.

3.4.2 Projeto do Controlador

No diagrama de blocos representativo do controlador auto-ajustável (fig. 3.2), o bloco denominado "Design" do Controlador nada mais é do que a solução em linha do problema de projeto do sistema de controle, onde os parâmetros do processo já estejam determinados.

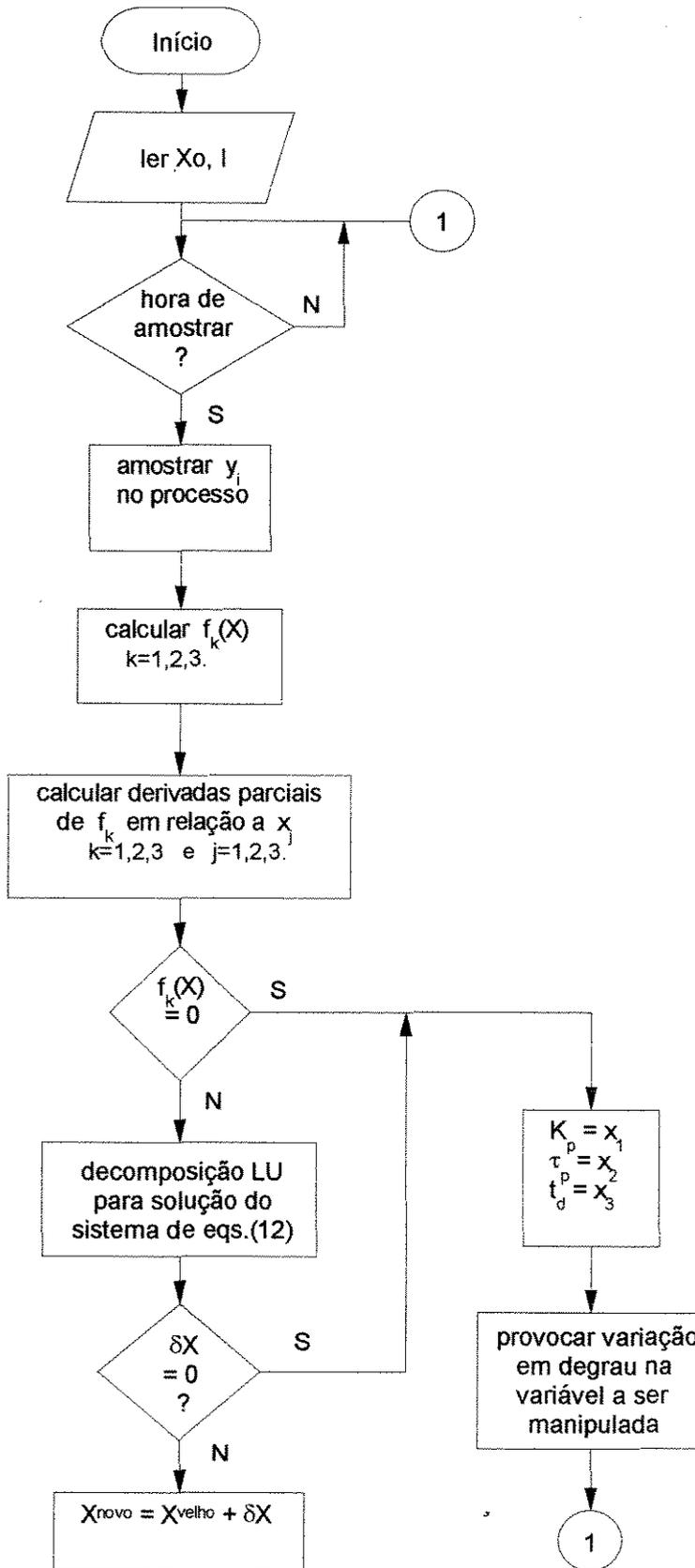


Fig. 3.3 - Algoritmo simplificado para identificação de processos em linha.

Assim como na escolha do esquema de identificação recursiva de parâmetros do processo, o STR é bastante amplo em relação à escolha da metodologia de projeto do controlador. Fazendo uso da flexibilidade computacional oferecida pelos controladores digitais, as leis de controle não se restringem mais somente àquelas com estrutura fixa convencional (equivalentes digitais ao P, PI e PID), mas podem-se implementar leis de controle derivadas de conjuntos de especificações impostas à resposta em malha fechada do processo a ser controlado.

No presente trabalho, o equivalente digital ao PI analógico, ajustado por critério de sintonia de Cohen-Coon, e uma lei de controle obtida do Algoritmo de Dahlin foram individualmente implementados.

O familiar controlador Proporcional-Integral tem sua forma geral descrita por:

$$u(t) = u_{ss} + K_c \left[E(t) + \frac{1}{\tau_I} \int_0^t E(t) dt \right] \quad (3.14)$$

onde K_c e τ_I são, respectivamente, os parâmetros ajustáveis ganho e constante de tempo integral do controlador, u_{ss} é o valor da variável manipulada no estado estacionário, e $E(t)$ é o erro entre a variável controlada e seu valor de referência no tempo t , ou seja:

$$E(t) = y(t) - y_{sp}(t) \quad (3.15)$$

A equação às diferenças correspondente ao PI digital se torna, em sua forma posição e no instante i :

$$u_i = u_{ss} + K_c \left[E_i + \frac{T}{\tau_I} \sum_{j=0}^i E_j \right] \quad (3.16)$$

Procurando prevenir a saturação do termo integral, o PI em sua forma velocidade foi alternativamente utilizado:

$$u_i = u_{i-1} + K_c \left(1 + \frac{T}{\tau_I} \right) E_i - K_c E_{i-1} \quad (3.17)$$

Independente da forma escolhida, posição ou velocidade, o controlador PI digital tem aqui seus parâmetros ressintonizados pelo conjunto de equações (3.3). Estas equações de sintonização propostas por Cohen-Coon foram utilizadas como uma primeira aproximação dos parâmetros, desde que normalmente as mesmas levam a ganhos (K_c) de valores elevados, resultando em atitudes bruscas do controlador.

Já o método alternativo implementado, sugerido por Dahlin (Stephanopoulos, 1984), impõe que a resposta em malha fechada de um Controlador Digital Direto (DDC) se comporte como um sistema de primeira ordem com tempo morto, frente a um degrau unitário no valor de referência (y_{sp}). Ou seja, no Domínio de Laplace:

$$\bar{y}(s) = \frac{e^{-\theta s}}{\mu s + 1} \cdot \frac{1}{s} \quad (3.18)$$

onde μ é a constante de tempo da resposta desejada (parâmetro ajustável) e θ o atraso de transporte da resposta ($\theta = kT$, com $k =$ inteiro). Na representação discreta através da transformada-z, tem-se a equação (3.18) dada por:

$$\hat{y}(z) = \frac{z^{-k} (1 - e^{-T/\mu}) z^{-1}}{(1 - z^{-1})(1 - e^{-T/\mu} z^{-1})} \quad (3.19)$$

Substituindo-se uma variação em degrau unitário no valor de referência ($\hat{y}_{sp}(z) = \frac{1}{1 - z^{-1}}$), pode-se encontrar:

$$\frac{\hat{y}(z)}{\hat{y}_{sp}(z)} = \frac{z^{-k} (1 - e^{-T/\mu}) z^{-1}}{1 - e^{-T/\mu} z^{-1}} \quad (3.20)$$

Conhecendo-se a resposta em malha fechada de um DDC típico, frente a variações no valor de referência, como sendo:

$$\hat{y}(z) = \frac{G_H G_p(z) D(z)}{1 + G_H G_p(z) D(z)} \hat{y}_{sp}(z) \quad (3.21)$$

pode-se encontrar a função transferência do controlador a ser projetado, $D(z)$, isolando-se a mesma e notando-se ser a única variável desconhecida da equação (3.21). Portanto:

$$D(z) = \frac{1}{G_H G_p(z)} \cdot \frac{\hat{y}(z) / \hat{y}_{sp}(z)}{1 - \hat{y}(z) / \hat{y}_{sp}(z)} \quad (3.22)$$

Substituindo-se a equação (3.20) na equação (3.22), obtém-se:

$$\frac{\hat{u}(z)}{\hat{E}(z)} = D(z) = \frac{1}{G_H G_p(z)} \cdot \frac{(1 - e^{-T/\mu}) z^{-k-1}}{1 - e^{-T/\mu} z^{-1} - (1 - e^{-T/\mu}) z^{-k-1}} \quad (3.23)$$

Finalmente, conhecendo-se (anexo 2) a função transferência de pulso (eq. A2.10), $G_H G_p(z)$, e substituindo na equação (3.23), tem-se:

$$\frac{\hat{u}(z)}{\hat{E}(z)} = \frac{1}{z^{-1}(b_1 + b_2 z^{-1})} \cdot \frac{a_1 z^{-1} (1 - e^{-T/\mu}) z^{-k-1}}{1 - e^{-T/\mu} z^{-1} - (1 - e^{-T/\mu}) z^{-k-1}} \quad (3.24)$$

Considerando-se o tempo morto de mesma ordem de grandeza do intervalo de amostragem para prevenir possível divisão por zero na lei de controle obtida, e aplicando-se a transformada inversa Z^{-1} , tem-se a lei de controle dada por:

$$u_i = \frac{(1 - e^{-T/\mu})}{K_p (1 - e^{-T/\tau_p})} \cdot (E_{i-1} - e^{-T/\tau_p} E_{i-2}) + e^{-T/\mu} u_{i-1} + (1 - e^{-T/\mu}) u_{i-3} \quad (3.25)$$

3.5 RESULTADOS

Primeiramente, com o objetivo de exemplificar a realização do projeto do sistema de controle adaptativo programável (PAC), apresentam-se alguns dados obtidos da etapa de caracterização do processo e as equações obtidas quando, de uma simples mistura binária de 60% molar de metanol e 40% etanol, deseja-se obter metanol 90% puro.

A separação, simulada, realizava-se em uma coluna de 7 pratos, com Eficiência de Murphree de 80% em cada um, e cuja geometria constituía-se por uma área de 44 cm² e comprimento e altura de vertedouro de, respectivamente, 0,8 e 2,0 cm². A pressão de operação da coluna era a atmosférica e o calor fornecido ao refeedor fixo em 48 kcal/min.

Como parte inicial do projeto, encontrou-se através de uma seqüência de simulações de uma coluna contínua de mesmas especificações acima e em malha aberta, uma série de dados dos parâmetros do processo (tabela 3.1). Para tanto, variava-se somente a hipotética composição de alimentação, imitando a diminuição da fração molar do componente mais volátil (x_B) ocorrida no refeedor da coluna em batelada e, depois de atingido o estado estacionário, perturbava-se, em degrau, a variável manipulada.

Utilizando-se, então, os dados da tabela 3.1 e as equações semiempíricas de ajuste (eqs.3.3), determinaram-se as variações programadas de K_c e τ_I , em função da composição de fundo da coluna:

$$K_c = -649,629 x_B + 458,344 \quad (3.26a)$$

$$\tau_I = -4,379 + 67,823 x_B - 272,628 x_B^2 + 364,04 x_B^3 \quad (3.26b)$$

Optando-se pela implementação da lei de controle PI na forma velocidade (eq. 3.17), não foi necessário encontrar o planejamento para a vazão de refluxo usada como “bias” do controlador, u_{ss} .

Deste modo, foi simulada a separação sob controle do PAC/PI e os resultados mostrados na figura 3.4.

Tabela 3.1 - Parâmetros do processo obtidos da aplicação do método da curva de reação do processo, em vários intervalos “pseudo-estacionários”, da separação metanol-etanol.

x_B	K_p (mol/min)	τ_p (min)	t_d (min)
0,428	0,110	24,184	1,1729
0,412	0,145	22,553	0,7989
0,397	0,155	23,186	0,6629
0,362	0,180	23,866	0,6119
0,330	0,190	25,062	0,4420
0,309	0,205	25,508	0,4080
0,293	0,210	25,704	0,3910
0,267	0,215	27,576	0,4590
0,252	0,225	27,384	0,2967
0,242	0,220	28,264	0,3570
0,225	0,230	28,517	0,4250
0,214	0,225	28,213	0,2890
0,200	0,240	28,965	0,3914
0,186	0,235	29,174	0,3740
0,163	0,265	30,535	0,3060
0,153	0,250	29,535	0,2890

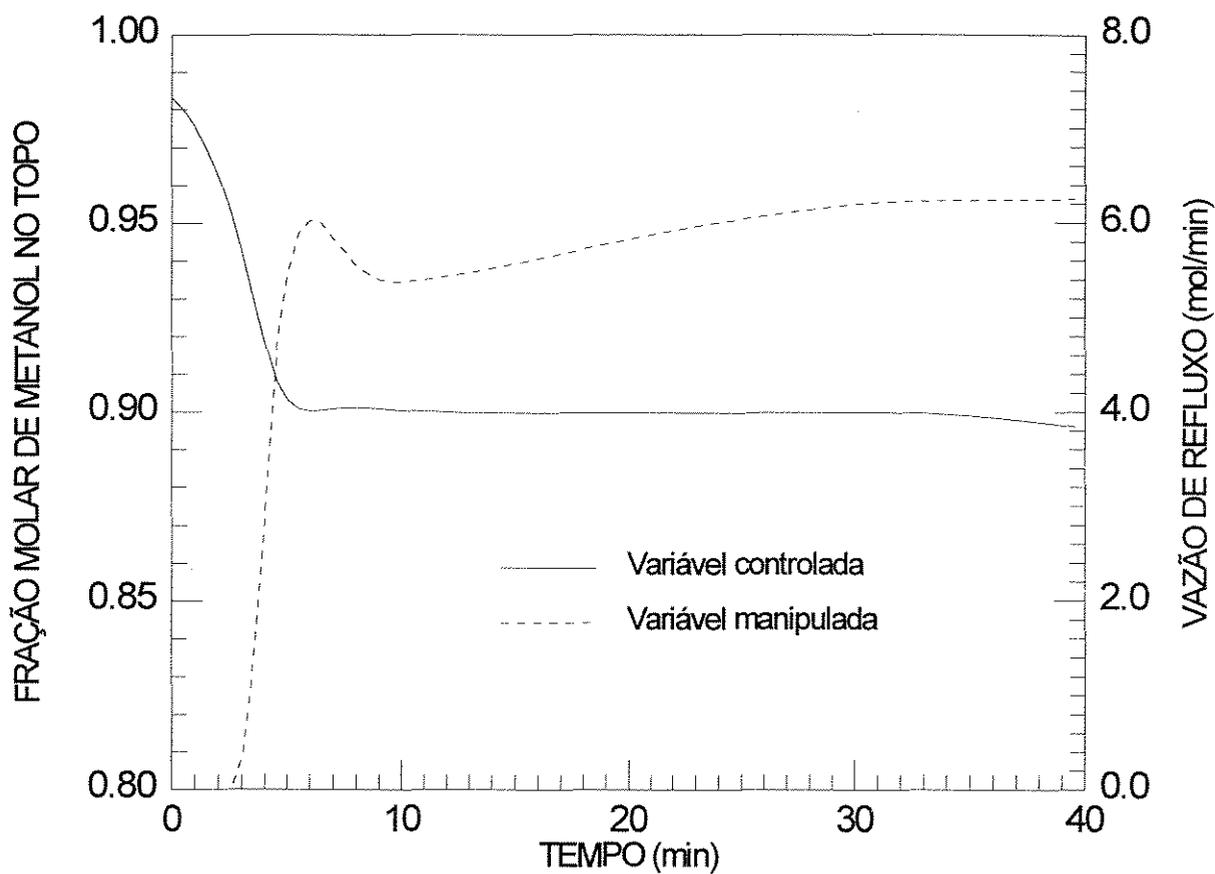


Fig. 3.4 - Recuperação de metanol 90% puro da mistura de 60%metanol-40%etanol sob controle PAC/PI (Forma Velocidade).

Para testar os sistemas de controle adaptativo propostos e comparar seus desempenhos quando implementados à coluna de destilação em batelada, uma série de simulações foram ainda efetuadas para diversas configurações deste equipamento e com dois sistemas ternários diferentes a serem decompostos em seus componentes.

A tabela 3.2 resume os principais dados utilizados em cada caso estudado.

3.5.1 Caso 1

Os controladores PAC e STR foram aplicados ao caso 1 com o objetivo de se operar a coluna a refluxo variável para obtenção de n-hexano 80% puro no topo da coluna.

Conforme descrito no item que trata da implementação do PAC, uma série de simulações “off-line”, com os dados do caso em estudo, foram efetuadas para o projeto do mesmo. A partir destas simulações, o planejamento da taxa de refluxo que permitisse alcançar o objetivo (exclusivamente para as condições do caso 1), foi determinado e representado pela seguinte relação entre a variável controlada, u_{ss} , e a variável auxiliar composição de fundo, x_B :

$$u_{ss} = 8,036 - 44,403x_B + 281,975x_B^2 - 652,015x_B^3 \quad (3.27)$$

Através do Método da Curva de Reação do Processo, aplicado em cada intervalo pseudo-estacionário, o processo foi então caracterizado, ou seja, K_p , τ_p e t_d foram identificados nestes intervalos. Fazendo uso das regras de Cohen-Coon (eqs. 3.3), os parâmetros do processo obtidos levaram à determinação das equações de adaptação do controlador PI:

$$K_c = -7427,61 + 184274,0x_B - 1307670,0x_B^2 + 2931470,0x_B^3 \quad (3.28a)$$

$$\tau_I = 11,498 - 218,346x_B + 1390,140x_B^2 - 2749,580x_B^3 \quad (3.28b)$$

Tabela 3.2 - Dados de entrada para simulações.

Especificações	CASO 1	CASO 2	CASO 3	CASO 4
Mistura	n-hexano benzeno tolueno	n-hexano benzeno tolueno	n-hexano benzeno tolueno	metanol etanol água
Comp. de alimentação (fração molar)	0,33;0,33;0,33	0,15;0,75;0,10	0,30;0,45;0,25	0,60;0,30;0,10
Quantidade alimentada (mol)	100	100	100	300
Pureza de topo desejada (fração molar)	0,80; 0 ;0	0; 0,98; 0	0,65; 0,70; 0	0,90; 0; 0
Desvio máximo permitido (%)	3	3	4	1
N ^o estágios	12	9	6	12
Eficiência Murphree (%)	100	100	100	100
“Hold-up” do condensador (mol)	0,50	0,50	0,50	0,50
Vazão de vapor do refervedor (mol/min)	5,0	5,0	5,0	5,0
Pressão de operação (atm)	1,0	1,0	1,0	1,0
Área do prato (cm ²)	50,0	50,0	50,0	50,0
Comprimento do vertedouro (cm)	2,0	2,0	2,0	2,0
Altura do vertedouro (cm)	5,0	5,0	5,0	5,0
Intervalo de amostragem (s)	12	12	12	12/30

Estas equações de projeto do PAC para o caso 1 (eqs. 3.28), foram implementadas no bloco de mecanismo de ajuste da fig. 3.1 e os resultados mostraram que 70% do n-hexano carregado puderam ser recuperados, com a pureza desejada, em 45 minutos.

Para comparação entre os sistemas adaptativos propostos, o STR foi também aplicado ao caso 1. Conforme explicado na descrição do STR, nenhuma simulação prévia foi efetuada no projeto do mesmo, já que este sistema é completamente auto-adaptativo e portanto de projeto totalmente independente das especificações do equipamento ou da mistura a ser separada. As leis de controle PI e algoritmo de Dahlin foram individualmente testadas.

Através da figura 3.5a pode-se comparar o comportamento da variável controlada quando o equipamento funcionava sob ação do PAC ou STR. Na figura 3.5b, pode-se notar que as ações de controle tomadas pelo PAC são mais abruptas resultando em resposta mais próxima do valor de referência desejado. Porém, estas grandes variações nem sempre são implementáveis na prática e portanto não são desejáveis.

Já no STR estas grandes variações não são permitidas para que se possa assegurar a convergência do método de otimização do algoritmo de identificação do processo. Se a região provável de localização das raízes do sistema de equações não-lineares (3.10) mudar abruptamente, o método pode não convergir, indicando de maneira errada que estas raízes não existem. Deste modo, um limite de flutuação de $\pm 2\%$ na variável manipulada (em relação ao seu valor imediatamente anterior) é imposto e, como consequência destas variações mais brandas, aparece um pequeno desvio na resposta em malha fechada.

Ainda na figura 3.5a pode-se notar uma queda acentuada da variável controlada pelo STR com PI na forma velocidade, o que não permitiu uma recuperação maior que 63% do n-hexano carregado, em sua pureza desejada, num tempo bem superior (57 min) ao gasto pelo PAC/PI (45 min).

O STR/PI na forma posição somente pode manter a composição de topo desejada se o somatório dos erros anteriores fosse anulado quando o erro atual se aproximasse de zero. Mesmo sob esta proteção contra a saturação do termo integral,

somente uma quantidade de 49% do n-hexano carregado pode ser recuperada com 80% de pureza.

Os diferentes comportamentos obtidos na figura 3.5b, para a variável manipulada, a partir do STR/PI e PAC/PI já eram esperados desde que o STR/PI não segue um planejamento imposto ao “bias” do controlador (u_{ss}) - como no PAC/PI Forma Posição- mas sim faz uso do valor anterior da variável manipulada para este termo de estado estacionário. Para iniciar este sinal de saída do controlador STR/PI, uma metodologia de tentativa-e-erro revelou o melhor valor a ser utilizado.

O resultado da implementação do STR com algoritmo de Dahlin, também mostrado nas figuras 3.5a e 3.5b, mostrou-se superior ao obtido com STR/PI, recuperando também os 70% alcançados com PAC/PI, porém com um tempo de 52 minutos. Resultado bastante bom em vista da limitação imposta às ações de controle e frente ao projeto com total desconhecimento prévio do sistema a ser controlado.

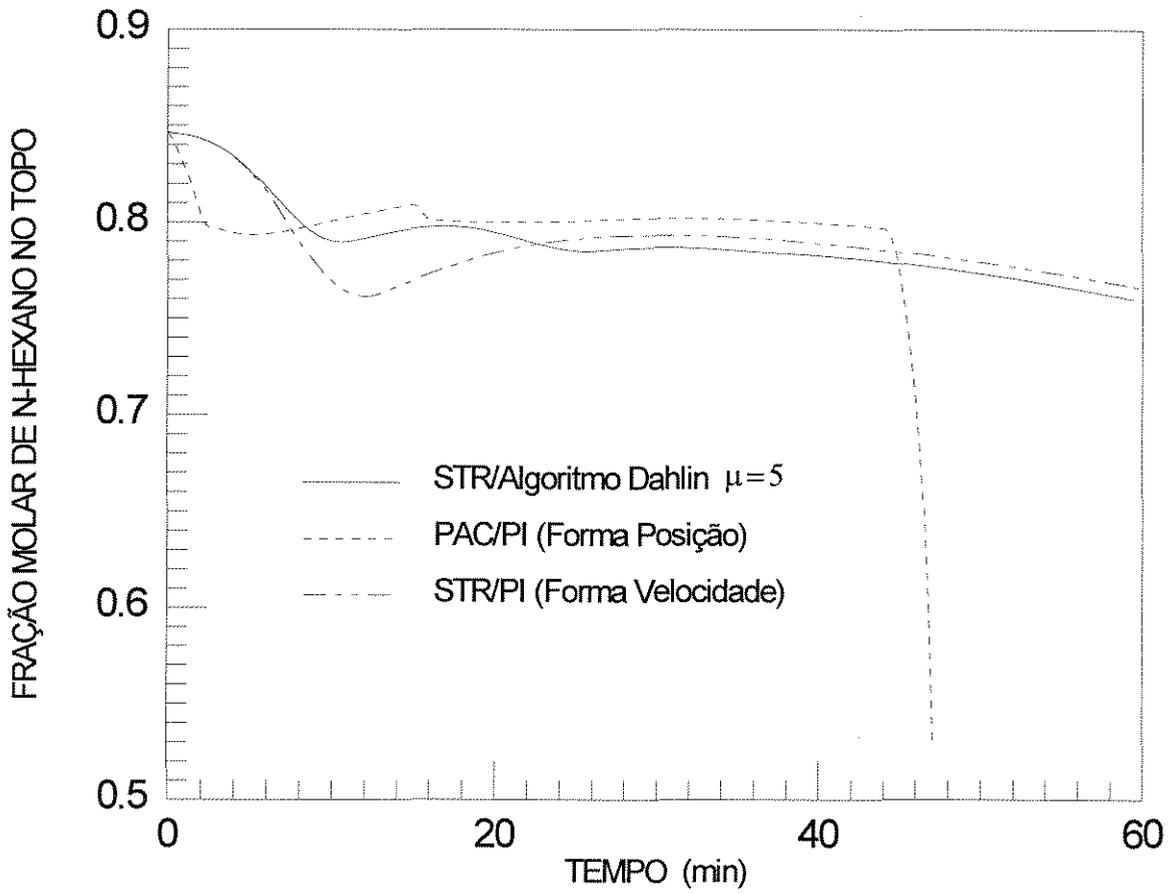


Fig. 3.5a - Comportamento da variável controlada para o caso 1.

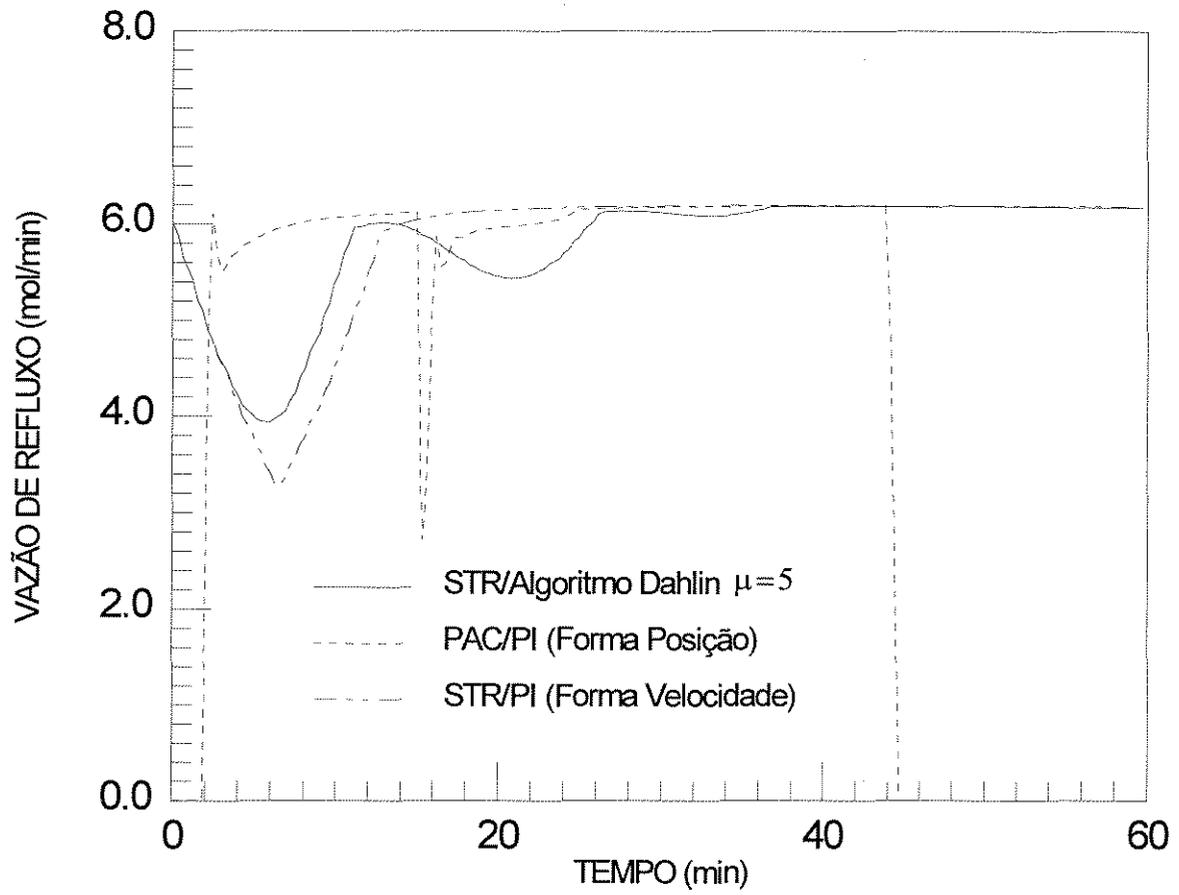


Fig. 3.5b - Comportamento da variável manipulada para o caso 1.

3.5.2 Caso 2

Este exemplo se concentra em apontar as diferenças entre as respostas dos algoritmos de Dahlin e PI quando empregados em controladores auto-ajustáveis (STR).

As figuras 3.6a e 3.6b mostram a atuação do STR quando se deseja obter Benzeno com pureza de 98% de uma mistura inicial de 15% de n-hexano, 75% de benzeno e tolueno.

Pode-se verificar na figura 3.6b que a lei de controle resultante do algoritmo de Dahlin reage rapidamente à queda de concentração de benzeno na coluna, enquanto que a ação integral da lei de controle PI provoca uma resposta muito lenta. Como resultado, a quantidade de benzeno recuperada no topo, com 98% de pureza, foi 10% superior quando o algoritmo de Dahlin foi utilizado. Esta percentagem reduz-se minimamente se o PI na forma velocidade é utilizado ao invés do PI na forma posição.

O STR/PI mostrou-se mais sensível que o STR/Dahlin a estimativas iniciais ruins fornecidas ao algoritmo de identificação do processo. Quando isto ocorria, o regulador com lei PI mostrou-se instável. Esta instabilidade aparece também se o ajuste de Cohen-Coon é utilizado sem correções. Para o PI, o ganho proporcional K_c , calculado aqui a cada intervalo de amostragem, teve de ser reduzido em torno de 1000 vezes para que fossem obtidas respostas aceitáveis.

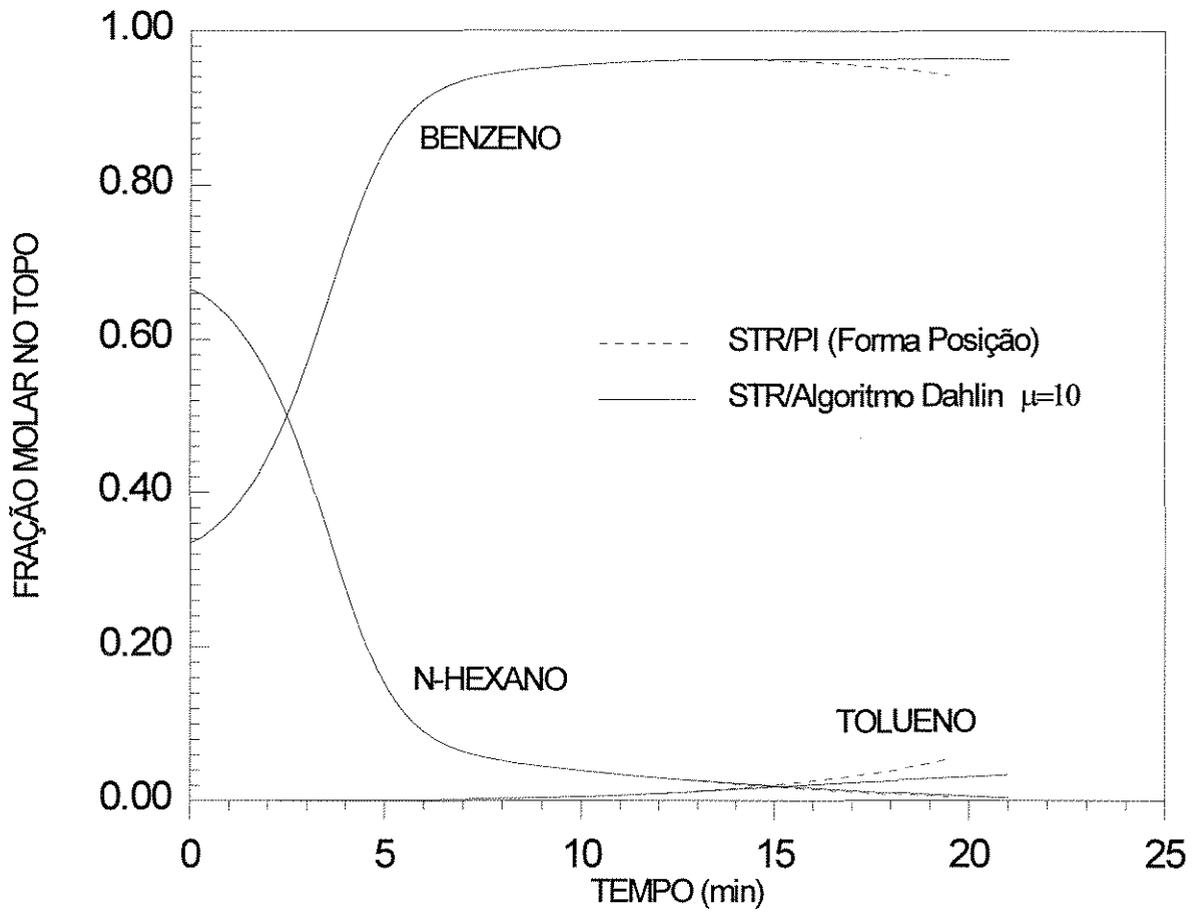


Fig. 3.6a - Comportamento da composição de topo, em malha fechada, para o caso 2.

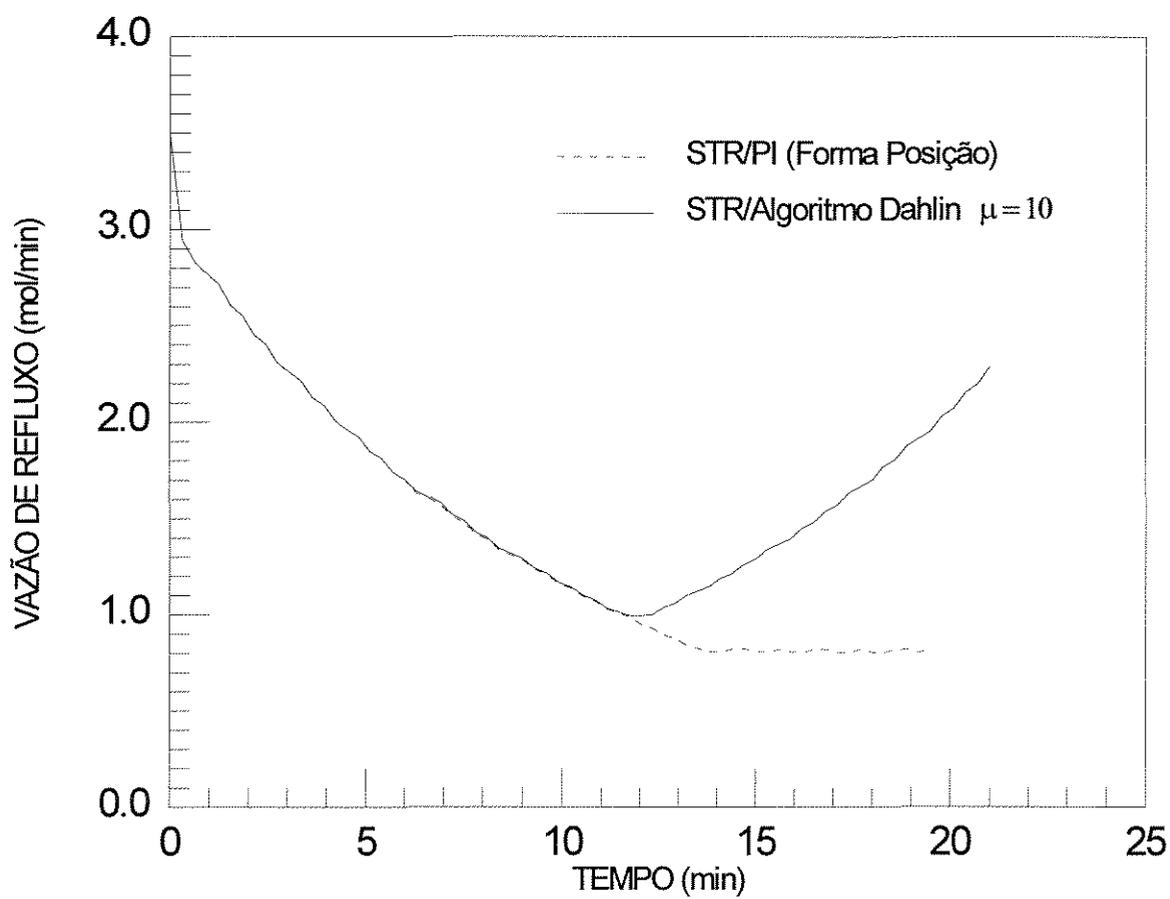


Fig. 3.6b - Comportamento da variável manipulada para o caso 2.

3.5.3 Caso 3

Com o propósito de obtenção, no topo da coluna, de mais de um componente com pureza especificada, o sistema de controle foi acrescido de um conjunto de condições que apontasse o fim de cada ciclo da batelada. Deste modo, quando identificado o fim da retirada de um determinado componente, a variável controlada passa a ser a fração molar do próximo componente mais volátil e, portanto, a ação de controle se concentra em maximizar a retirada deste segundo componente dentro de suas especificações.

A condição mais importante para a troca de variável controlada é a fração molar, atualmente controlada, estar abaixo de seu valor de referência quando a variável manipulada já estiver em seu limite máximo. Entretanto, oscilações indesejáveis podem ocorrer na variável controlada, indicando erroneamente que seria o momento de efetuar a troca. Para evitar que isto ocorra, acrescenta-se ao sistema de controle um filtro para “amaciar” a curva de dados amostrados. Informações da variável controlada são estocadas por determinado tempo, resultando em valores médios ao final deste tempo. Assim, se a curva “amaciada” obtida decresce e a condição ditada acima for verdadeira, o sistema de controle adapta-se para a retirada do próximo componente, ou seja, novas estimativas iniciais dos parâmetros do processo devem ser fornecidas ao Método de Newton-Raphson e conseqüentemente efetuam-se as necessárias adaptações dos parâmetros de projeto da lei de controle utilizada.

As figuras 3.7a e 3.7b ilustram os resultados da aplicação das condições citadas acima para obtenção de n-hexano com 65% de pureza e benzeno 70% puro a partir de uma mistura ternária inicial de 30% n-hexano, 45% benzeno e 25% tolueno (Tab. 3.2).

A brusca mudança de direção da variável controlada depois de atingido o valor máximo, mostrada na figura 3.7b, identifica a mudança da variável controlada quando aproximadamente 28% do valor da carga do refervedor remanescente - depois do equilíbrio inicial atingido e os “hold-up’s” iniciais estabelecidos - foram obtidos como produto de topo.

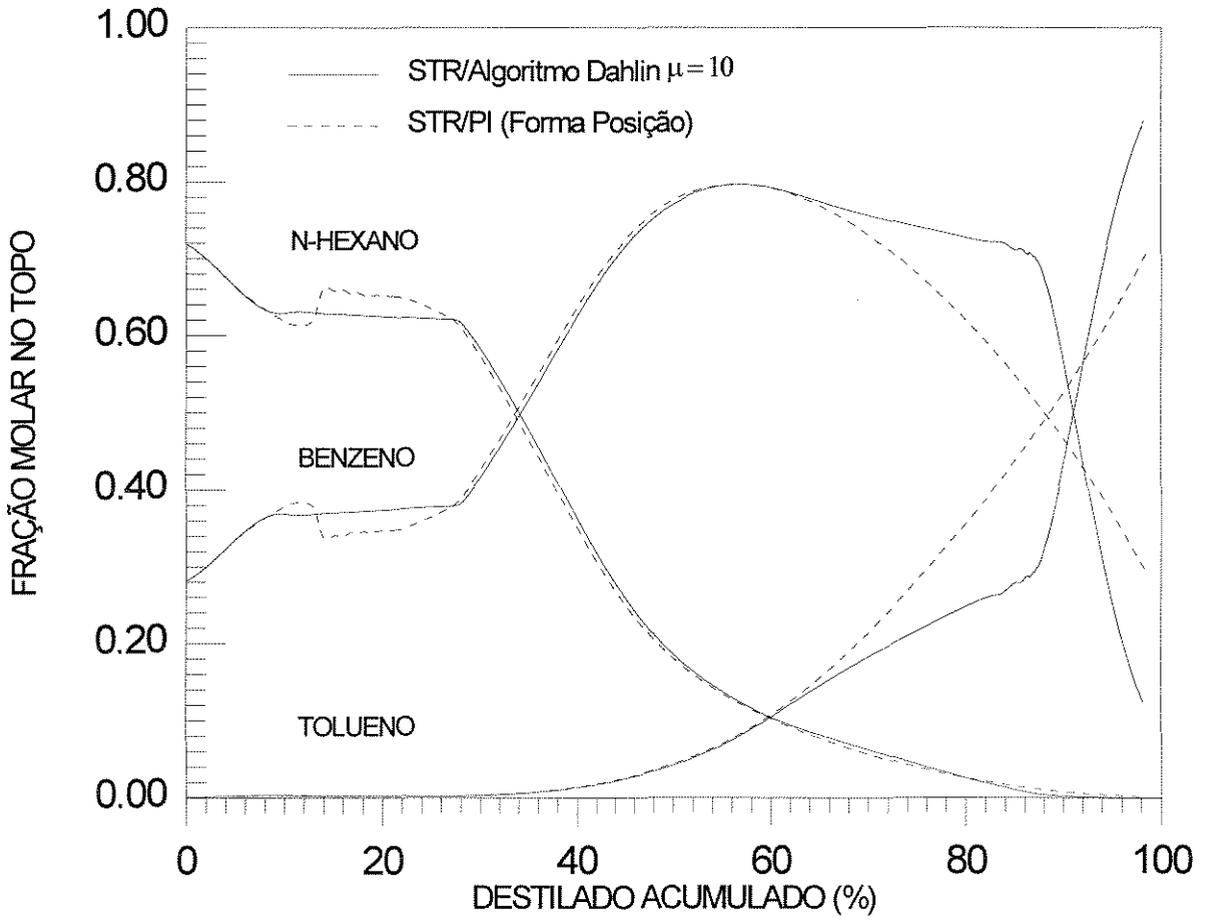


Fig. 3.7a - Comportamento da composição de topo, em malha fechada, para o caso 3.

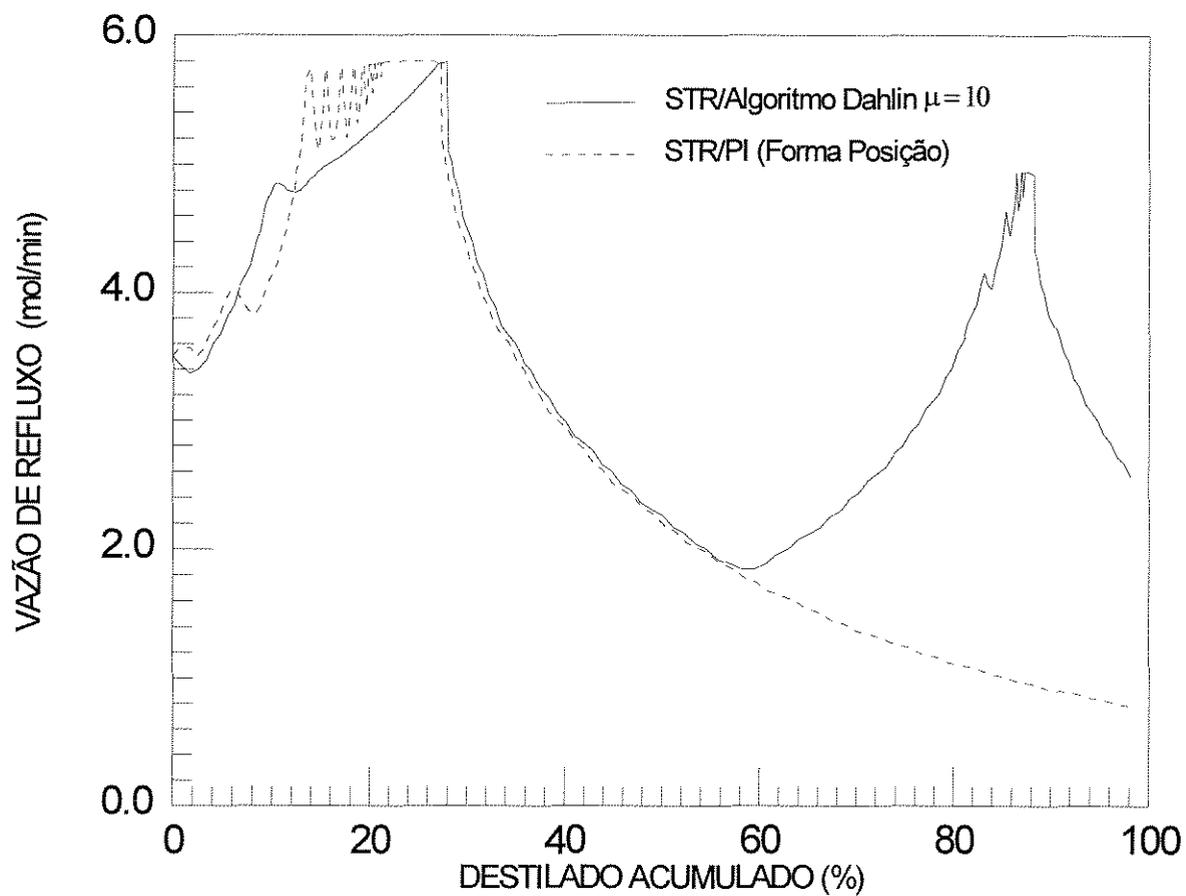


Fig. 3.7b - Comportamento da variável manipulada para o caso 3.

A oscilação ocorrida devida à ação de controle imposta pela lei PI (fig. 3.7b) influi diretamente na quantidade de n-hexano recuperado dentro das especificações: enquanto 52% do n-hexano da carga inicial foi recuperado em 36 minutos sob ação de controle STR/Dahlin, somente 46% foi recuperado em 60 minutos sob ação do STR/PI.

Assim como no caso 2, as figuras 3.7 mostram a demora da reação do PI frente à queda de concentração do segundo componente controlado, enquanto o Algoritmo de Dahlin responde rapidamente aumentando a variável manipulada vazão de refluxo.

A quantidade total recuperada de Benzeno dentro de especificação, em relação à carga inicial, foi de 66% sob ação STR/Dahlin e 50% sob STR/PI.

3.5.4 Caso 4

Para ilustrar a robustez do sistema de controle mais genérico proposto neste capítulo, o STR foi utilizado no controle da separação de um sistema não-ideal, onde uma pureza de 90% seria requerida para o metanol vindo de uma mistura inicial de 60% metanol, 30% etanol e água.

Diante dos bons resultados obtidos anteriormente quando da utilização do STR/Dahlin, é interessante mostrar como seu parâmetro de projeto, μ , afeta a resposta do processo em malha fechada.

Nas figuras 3.8a e 3.8b pode-se verificar que baixos valores de μ acentuam a rapidez da ação de controle, enquanto valores maiores resultam em ajustes mais lentos na variável manipulada.

Sabe-se também que o período de amostragem afeta a qualidade da resposta em malha fechada, conseqüentemente novos valores de parâmetros de projeto do controlador devem ser selecionados se este intervalo for modificado. Da figura 3.8a percebe-se que a maior rapidez de ação de controle provocada por valores mais baixos de μ , no caso do intervalo de amostragem maior, não resultou em melhor resposta da variável controlada. Isto prova a necessidade da escolha, por parte do projetista, dos valores de μ e T que resultem na resposta desejada para o processo em malha fechada.

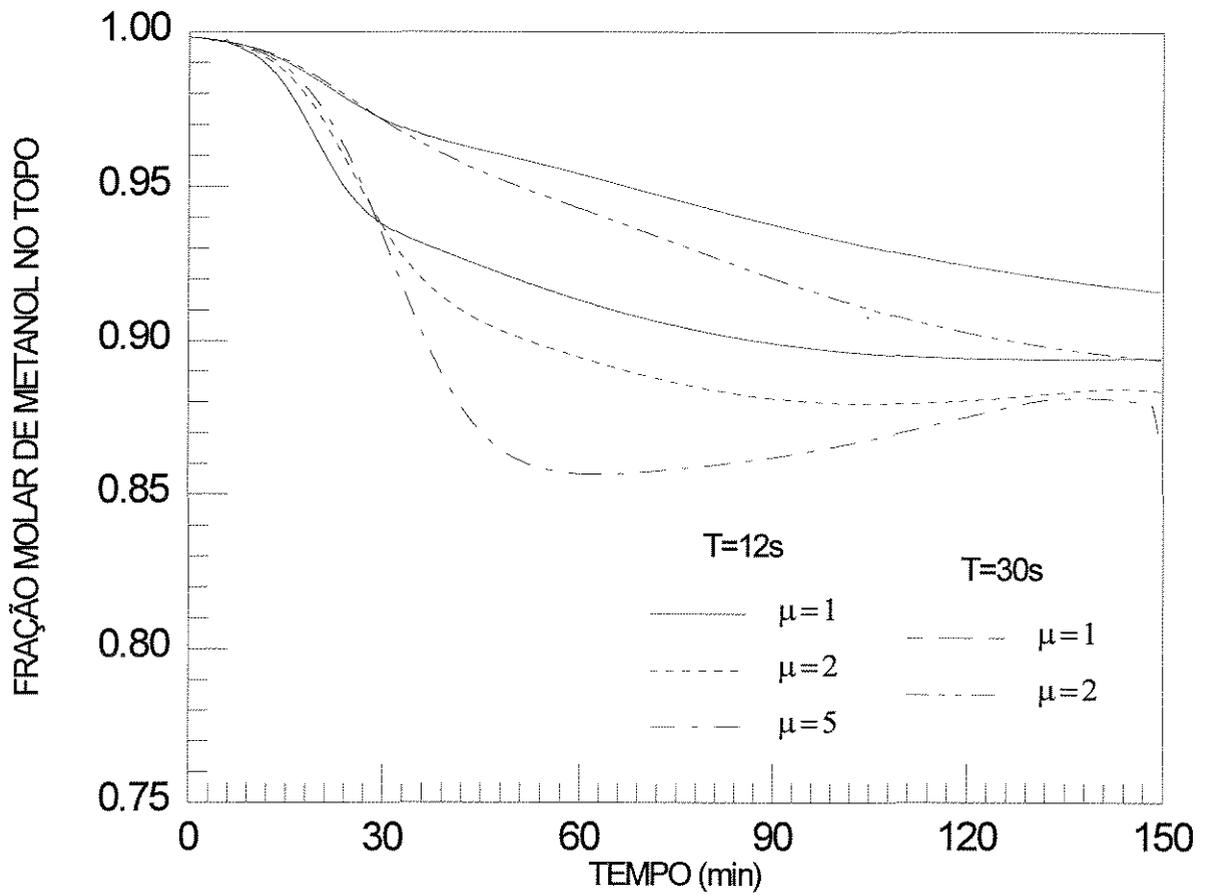


Fig. 3.8a - Comportamento da variável controlada pelo STR/Dahlin com diferentes valores de parâmetro de projeto (μ) e de intervalo de amostragem (T), para o caso 4.

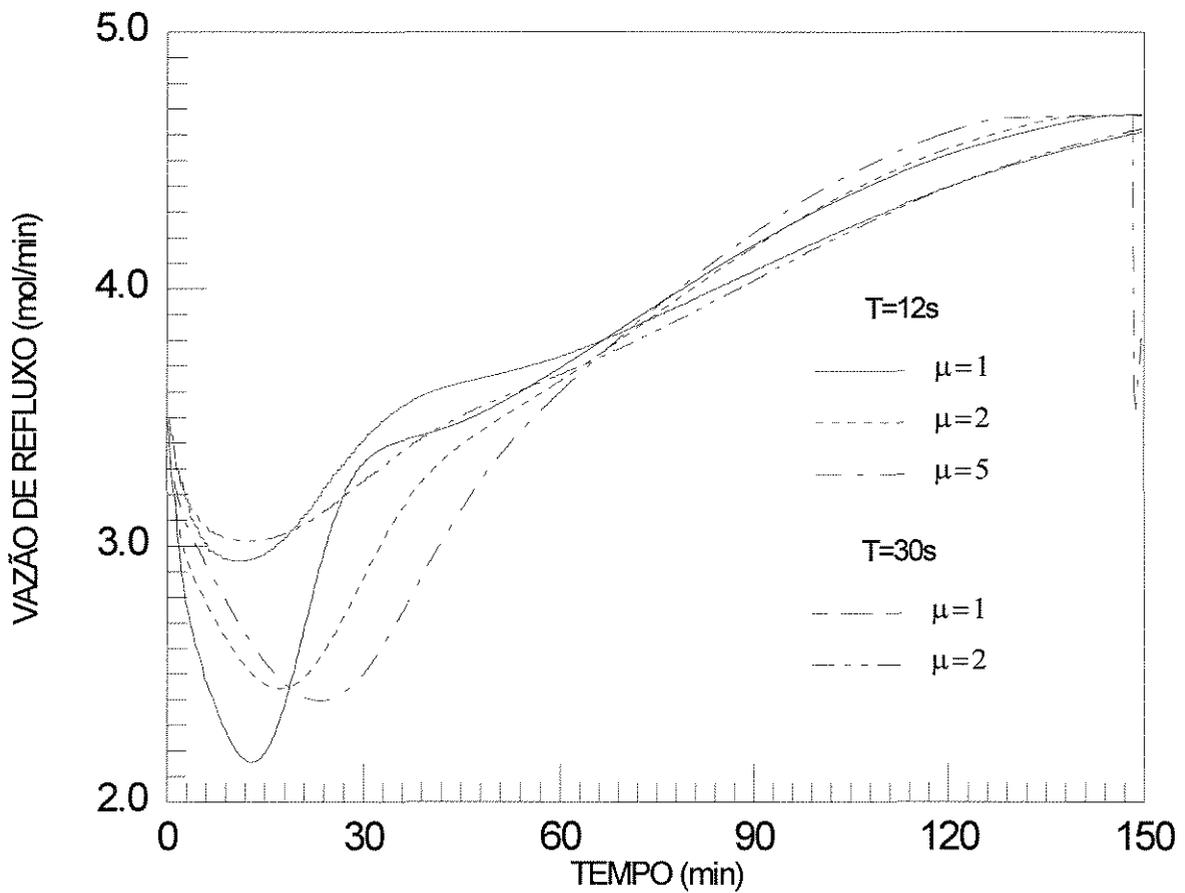


Fig. 3.8b - Comportamento da variável manipulada sob controle do STR/Dahlin, com diferentes parâmetros de projeto, para o caso 4.

Aproveitando o caso em estudo, a figura 3.9 mostra as variações efetuadas no suprimento de calor do refeedor para manter a taxa de evaporação fixa no fundo da coluna. Conforme os componentes mais voláteis se esgotam, o ponto de bolha no refeedor aumenta e uma maior quantidade de calor deve ser suprida ao mesmo. Valores da composição de fundo são amostrados para que se calculasse, em linha, a entalpia de evaporação da mistura e conseqüentemente a taxa de calor necessária para manter o fluxo de vapor. Entretanto, em implementações práticas, medições de pressão no refeedor constituem guias sensíveis para a taxa de requerimento de calor em colunas reais.

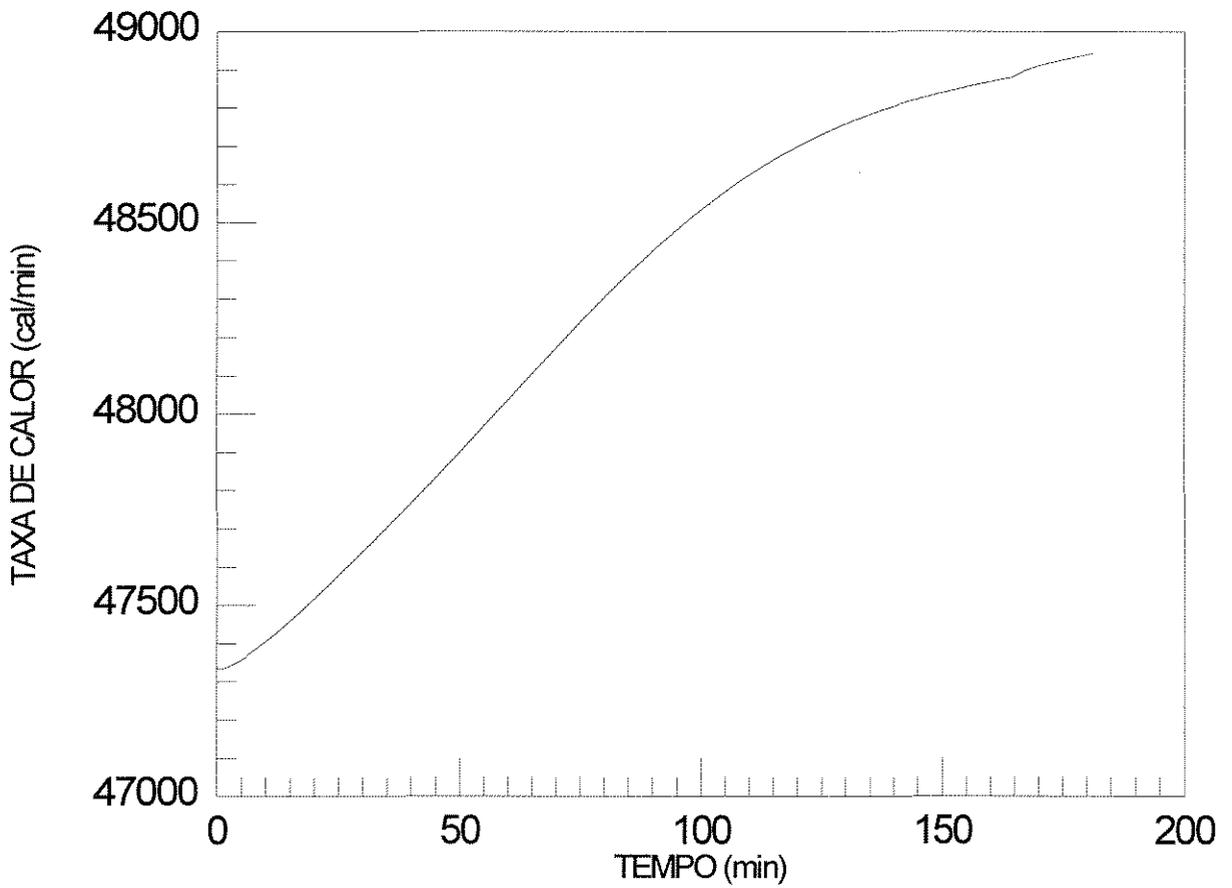


Fig. 3.9 - Variação do calor suprido ao refeedor para manter fixa a taxa de evaporação.

3.6 ANÁLISE

A partir dos resultados obtidos das simulações, verificou-se que o bom desempenho do controlador adaptativo programável (PAC), adequadamente planejado, o torna uma simples e eficiente maneira de automação de colunas em batelada. Além disso, os cálculos efetuados em linha por este controlador são extremamente rápidos, permitindo que um simples computador do tipo PC-XT acoplado à coluna seja suficiente para a implementação do mesmo, sendo que os fatores limitantes dependem somente da velocidade com que as variáveis auxiliares respondem às mudanças do processo.

Entretanto, como dito anteriormente, um dos pontos negativos do PAC é ser uma compensação em malha aberta, ou seja, não há correções que compensem um possível planejamento inadequado. Outro ponto negativo é, sem dúvida, o tempo consumido para o projeto do sistema, que tem de ter seus parâmetros determinados para diversas condições de operação do processo através de exaustivas simulações, além do que para cada configuração do equipamento ou composição da mistura a ser destilada é necessário um projeto diferente.

Porém, o pior de todos os aspectos negativos do PAC, é a necessidade do conhecimento das propriedades dos componentes da mistura a ser destilada para que se possam efetuar as simulações prévias necessárias ao projeto do sistema de controle, já que em muitas aplicações práticas isto é impraticável sendo que normalmente equipamentos em batelada são multiprodutos.

Sendo o controlador auto-ajustável (STR), um controlador totalmente independente das propriedades da mistura a ser separada e até do equipamento a ser controlado - desde que este possa ser representado por uma função de transferência de 1ª ordem com tempo morto - e, principalmente diante do bom desempenho apresentado nos casos estudados, o STR mostra-se como sendo a melhor ferramenta testada neste capítulo para nossos objetivos de controle.

Dentre as leis de controle testadas, o algoritmo PI se mostrou mais sensível quando submetido a estimativas iniciais inadequadas, levando o processo a instabilidade. Quanto à quantidade recuperada de produto dentro da especificação, o algoritmo de Dahlin

foi superior em todos os casos, desde que seus parâmetros de projeto fossem adequadamente supridos.

Apesar da maior complexidade de cálculos a serem efetuados em linha, desde que se forneçam estimativas iniciais adequadas e se limitem as mudanças das variáveis manipuladas para garantia de convergência mais rápida do método de identificação do processo, o STR pode ser normalmente implementado através de um PC-386 acoplado ao processo.

**CAPÍTULO 4 - CONTROLE PREDITIVO COM MODELO
BASEADO EM REDES NEURASIS ARTIFICIAIS.**

4.1 INTRODUÇÃO

Nenhum pesquisador ativo no campo de controle de processos pode hoje negar o crescimento do número de artigos, jornais e conferências dedicado exclusivamente ao tópico “Redes Neurais Artificiais”. Isto é um indicativo claro do atual crescimento de interesse nos conceitos associados a Redes Neurais (RNA) com o desejo de usar algoritmos correspondentes em diversas áreas de aplicações potenciais. Sistemas de Controle constituem atualmente numa destas áreas de aplicação de RNA que vêm sendo agraciadas com diversas publicações específicas recentes desta importante fração do universo da Inteligência Artificial.

Se para causas que aparentem ser similares são esperados efeitos similares, por que não treinar uma máquina para predizer o futuro, baseada em conhecimentos adquiridos no passado e então tomar atitudes mais acertadas no presente?

Neste capítulo será dada uma breve explicação sobre o funcionamento biológico das redes neurais e sobre a essência da ferramenta computacional que imita o comportamento do cérebro humano. Mais detalhadamente serão apresentados a computação neural, os diferentes tipos de conexões entre os neurônios artificiais (topologia) e o procedimento utilizado para o treinamento da RNA.

Um enfoque especial será dado ao histórico da aplicação de RNA's em controle de processos e ao histórico do tipo de controlador escolhido para o desenvolvimento deste trabalho com redes. O diagrama de blocos do controlador, a topologia escolhida para a RNA e o algoritmo de otimização utilizados no cálculo das ações de controle serão apresentados no item que trata da implementação do controlador. Finalmente, alguns resultados e conclusões serão apresentados.

4.2 REDES NEURAIAS

4.2.1 A Rede Neural Biológica

No cérebro humano, o sistema nervoso engloba neurônios que interagem de maneira complexa. Os sensores (cinco sentidos) humanos detectam os estímulos e enviam

estas informações de entrada ("inputs") através dos neurônios ao cérebro. Dentro do cérebro, neurônios são excitados e interagem entre si. Baseado nas informações de entrada, o cérebro então chega a uma solução ou conclusão que é enviada como resposta na forma de informação de saída ("output"). Esta interação entre neurônios é identificada como responsável pelo comportamento inteligente do Ser Humano.

A interpretação neurofisiológica da ativação dos neurônios e de seus sinais de saída envolve pulsos elétricos de diferentes potenciais e sua soma temporária. A ativação envolve pequenos pulsos nas membranas e, os sinais de saída, grandes pulsos nos eixos ou grande potencial de ação.

A diferença de potencial na carga elétrica ocorre entre as partes interna e externa da membrana que envolve o neurônio. Encontrando sinais químicos neurotransmissores, enviados de outros neurônios através de sinapses adjacentes, a membrana varia a intensidade de seu potencial elétrico. As sinapses convertem os sinais neuronais elétricos pulsantes em sinais químicos neurotransmissores.

Segundo Shepherd (1979), os diferenciais de potência, ou pulsos, se acumulam na membrana do eixo Hillock, onde o neurônio se conecta com um de seus eixos (ou ramificação longa). Esta membrana gera um pulso ou potencial de ação se os pequenos pulsos de diferença de potencial, que chegam ao eixo Hillock do neurônio, excederem o valor residual atual de atividade ("Threshold"). Este grande pulso gerado se propaga pelo eixo e suas ramificações, sendo restaurado e amplificado o sinal durante sua propagação até que encontre uma junção sináptica. Os processos sinápticos convertem então os sinais elétricos em sinais químicos neurotransmissores, afetando mais uma vez a diferença de potencial da membrana do neurônio adjacente e assim por diante. O potencial residual de atividade varia com o tempo e assume valores próximos de 40 milivolts.

Ao nível molecular, íons comandam as mudanças nos diferenciais de potência. O fluxo iônico modifica o potencial da membrana do eixo Hillock através da variação da condutância da membrana. Esta variação de condutância envolve seletivamente abertura e fechamento de canais moleculares. Os íons sódio (Na^+) e potássio (K^+) fluem através destes canais moleculares. Sódio e potássio formam um par iônico antagônico: os íons Na^+ são excitadores, isto é, aumentam o potencial da membrana e os íons K^+ são inibidores, ou seja, diminuem o potencial. Outros íons como cálcio e cloreto também

podem afetar a transmissão e recepção dos sinais neuronais. Em neurônios biológicos, o intervalo de amostragem consiste de aproximadamente 10 a 20 milissegundos.

4.2.2 A Essência da Rede Neural Artificial

Uma Rede Neural Artificial (RNA) é uma ferramenta computacional que resultou de pesquisas no campo da Inteligência Artificial (IA) que procurava entender e modelar o comportamento do cérebro humano. Ela difere das aplicações convencionais da IA por utilizar processamento subsimbólico, de forma microscópica, enquanto que as convencionais normalmente se utilizam de processamento simbólico, ou seja, manipulam símbolos, como átomos, e registros de relações causa-efeito para resolver problemas.

A maior parte dos neurologistas acreditam que a verdadeira inteligência vai além do processamento simbólico, macroscópico, que não se importa em verificar como as relações de conhecimento são estabelecidas. Alguma manipulação subsimbólica deve existir e se manifestar eventualmente como comportamento macroscópico (simbólico) e inteligente.

Como dito no item anterior, a interação entre neurônios no cérebro humano, na realidade, não é visível mas é identificada como comportamento inteligente. Com o propósito de desenvolver o mesmo tipo de estrutura para um modelo computacional de comportamento inteligente, neurologistas e pesquisadores de IA propuseram uma rede altamente interconectada de “neurônios”, ou nódulos.

Informações de entrada são colocadas em uma rede de nódulos que interagem matematicamente entre si. Baseado nestas informações, surge um mapeamento do modelo entrada/saída macroscópico esperado, ou seja, as interações entre os nódulos são bem definidas e ajustadas até que as relações entrada/saída desejadas sejam apropriadamente obtidas. Assim sendo, verifica-se que as RNA's se preocupam intimamente com o modo com que as relações se desenvolvem, ao contrário da maior parte das aplicações de IA, como por exemplo os Sistemas Especialistas.

A interconecção de nódulos forma a RNA, conforme mostra a figura 4.1.

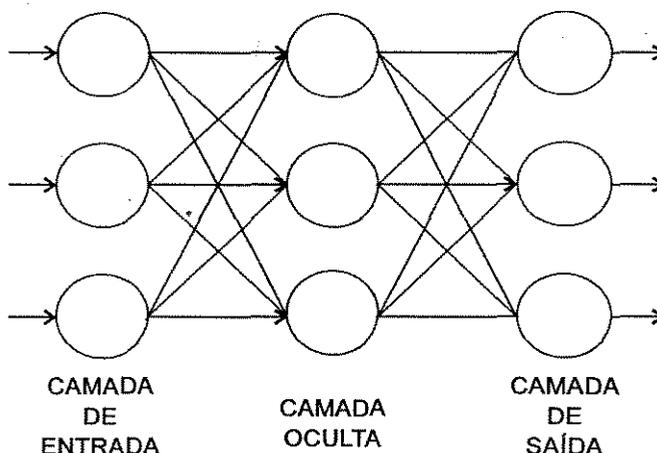


Fig. 4.1 - RNA com apenas uma camada oculta.

Todas as RNA's devem ter uma camada de neurônios de entrada, uma de saída e, no mínimo, uma camada de neurônios escondida. Cada camada é essencial para o sucesso da RNA.

Uma RNA pode ser vista como uma caixa preta que recebe informações de entrada específicas em cada nódulo da camada de entrada e então processa estas informações através de suas interconexões entre os nódulos (ou neurônios) de forma totalmente oculta. Finalmente, a RNA resulta informações finais de saída nos nódulos da camada de saída. Resumindo, a camada de entrada recebe informações de fontes externas e passa estas informações para dentro da RNA para processamento. As camadas escondidas recebem informações da camada de entrada e "em segredo" realizam o processamento destas informações. A camada de saída recebe as informações processadas da RNA e envia os resultados para o receptor externo.

Quando a camada de entrada recebe informações da fonte externa, ela se torna ativada e emite sinais para sua vizinhança. Esta recebe excitação da camada de entrada e mais uma vez emite uma saída para suas vizinhanças. O resultado é um modelo de ativação que se manifesta na camada de saída. Dependendo da força das conexões, sinais podem excitar ou inibir os nódulos.

Uma característica importante das RNA's é que, dentro da rede, o processamento é numérico, embora os resultados se manifestem simbolicamente (daí o nome subsimbólico).

A rede retém informações através da magnitude dos sinais que passam através das conexões dos neurônios com suas vizinhanças.

Sendo numéricas as informações retidas na RNA, esta pode ser usada como uma ferramenta importante de modelagem empírica multivariável e com algumas vantagens aparentes, sendo a principal sua habilidade para adaptação.

Imitando o processo humano de aprendizagem, as RNA's "aprendem" e "resolvem" problemas por tentativa-e-erro. Por exemplo, se uma parte de um equipamento não está operando corretamente, o operador observa seus principais sintomas e recomenda ações corretivas. Baseado nos resultados destas ações, recomenda então outras ações corretivas. Este processo continua até que ele consiga correlacionar adequadamente os sintomas com as ações corretivas e a máquina opere corretamente.

As RNA's se utilizam de algoritmos particulares de treinamento ou aprendizagem, onde iterativamente são ajustados os pesos das conexões entre os nódulos até que os pares desejados de informações entrada/saída sejam obtidos e as relações causa-efeito possam ser estabelecidas. Se as condições mudarem de tal modo que o desempenho do modelo não seja mais adequado, pode-se submeter a RNA a mais um treinamento sob estas novas condições de entrada/saída para corrigir seu desempenho. Assim sendo, pode-se submeter a RNA a uma atualização periódica, resultando num modelo auto-ajustável em linha.

Notando-se que na indústria química a maioria dos equipamentos operam em regime transiente e os que operam em regime estacionário sofrem perturbações ou têm comportamentos não-lineares, as RNA's se constituem ferramentas alternativas de grande importância na simulação e identificação em linha deste tipo de processo.

Para o estudo de colunas de destilação em batelada, onde se verificam variações constantes nas condições de operação e as não-linearidades são evidentes, e para o objetivo de controle com base em predições de comportamento num determinado horizonte de tempo, a possibilidade do uso desta ferramenta da inteligência artificial se deve efetuar com grandes méritos.

4.2.3 A Computação Neural

A base de uma Rede Neural Artificial é o neurônio artificial ou nódulo. A figura 4.2 mostra a anatomia deste elemento de processamento onde a maioria dos cálculos são efetuados.

O primeiro elemento do j -ésimo nódulo é o vetor-entrada A de componentes a_1, a_2, \dots, a_n . O nódulo manipula essas entradas ou atividades, para gerar a saída b_j . Esta saída pode então se tornar parte da entrada de outros nódulos.

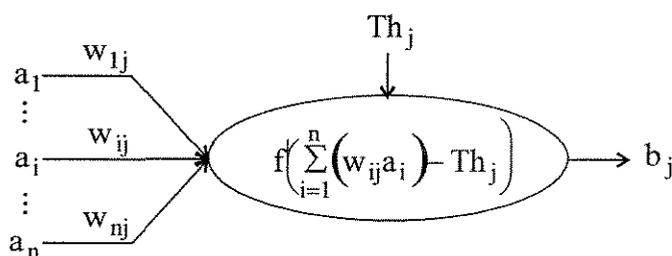


Fig. 4.2 - Anatomia do j -ésimo nódulo (neurônio artificial).

Certamente, os valores dos componentes do vetor-entrada A têm um efeito sobre a saída b do neurônio, mas alguns componentes adicionais do nódulo também afetam b . Um destes componentes é o fator ponderal, w_{ij} , correspondente ao componente a_i de entrada do j -ésimo nódulo. Cada entrada é multiplicada pelo seu fator ponderal respectivo e esta entrada ponderada é então utilizada para os próximos cálculos. Estes fatores ponderais, ou pesos, podem assumir efeitos inibitórios ou excitatórios, representando a função dos íons Na^+ e K^+ na Rede Neural Biológica. Se w_{ij} é ajustado tal que o produto $a_i w_{ij}$ seja positivo (e de preferência grande), a tendência é de excitação do neurônio j . Se $a_i w_{ij}$ for negativo, esta entrada ponderada inibirá o nódulo. Se $a_i w_{ij}$ assumir um valor muito pequeno em relação aos outros sinais (ou entradas ponderadas $a_i w_{ij}$), o efeito será muito pequeno ou nulo sobre o nódulo.

A atividade residual interna do j -ésimo nódulo, Th_j , controla a ativação total do nódulo. O nódulo primeiro calcula o somatório de todas as entradas ponderadas e depois calcula a ativação total pela subtração do valor residual interno:

$$\text{Ativação Total} = \sum_{i=1}^n (w_{ij}a_i) - Th_j \quad (4.1)$$

Se Th_j tem valor grande e positivo, o nódulo tem uma ativação residual interna alta, o que inibe a excitação do mesmo. Ao contrário, se Th_j for nulo ou assumir valores negativos (em alguns casos), o neurônio artificial tem uma baixa ativação residual interna, sofrendo excitação mais facilmente. Se nenhuma ativação interna for especificada, deve-se assumir Th_j nulo.

Verifica-se então que o neurônio artificial realiza seus cálculos baseado em suas informações de entrada. Ele faz o somatório do produto entre os vetores A e W_j , subtrai a ativação residual interna e então passa este resultado para uma forma funcional, $f()$, ou seja:

$$f(W_j.A - Th_j) = f\left(\sum_{i=1}^n (w_{ij}a_i) - Th_j\right) \quad (4.2)$$

Esta forma funcional poderia ser qualquer função como logarítmica, exponencial, raiz quadrada, etc., mas os matemáticos e cientistas da computação afirmam que as funções sigmoidais são vantajosas nesta aplicação particular. A função típica utilizada pode ser representada pela equação (4.3) e vista na figura 4.3.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4.3)$$

Esta função é uma função monotonamente crescente, com valores limitantes de 0 em $x \rightarrow -\infty$ e 1 em $x \rightarrow \infty$. Por causa destes limites, as funções sigmoidais são chamadas de funções de valores residuais. Sendo contínuas e monótonas, resultam normalmente em RNA's bem comportadas, além de propiciarem um treinamento mais rápido e eficiente.

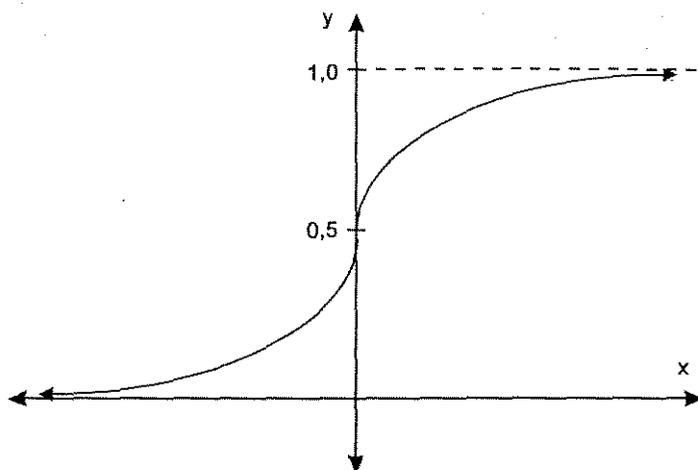


Fig. 4.3 - Função sigmoial.

4.2.4 Topologia da Rede Neural Artificial

A topologia de uma RNA se refere à maneira como os neurônios artificiais estão interconectados e organizados em camadas. Existem três opções de conexões de neurônios, conforme figura 4.4.

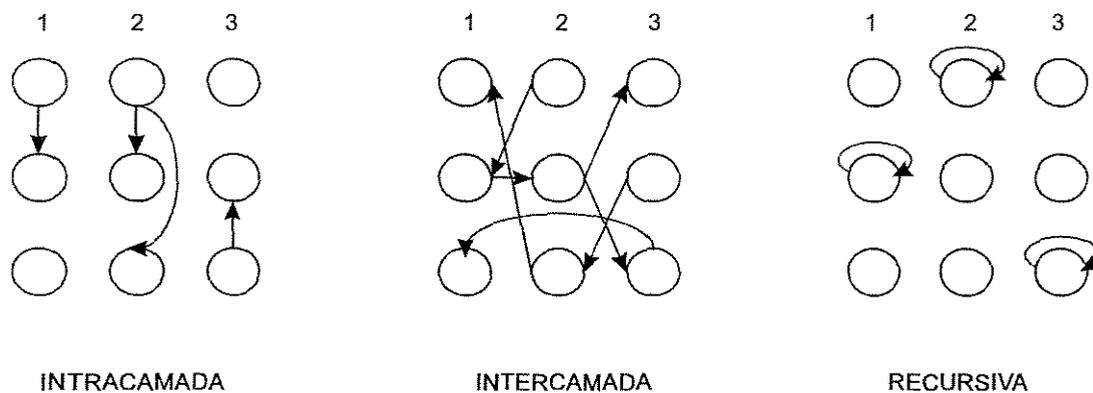


Fig. 4.4 - Opções de conexões entre neurônios em uma RNA.

Em conexões intracamadas, as saídas de um neurônio alimentam outros neurônios de uma mesma camada. Em conexões intercamadas, as saídas de um neurônio em uma camada alimentam nódulos de outras camadas. Finalmente, em conexões recursivas, a saída de um nódulo alimenta ele próprio.

Segundo Quantrille e Liu (1991), a conexão intercamadas é particularmente importante para as aplicações de engenharia. Dentro deste esquema de interconexão, existem duas opções: as conexões por retroalimentação e por alimentação direta, conforme figura 4.5.

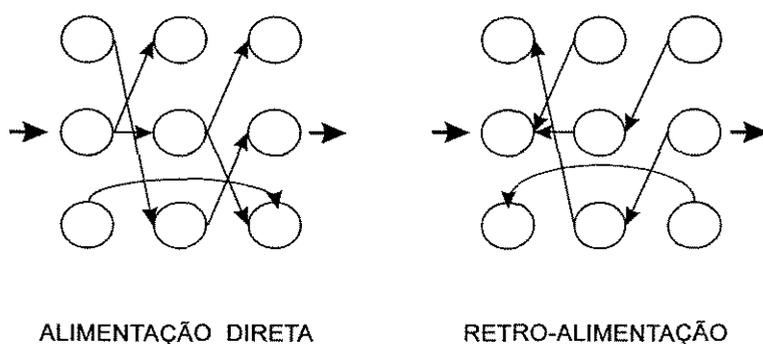


Fig. 4.5 - Opções de conexões intercamadas.

A topologia ideal deve ser escolhida de acordo com a natureza do problema a ser resolvido. Segundo os autores acima, para modelagens dinâmicas de equipamentos, o que se quer é mapear uma resposta baseada em informações de entrada e para tanto deve ser utilizada a conexão intercamada com alimentação direta. Uma Rede Neural Artificial com esta topologia também pode ser chamada de “Perceptron” e, por ser menos complicada, tem um número maior de teorias matemáticas relacionadas.

4.2.5 Treinamento da RNA

Para treinar RNA's, devem-se ajustar os fatores ponderais, W , até que a resposta (informação de saída) calculada, baseada em informações de entrada, alcance a relação causa-efeito desejada. Se o processo de ajuste de pesos é feito através de tentativa-e-erro, esta etapa na verdade se denomina aprendizagem (“learning”).

A aprendizagem, ou treinamento, é a etapa onde se codificam as informações. Um sistema aprende uma regra se ele codifica esta regra em sua estrutura. Portanto, esta estrutura deve mudar durante o aprendizado desta informação.

Sendo numerosas as variáveis ajustáveis numa RNA, o ajuste sistemático das mesmas para que a rede forneça resultados apropriados, e de maneira eficiente, é uma tarefa desafiadora. A rede resultante desta fase de treinamento tem que ser estável e convergente, ou seja, tem que garantir respectivamente um mapeamento para qualquer par entrada/saída e um erro pequeno entre os resultados do mundo real e os preditos pela rede.

Há diferentes maneiras de se treinar uma RNA. A maneira mais usual é o Aprendizado por Correção do Erro ("Error-correction Learning"). Esta é uma forma de aprendizado supervisionado - a cada instante de amostragem se sabe a resposta desejada - onde se usa a diferença entre a resposta desejada e a vigente para corrigir o comportamento da rede e continuar o treinamento. Assim sendo, o vetor-fator-ponderal, W , é corrigido de acordo com a variação do vetor-erro, E , da resposta da rede. Este vetor E tem n componentes, onde n é o número de nódulos da camada de saída da rede.

Para que este tipo de aprendizado pudesse não se restringir somente a redes compostas por duas camadas (uma de entrada e uma de saída), uma técnica chamada Retropropagação ("Backpropagation") foi desenvolvida, por matemáticos e cientistas da computação, somente para RNA's com conexões intercamadas com alimentação direta ("perceptrons" - figura 4.6).

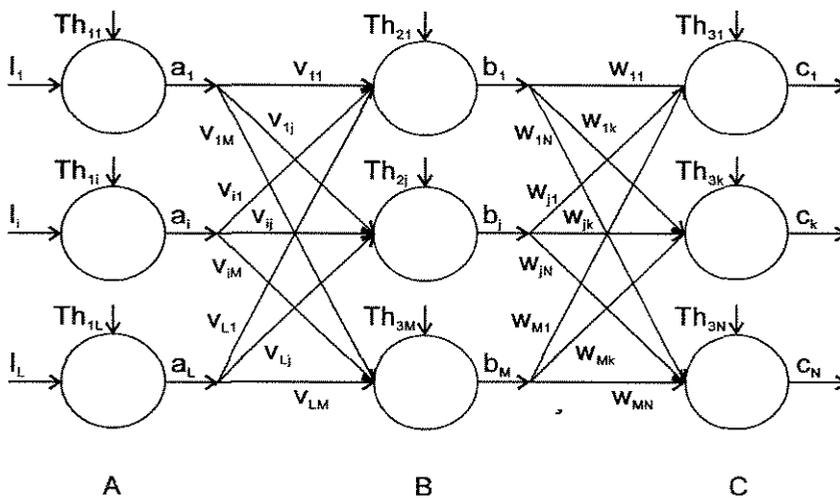


Fig. 4.6 - "Perceptron" RNA com 3 níveis.

Sendo esta topologia a ideal para o objetivo da implementação do Controlador Preditivo com Modelo RNA (onde uma RNA será utilizada para simular, em linha, uma coluna de destilação em batelada), a técnica de Retropropagação foi a escolhida para a etapa de treinamento da rede.

Tomando como exemplo o “perceptron” composto de três camadas da figura 4.6, verifica-se que a camada de entrada A tem L neurônios artificiais, ou seja, $a_1, a_2, \dots, a_i, \dots, a_L$. Alimentando esta camada tem-se o vetor-entrada I . A camada escondida B tem M nódulos ($b_1, b_2, \dots, b_j, \dots, b_M$) e a camada de saída C, N nódulos ($c_1, c_2, \dots, c_k, \dots, c_N$). Apesar da figura 4.6 estar representada com L, M e N iguais, na prática isto não é necessário, ou seja, $L \neq M \neq N$ é totalmente aceitável.

O fator ponderal da interconecção entre o i -ésimo nódulo da camada A e o j -ésimo da camada B é chamado de v_{ij} , e entre o j -ésimo nódulo da camada B e o k -ésimo da camada C, de w_{jk} . Cada nódulo possui um residual de atividade Th .

Uma grande dificuldade associada com o uso de algoritmos de Retropropagação é o tempo requerido para treinar a RNA. Segundo Quantrille e Liu (1991), pesquisadores têm investigado diversos procedimentos de treinamento numa tentativa de agilizar o processo de aprendizagem, e uma técnica que tem sido bastante aceita é o Aprendizado por Gradiente Descendente (“Gradient-descent Learning”).

4.2.5.1 Algoritmo de Treinamento Regra de Delta Generalizada (GDR)

O Método conhecido por Regra de Delta Generalizada (“Generalized Delta-Rule” ou GDR) é um método iterativo por gradiente descendente, que minimiza o quadrado do erro e se utiliza de uma técnica conhecida por “momentum” para agilizar o processo de treinamento. “Momentum” é um peso-extra, adicionado aos fatores ponderais, enquanto estes são ajustados. Acelerando a variação destes fatores ponderais, a velocidade de treinamento é aumentada e, por este motivo, o GDR foi o método escolhido para o desenvolvimento deste trabalho.

Além da utilização da técnica do “momentum”, o GDR se diferencia de outros algoritmos de Retropropagação pela presença da função “bias” ao invés dos valores

internos de residuais de atividade (“Threshold”). Estes valores residuais internos (Th_{1i} , Th_{2j} , Th_{3k}) se transformam na função “bias” quando são adicionados, ao invés de subtraídos, ao somatório nodal. Quando usados como função “bias”, os valores de Th não devem ser atualizados durante o processo de treinamento e, no GDR, devem assumir os seguintes valores fixos: $Th_{1i} = 0$, e $Th_{2j} = Th_{3k} = 1$.

Segue abaixo uma descrição detalhada do algoritmo Regra de Delta Generalizado (GDR) aplicado ao “perceptron” mostrado na figura 4.6:

1º passo: Assumir randomicamente valores entre 0 e 1 para os pesos v_{ij} e w_{jk} . Para o GDR, os valores residuais internos devem seguir as seguintes regras: todo residual dos nódulos da camada de entrada devem ser iguais a zero, isto é, $Th_{1i} = 0$; todo residual dos nódulos das camadas escondidas e da camada de saída devem ser iguais à unidade, isto é, $Th_{2j} = Th_{3k} = 1$.

2º passo: Introduzir o vetor-entrada I_i na RNA, e calcular a saída da 1ª camada, a_i , de acordo com as equações:

$$x_{1i} = I_i - Th_{1i} = I_i - 0 = I_i \quad (4.4)$$

$$a_i = \frac{1}{1 + e^{-x_{1i}}} \quad (4.5)$$

3º passo: Conhecida a saída da 1ª camada, calcular a saída da segunda, b_j , usando a equação:

$$b_j = f\left(\sum_{i=1}^L (v_{ij} a_i) + Th_{2j}\right) \quad (4.6)$$

onde $f()$ é a função sigmoidal e $Th_{2j} = 1$ está sendo adicionado ao somatório (ao invés de subtraído), agindo assim como função “bias”.

4º passo: Conhecida a saída da 2ª camada, calcular o resultado da camada de saída, c_k , de acordo com a equação:

$$c_k = f\left(\sum_{j=1}^M (w_{jk} b_j) + Th_{3k}\right) \quad (4.7)$$

sendo $f(\cdot)$ a função sigmoidal e $Th_{3k} = 1$.

5º passo: Continuar etapas 1 a 4 para P número de pares de vetores entrada/saída. Calcular o erro quadrático total, ε , de acordo com a seguinte equação:

$$\varepsilon = \sum_{p=1}^P \sum_{k=1}^N (d_k^p - c_k^p)^2 \quad (4.8)$$

onde N é o número de nódulos da camada de saída, d_k^p é o valor desejado (conhecido) da saída do k-ésimo nódulo do p-ésimo par entrada/saída para treinamento e c_k^p é o valor obtido de saída do k-ésimo nódulo do p-ésimo treinamento da RNA.

6º passo: Calcular o termo de declínio do gradiente, δ_{3k}^p , do k-ésimo neurônio da camada de saída para o p-ésimo par de treinamento da rede. Usar a seguinte equação:

$$\delta_{3k}^p = (d_k^p - c_k^p) \frac{\partial f}{\partial x_{3k}} \quad (4.9)$$

onde f é a função sigmoidal representada pela equação (4.3) e sua derivada parcial em relação a x_{3k} é dada por:

$$\frac{\partial f}{\partial x_{3k}} = \frac{e^{-x_{3k}}}{(1 + e^{-x_{3k}})^2} \quad (4.10)$$

Deve-se notar que x_{3k} é a soma das entradas ponderadas do k -ésimo neurônio da camada de saída, ou seja, para a p -ésima sessão de treinamento tem-se:

$$\dot{x}_{3k}^p = \sum_j w_{jk}^p b_j^p + Th_{3k}^p \quad (4.11)$$

7º passo: Calcular o termo de declínio do gradiente, δ_{2j}^p , do j -ésimo neurônio da camada escondida. Usar a seguinte equação:

$$\delta_{2j}^p = \left(\sum_k \delta_{3k}^p w_{jk}^p \right) \frac{\partial f}{\partial x_{2j}} \quad (4.12)$$

onde o subscrito k indica o nóculo da camada de saída; x_{2j} é definido por:

$$x_{2j}^p = \sum_i v_{ij}^p a_i^p + Th_{2j}^p \quad (4.13)$$

e a derivada parcial da função sigmoideal em relação a x_{2j} é dada por:

$$\frac{\partial f}{\partial x_{2j}} = \frac{e^{-x_{2j}}}{(1 + e^{-x_{2j}})^2} \quad (4.14)$$

8º passo: Conhecendo δ_{2j}^p para a camada escondida e δ_{3k}^p para a camada de saída, calcular as variações dos fatores ponderais usando as equações abaixo:

$$\Delta v_{ij}^p = \eta \delta_{2j}^p a_i^p + \alpha \Delta v_{ij}^{p-1} \quad (4.15)$$

$$\Delta w_{jk}^p = \eta \delta_{3k}^p b_j^p + \alpha \Delta w_{jk}^{p-1} \quad (4.16)$$

onde η é a taxa de aprendizado e α é o coeficiente de “momentum”. Como mencionado anteriormente, “momentum” é simplesmente um peso-extra usado para agilizar a etapa de treinamento e tem seu valor normalmente restrito entre 0 e 1. Assim sendo, os termos de “momentum”, $\alpha\Delta w_{jk}^{p-1}$ e $\alpha\Delta v_{ij}^{p-1}$, são frações da variação do fator ponderal sofrida na iteração anterior.

9º passo: Atualizar os pesos de acordo com as equações:

$$w_{jk}^p = w_{jk}^{p-1} + \Delta w_{jk}^p \quad (4.17)$$

$$v_{ij}^p = v_{ij}^{p-1} + \Delta v_{ij}^p \quad (4.18)$$

onde v_{ij}^p é o fator ponderal da conexão entre o i -ésimo elemento da camada de entrada e o j -ésimo elemento da camada escondida e w_{jk}^p é o peso da conexão entre o j -ésimo elemento da camada escondida e o k -ésimo elemento da camada de saída, ambos na iteração p .

10º passo: Repetir os passos 2 a 9 para todos os pares entrada/saída para treinamento até que o erro quadrático seja zero ou suficientemente baixo.

Das etapas descritas acima, pode-se verificar que o GDR primeiramente calcula o erro de saída e então gera novos valores para os fatores ponderais baseado neste erro e no gradiente da função sigmoideal. O algoritmo atualiza os nódulos de saída e propaga de volta para as camadas mais próximas até encontrar a camada de entrada.

O fluxograma relativo a este algoritmo é mostrado na figura 4.7 e sua implementação é descrita, em linguagem C, no ANEXO 6 deste trabalho.

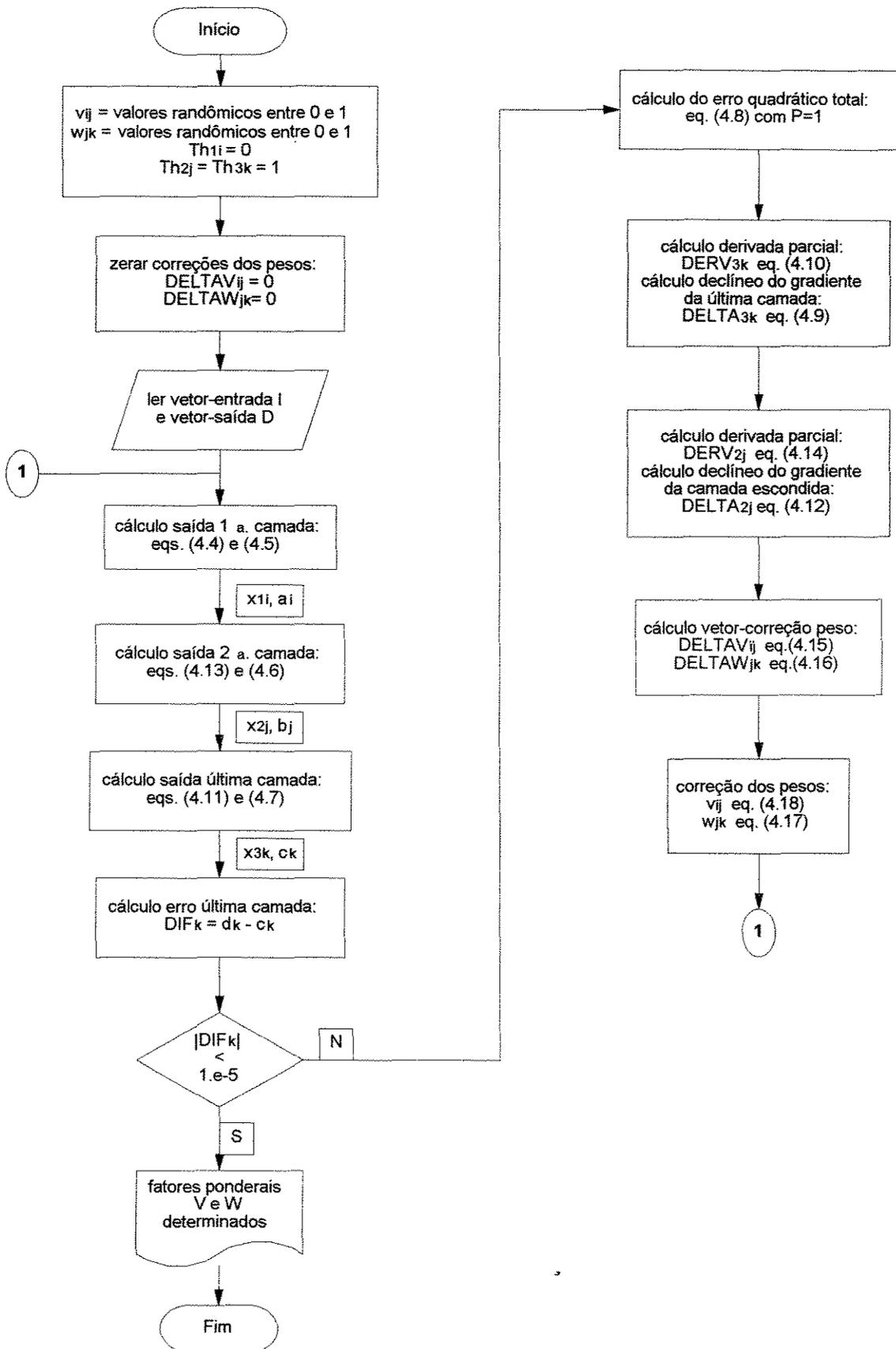


Fig. 4.7 - Algoritmo de treinamento Regra de Delta Generalizado (GDR).

4.3 RNA's EM CONTROLE DE PROCESSOS - UM BREVE HISTÓRICO

As aplicações de RNA's na engenharia química têm aumentado significativamente desde o fim da década de 80, quando Hoskins e Himmelblau (1988) publicaram um dos primeiros artigos de aplicação prática de RNA's nesta área.

Dentro da engenharia química, a área de controle de processos tem sido a mais popular das aplicações de RNA's. Sendo o modelo do processo muito utilizado em técnicas avançadas de controle de processos, a área de modelagem dinâmica pode ser considerada como um subconjunto destas aplicações.

Bhat e McAvoy (1990) aplicaram uma RNA na modelagem dinâmica de um CSTR com controle de pH. O CSTR reagia ácido acético com hidróxido de sódio formando acetato de sódio. Como o fluxo de hidróxido de sódio sofria variações, o pH do meio reacional também variava e obviamente de modo não-linear. Os pares de informações entrada/saída da RNA eram compostos por informações atuais (última amostragem) ou informações atuais mais informações anteriores (últimas 3, 4 ou 5 amostragens) de valores de pH e fluxo de hidróxido, atribuindo característica de janela móvel nas camadas externas da rede. O intervalo de amostragem era de 24 segundos e o número de neurônios artificiais da primeira camada era ajustado de acordo com o número de amostragens incluídas na entrada da rede. A saída da RNA treinada era o valor predito de pH. Bhat e McAvoy usaram ainda uma técnica tradicional de modelagem dinâmica e compararam as duas aproximações. Os resultados demonstraram que a RNA predizia futuros valores de pH mais precisamente, provavelmente porque a RNA modelava melhor as não-linearidades do processo.

Bhat *et al.* (1990) estudaram outro exemplo de modelagem, a modelagem estacionária de um CSTR com reações em série. Eles puderam prever resultados de menos de 2% de raiz do erro quadrático médio. Mas, este simples exemplo levou 10.000 iterações para completar o treinamento da rede, provando a necessidade de aperfeiçoamento dos algoritmos de treinamento das RNA's.

O uso de um controlador baseado em RNA's para monitoramento da temperatura interna de um CSTR encamisado foi demonstrado por Hoskins e Himmelblau (1990). Para tanto foi utilizado um esquema com duas RNA's: uma para avaliação da

aceitabilidade do desempenho do reator e outra para cálculo da ação de controle baseada em predições de comportamento do processo provenientes da primeira rede.

Uma aplicação interessante de RNA na modelagem da cinética de reações eletroquímicas pode ser encontrada em Hudson *et al.* (1990).

Já Psychogios e Ungar (1990) usaram RNA's para o que eles mesmos chamaram de controle baseado em modelo ("Model-based Control") direto e indireto. Em controle direto, a RNA é treinada com dados de entrada/saída do processo. Dados os estados atual e desejado do processo, a RNA ajusta sua saída conduzindo o processo ao estado desejado. Portanto, em controle direto, o controlador RNA toma a ação corretiva baseado no desvio do processo.

No controle indireto, o controlador funciona de maneira preditiva. Dado o estado atual do processo e a saída atual do controlador, a RNA prediz a resposta do processo num determinado horizonte de tempo. Sendo assim, no controle indireto, a RNA age como um modelo do processo ou preditor.

Os mesmos autores compararam as aproximações baseadas em RNA's para controles direto e indireto usando duas estratégias de controle com modelo: Controle com Modelo Interno (IMC) e Controle Preditivo com Modelo (MPC). O IMC é uma estratégia de controle avançado por retroalimentação que usa um modelo em linha dentro da malha de controle, conforme figura 4.8. O ajuste do controlador é feito diretamente baseado no modelo do processo (Seborg *et al.*, 1989).

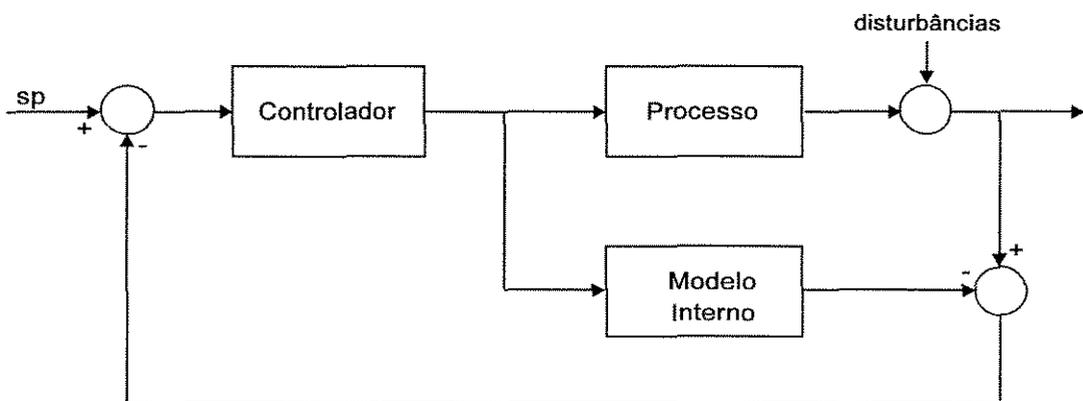


Fig. 4.8 - Diagrama de blocos do Controlador com Modelo Interno (IMC).

O MPC usa um modelo para prever a resposta do processo num determinado horizonte de tempo. Baseado nestas predições, o MPC pode ajustar numerosas variáveis, como os parâmetros do controlador, o tempo de amostragem e os horizontes de predição e controle. Em contrapartida, a estratégia MPC é computacionalmente intensa e exige computadores mais velozes para ser implementada em tempo real.

Psichogios e Ungar usaram RNA's para ambas estratégias IMC e MPC. Para treinar a RNA eles usaram um método baseado no gradiente descendente, com dados tirados diretamente do processo real. O IMC mostrou boa rejeição às perturbações e nenhum desvio em relação ao estado estacionário ("offset"), mas a resposta obtida era oscilatória. Já para o MPC, onde uma computação intensa foi requerida, o controlador regulou bem o processo e não exibiu oscilações. Para fazer um teste, os autores ainda aplicaram IMC e MPC com uma RNA treinada com dados gerados de um modelo linear representativo do processo e não com dados diretos do mesmo, resultando numa RNA linear. Ambas estratégias não tiveram sucesso no controle do processo, aparecendo grandes oscilações na resposta e desvios finais substanciais. Deste modo, os autores puderam concluir que apesar da rapidez de treinamento da RNA linear - em contrapartida ao demorado e trabalhoso treinamento não-linear - os resultados da implementação do sistema de controle linear foram inaceitáveis.

Para indicar o potencial da metodologia de controle de processos baseado em Redes Neurais Artificiais, Willis *et al.* (1992) implementaram um controlador preditivo com modelo RNA a um sistema não-linear de destilação binária contínua.

Chamando este controlador de DNC ("Dynamic Network Control"), os autores aplicaram-no a uma simulação não-linear de uma coluna piloto de 10 estágios instalada na Universidade de Alberta, Canadá. Tendo como alimentação no 4º prato uma mistura de 50% em peso de metanol-água, o objetivo seria manter uma composição de 5% em peso de Metanol no produto de fundo, manipulando o fluxo de vapor no refeedor. Caracterizando o sistema como múltipla-entrada-única-saída (MISO), a composição de topo era controlada através de um PI que manipulava a razão de refluxo. Ambas composições assumiam-se perfeitamente inferenciadas a partir de medições de temperatura e pressão. O desempenho do esquema de controle preditivo proposto foi então comparado

com o desempenho de um controlador PI convencional e uma estratégia linear de controle preditivo. O tempo de amostragem em todos os casos foi de 1 minuto.

Para estabelecer a relação entre a composição do produto de fundo (variável controlada) e o fluxo de vapor (variável manipulada) foram usados distintamente um modelo linear e uma RNA, ambos levando em consideração os efeitos da variação do refluxo (interações) e da alimentação (distúrbância). No modelo linear, uma função transferência de 1^a ordem com tempo morto foi utilizada, enquanto no modelo não-linear a RNA tinha uma estrutura do tipo (3-4-4-1), ou seja, 3 neurônios na camada de entrada, duas camadas escondidas com 4 neurônios cada e uma camada de saída com apenas 1 neurônio. Depois da identificação dos respectivos modelos, eles foram incorporados dentro do esquema de controle proposto, gerando respectivamente um controlador preditivo linear e um não-linear. Os horizontes de predição e de controle para ambos controladores assumiram valores respectivos de 7 e 1.

Quando sujeitos a perturbações na alimentação, os três controladores (PI, preditivo linear e preditivo não-linear) foram comparados e o controlador PI foi o que demonstrou pior desempenho, com um IAE (Integral do Erro Absoluto) de 78,1. Tendo em mente que o modelo linear usado no Controlador Preditivo Linear levava em consideração as interações e as perturbações na sua formulação, um IAE de 62,1 não representou melhora significativa em relação ao PI devidamente sintonizado. O melhor desempenho resultou do Controlador Preditivo Não-linear, cujo IAE foi de 5,7. Este comportamento bastante superior foi atribuído principalmente à habilidade da RNA em representar mais precisamente as características dinâmicas do processo. Portanto, já que o desempenho dos controladores baseados em modelos é diretamente dependente da precisão do modelo utilizado, um controle muito mais efetivo foi resultante da Rede Neural Artificial.

Procurando demonstrar a aplicabilidade deste Controlador Preditivo Não-linear em problemas de múltiplas-entradas-múltiplas-saídas (MIMO), Willis *et al.* ainda utilizaram o simulador citado acima para implementar, concomitantemente, um sistema de controle de composição de topo e fundo da coluna. O objetivo de controle seria então seguir uma série de modificações do “set-point” na malha de controle de fundo, enquanto simultaneamente manter a composição do produto de topo em 95% de metanol. Com a metodologia do Controle Preditivo Não-linear, um IAE de 3,0 foi encontrado para a

composição de topo e 22,24 para a composição de fundo, comparados respectivamente aos 11,52 e 31,58 obtidos com um PI para o topo e outro para o fundo. Para a malha de fundo, apesar da proximidade destes valores de IAE, a resposta em malha fechada para o preditivo se mostrou bem menos oscilatória. Concluem os autores que estes resultados se devem à capacidade de desacoplamento de malhas do controlador multivariável DNC. A habilidade da RNA em prever os efeitos das perturbações causadas pela malha de fundo na saída da malha de topo, permite uma regulação mais eficaz da composição do produto de topo.

Ainda neste artigo publicado por Willis *et al.*, a utilização de RNA's é sugerida para inferências de variáveis controladas de difícil medição. Apesar dos bons resultados obtidos por estimadores adaptativos lineares, que proporcionam inferências rápidas destas variáveis, os autores sugerem que a habilidade da RNA em capturar características não-lineares deve melhorar o desempenho destes estimadores.

A eficácia da estratégia de inferência não-linear foi demonstrada por Guilandoust *et al.* (1987), onde algoritmos adaptativos não-lineares eram usados para inferir estimativas da variável controlada em reguladores por retroalimentação. No trabalho de Willis *et al.*, a praticidade de esquema de controle inferencial baseado em RNA's é explorada na aplicação à fermentação industrial para produção de penicilina G através do *Penicillium chrysogenum*. Deste modo, afirmam os autores, RNA's exibem grande potencial como "soft-sensores", isto é, sensores baseados em "software" ao invés de "hardware".

Uma revisão do papel das Redes Neurais Artificiais no universo da modelagem, identificação e controle de sistemas não-lineares foi publicada por Hunt *et al.* (1992). Aplicações de uma variedade de arquiteturas de RNA's em controle de sistemas foram revisadas por eles.

Ainda na área de controle de processos, porém com outro enfoque, descrevem-se aplicações de RNA's na interpretação qualitativa de dados de processo e na detecção de falhas em sensores, como descrito a seguir.

Whitely e Davis (1990) descreveram o uso de RNA's para interpretação qualitativa de dados quantitativos de processo. Embora em controle de processos se possam utilizar sistemas baseados em conhecimento específico (Sistemas Especialistas), uma limitação importante é que a informação processada por estes sistemas é geralmente

simbólica e não numérica. Entretanto, os dados de processo são numéricos e não simbólicos e transformar dados numéricos em abstrações simbólicas não é uma tarefa fácil. Sendo assim, os autores citados acima testaram a utilização de RNA's para converter dados quantitativos de uma planta em interpretação qualitativa. Especificamente, eles usaram uma RNA para identificar quando uma variável de processo exibia comportamento cíclico com o tempo.

Naidu *et al.* (1990) introduziram o uso de RNA's para detecção de falhas em sensores. Um módulo de diagnóstico no sistema de controle detecta falhas nos atuadores e nos sensores do sistema.

Diferenças entre o modelo do processo e o processo real podem acionar alarmes falsos, indicando indevidamente falhas nos sensores. Se então se relaxa o critério de acionamento dos alarmes, falhas críticas nos sensores podem passar despercebidas. Conseqüentemente, um dos maiores problemas na detecção de falhas nos sensores é a inexatidão dos modelos de processo utilizados.

Naidu *et al.*, preocupados em solucionar este problema, sugerem o uso de RNA's para reconhecer problemas típicos causados por verdadeiras falhas nos sensores e os causados por diferenças entre o modelo e o processo real, ou por ruído, ou por perturbações. Eles usaram Retropropagação para treinamento da rede e compararam o desempenho da RNA com dois outros algoritmos de detecção de falhas em sensores. Aplicando estes algoritmos a uma estrutura de controle IMC para um processo estacionário, de primeira ordem, sujeito a grandes imprecisões no modelo interno, ficou demonstrado que a RNA resultou numa predição mais exata de falhas nos sensores do que os outros algoritmos de detecção, além de ser mais rápida.

Estendendo o estudo nesta área de detecção de falhas em sensores através de RNA's, Yao e Zafiriou (1990) demonstraram uma maneira de se aumentar a eficiência do método eliminando nódulos redundantes da rede neural.

4.4 CONTROLE PREDITIVO COM MODELO

4.4.1 Conceito

O Controle Preditivo com Modelo (MPC) é uma família de controladores que faz uso direto de um modelo explícito do processo, completamente identificável separadamente do controlador. As metodologias de projeto de controladores baseadas no conceito do MPC têm sido bastante aceitas em aplicações industriais e vêm sendo cada vez mais estudadas academicamente. A razão de tal popularidade provavelmente se deva à sua capacidade de projeto de sistemas de controle objetivando obtenção de ótimo desempenho, além da capacidade de operação sem intervenções de especialistas por grandes períodos.

O nome “Controle Preditivo com Modelo” vem da maneira como a lei de controle é calculada. No instante de tempo atual, k , o comportamento do processo num horizonte de predição, N_p , é levado em consideração (figura 4.9). Usando um modelo, a resposta do processo às modificações da variável manipulada são previstas e tais variáveis são selecionadas de maneira que a resposta predita obtenha determinadas características desejadas. Somente a primeira variação da variável manipulada é implementada. No intervalo de tempo $k+1$, os cálculos são então repetidos e o horizonte movido de um intervalo de tempo.

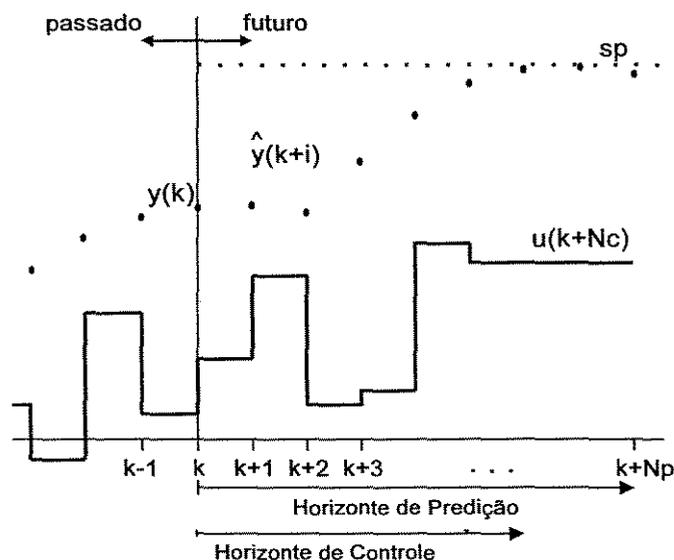


Fig. 4.9 - Horizonte móvel do Controle Preditivo com Modelo.

Para se quantificar a diferença entre a resposta predita e a trajetória de referência, uma função objetivo é utilizada. Tipicamente esta função critério é uma função de \hat{y} , sp e u , ou seja, da variável controlada predita, da trajetória de referência e da variável manipulada. Um exemplo de função objetivo típica é dado pela equação abaixo:

$$J = \sum_{i=1}^{N_p} (\hat{y}(k+i) - sp(k+i))^2 \quad (4.19)$$

Minimizando J em relação a u , obtém-se a seqüência de ações de controle U num determinado horizonte de predição N_p . Deste modo, as ações u são ótimas em relação à função critério que é minimizada no instante k . Como resultado, o erro futuro é minimizado e portanto, se o modelo for idêntico ao processo e não ocorrerem perturbações, o processo seguirá fielmente a trajetória de referência nos instantes de amostragem.

Quando o processo em estudo tiver tempo morto associado, t_d , o desenvolvimento do sistema de controle não é diferente, a não ser pelo momento de início dos cálculos de otimização. Desde que o tempo morto seja um múltiplo do intervalo de amostragem, os cálculos deverão ser efetuados de $i = t_d/TA$ até N_p .

Embora todos os tipos de controladores preditivos com modelo estejam baseados no mesmo princípio descrito acima, existem diferenças significativas entre os propostos em literatura. Algumas das principais técnicas que emanam do MPC são o Controle por Matriz Dinâmica (DMC), Controle por Modelo Algorítmico (MAC), Controle por Rede Dinâmica (DNC). Estas três técnicas diferem entre si pelos modelos internos utilizados para se efetuarem as predições, que são respectivamente: Modelo de Convolução, Modelos Heurísticos, e Redes Neurais Artificiais. Além das diferenças entre os modelos, existem outras diferenças relacionadas ao método de otimização utilizado para o cálculo das ações de controle, às funções objetivo a serem otimizadas, às restrições do processo, e às trajetórias de referência a serem seguidas. Todas estas opções geram uma família bastante grande de controladores preditivos com modelo.

4.4.2 Histórico

Segundo Garcia *et al.* (1989), o interesse atual da indústria de processo no Controlador Preditivo com Modelo se deve ao conjunto de publicações que apareceram no final da década de 1970.

Em 1978, Richalet *et al.* descreveram aplicações industriais de sucesso do Controle Preditivo com Modelo Heurístico (MAC) e em 1979, engenheiros da Shell (Cutler e Ramaker, 1979; Prett e Gillette, 1979) lançaram o Controle por Matriz Dinâmica e mostraram algumas aplicações deste a uma unidade de craqueamento catalítico.

Em ambos algoritmos, um modelo dinâmico explícito da planta era utilizado para prever o efeito das futuras ações de controle na resposta do processo, sendo estas modificações determinadas por otimização, sujeitas a restrições de operação, com o objetivo de minimizar o erro predito. A otimização se repetia a cada intervalo de amostragem baseada em informações (medições) atualizadas da planta. Portanto, no contexto do MPC, o problema de controle (incluindo a importância relativa dos diferentes objetivos, restrições, etc.) é formulado como um problema de otimização dinâmica, constituindo um dos primeiros exemplos de aplicação rotineira, em larga escala, de otimização dinâmica em tempo real nas indústrias de processos.

O conceito do MPC tem uma longa história. Ainda segundo Garcia *et al.* (1989), Zadeh e Whalen (1962) foram os primeiros a reconhecer as conexões entre os intimamente relacionados problemas de Controle Ótimo de Tempo Mínimo e os de Programação linear. Propoi (1963) propôs o modelo do Horizonte Móvel, que hoje é o coração de todos os algoritmos de MPC, ficando conhecido por Retroalimentação Ótima em Malha Aberta. O trabalho extensivo, durante a década de 1970, sobre este problema foi revisado na tese de Gutman (1982). A conexão deste trabalho e o MPC foi descoberta por Chang e Seborg (1983).

Desde o redescobrimento do MPC em 1978 e 1979, sua popularidade nas indústrias de processos químicos vêm crescendo rapidamente. Mehra *et al.* (1982) revisaram muitas destas aplicações, incluindo um superaquecedor, um gerador de vapor, um refeedor de uma coluna de destilação, etc. A Shell aplicou o MPC em vários

sistemas, além da unidade de craqueamento catalítico citada, dentre os quais um reator altamente não-linear em batelada (Garcia, 1984). Matsko (1985) resumiu várias destas aplicações de sucesso do MPC às indústrias de papel e celulose. Cutler e Hawkins (1987) descreveram um aplicação industrial bastante complexa a um reator de hidrocrackeamento, envolvendo sete variáveis independentes (cinco manipuladas e duas perturbâncias) e quatro dependentes (controladas), sujeito a algumas restrições de processo. Martin *et al.* (1986) citaram sete aplicações industriais em funcionamento e dez em projeto. Grosdidier (1987) implementou a tecnologia do MPC a colunas de destilação, de absorção e retificação e outras operações de refinamento de petróleo.

Academicamente, até 1989, o MPC vinha sendo aplicado, sob condições controladas, a simples tanques de mistura e trocadores de calor (Arkun *et al.*, 1986) assim como a sistemas de destilação para separação de misturas ternárias (Levien, 1985; Levien e Morari, 1987). Parrish e Brosilow (1985) compararam o MPC com esquemas de controle convencionais aplicados a um trocador de calor e a uma autoclave industrial.

Procurando comparar aspectos computacionais da implementação de diferentes estratégias não-lineares do MPC a um CSTR exotérmico, Sistu *et al.* (1993) afirmam que a eficiência computacional de qualquer estratégia MPC depende do método utilizado para predição da resposta do modelo no laço de otimização. Os respectivos tempos de computação requeridos de um controlador preditivo não-linear (NLPC), com métodos de solução seqüencial e simultâneo das equações diferenciais restritivas, foram comparados.

Preocupados com a robustez dos controladores MPC, Lee e Yu (1994) caracterizaram os efeitos de vários parâmetros ajustáveis no desempenho do sistema de controle baseados em análises do comportamento final no domínio da freqüência. Vários exemplos são apresentados para demonstrar como o desempenho e a robustez podem ser melhorados através de algumas regras práticas de sintonização.

A maioria das aplicações citadas acima são multivariáveis e envolvem restrições, sendo exatamente o tipo de problema que motivou o desenvolvimento das técnicas de controle MPC.

De um modo bastante independente, apareceu uma segunda ramificação do MPC cujo principal objetivo seria o controle adaptativo. Martín-Sánchez e Shah (1984)

apresentam uma revisão instrutiva sobre Sistemas de Controle Adaptativo Preditivo (APCS), onde maior ênfase é dada aos detalhes da aplicação prática do APCS em processos reais e um exemplo de implementação a uma coluna-piloto de destilação binária é ilustrado.

Nesta categoria encontram-se também o controlador preditivo proposto por Peterka (1984), o projeto por horizonte estendido (Ydstie, 1984) e o EPSAC (Controle Auto-Adaptativo com Predição Estendida) desenvolvido por DeKeyser e van Cauwenberghe (1985). DeKeyser *et al.* (1985) também se enquadram nesta categoria, assim como o algoritmo de Controle Preditivo Generalizado de Clarke *et al.* (1987a, b).

Visando melhorar o desempenho de sistemas de controle adaptativos inferenciais, Shen e Lee (1989) propuseram o uso de um algoritmo cauteloso de estimação de parâmetros em linha e um modelo preditivo adaptativo para predição dos efeitos futuros das freqüentes perturbações não mensuráveis.

Em 1992, Morningred *et al.* apresentaram projeto de um novo controlador preditivo não-linear adaptativo baseado numa modelagem geral não-linear para a variável dependente (controlada). Este modelo, discreto e estacionário, chamado NARMAX, que permite fácil estimação de parâmetros através de métodos de otimização não-linear, foi então implementado com sucesso a um CSTR altamente não-linear.

Membro desta importante família de controladores (MPC), o histórico da tecnologia do MPC baseado em Redes Neurais Artificiais já foi anteriormente descrito no item que trata da revisão do uso de RNA's em controle de processos.

4.5 CONTROLE PREDITIVO COM MODELO RNA

4.5.1 Implementação

No universo dos métodos do controle preditivo, a técnica do horizonte rescitivo tem sido considerada como uma lei "feedback" normal, sendo computacionalmente aplicável. Segundo Hunt *et al.* (1992), tem sido provado em literatura

que este método realmente possui propriedades de estabilidade desejadas para sistemas não-lineares (Keerthi e Gilbert, 1986; Mayne e Michalska, 1990).

Na estrutura de controle proposta para implementação, uma Rede Neural Artificial, treinada em tempo real, fornece a predição da resposta futura da planta por um determinado horizonte de tempo. As predições supridas pela RNA são passadas para uma rotina de otimização numérica, que visa minimizar uma função objetivo, resultando em ações de controle ótimas.

Chamado por Willis *et al.* (1992) de Controle por Rede Dinâmica (DNC), o diagrama de blocos deste controlador preditivo está representado na figura 4.10.

Para o desenvolvimento do modelo do processo usando aproximação por RNA, a topologia da rede foi primeiramente especificada. Como dito anteriormente no item que trata do treinamento da RNA, a arquitetura ideal para o objetivo da implementação do controlador DNC é a representada pela figura 4.6, ou seja, rede com conexão intercamadas e com alimentação direta ("Feedforward Artificial Neural Network"). Já a definição do número de neurônios das camadas de entrada e saída dependerá das variáveis a serem controladas, manipuladas e das que possam interferir diretamente no comportamento do processo.

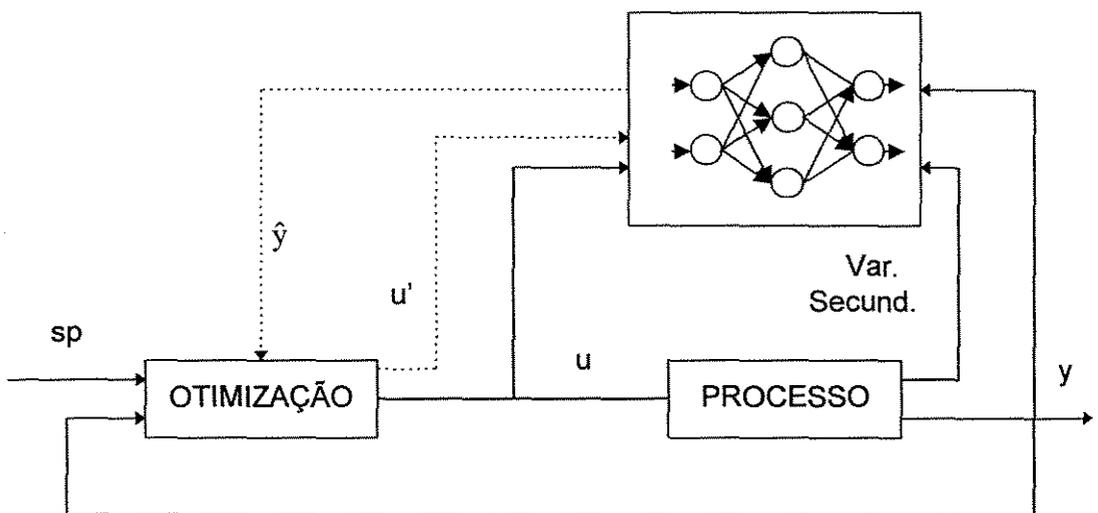


Fig. 4.10 - Diagrama de blocos do Controlador Preditivo com Modelo RNA.

Como dito anteriormente, no equipamento de destilação em batelada (figura 2.1) para que a composição de topo se mantenha fixa é necessário que se opere variando a razão de refluxo para compensar a constante mudança operacional da mesma. Durante o estudo, a quantidade de calor suprida ao refeedor foi manipulada com o objetivo de manter a taxa de evaporação fixa no fundo da coluna. Sendo a composição de fundo o tempo todo afetada pela descontinuidade de alimentação e pelas variações do fluxo de calor no refeedor, se torna importante indicador das condições operacionais da coluna, caracterizando o processo como múltipla-entrada-única-saída (MISO). Determinam-se então três variáveis básicas fundamentais para predição da composição de topo no horizonte de predição: composição (ou temperatura) de topo e fundo atuais e razão de refluxo determinada no instante anterior.

Quando se deseja a predição de comportamento em mais de um instante futuro de amostragem é necessário que se utilizem os próprios valores preditos pela rede. Portanto, sabendo-se que a razão de refluxo será calculada paralelamente através de um método de otimização no horizonte de controle, a saída da rede será composta pelas composições (ou temperaturas) de topo e fundo para poder realimentar a mesma, tornando-a recursiva.

Deste modo, define-se a arquitetura básica da rede com 3 neurônios de entrada, 2 de saída e Mn na camada oculta (figura 4.11).

Devido à característica transiente do processo em estudo, uma rede estática levando em consideração somente os dados do instante atual de amostragem não é suficiente para predição exata deste comportamento. Assim, para transformá-la em uma rede dinâmica deve-se fornecer um histórico das variáveis de entrada, incorporando duas, três ou mais amostragens anteriores consecutivas destas variáveis na primeira camada. Deste modo altera-se a arquitetura básica proposta na figura 4.11 para (6-Mn-2), (9-Mn-2), etc. e obtém-se uma rede dinâmica com janela móvel de entrada, acomodando também os atrasos de transporte inerentes ao processo.

Na simulação de processos normalmente se utiliza rede neural com somente uma camada oculta, sugerida ser suficiente para captação de não-linearidades. Segundo Su *et al.* (1992) apesar de vários pesquisadores considerarem este tópico, não se tem disponível uma metodologia geral e eficiente para a determinação do número de neurônios

presentes nesta camada, principalmente quando a rede é dinâmica. Os autores afirmam ainda que um número de neurônios excessivo poderia ocasionar um sobreajuste (“overfitting”) dos dados de treinamento, além de tornar mais lenta a etapa de aprendizagem. Portanto devem ser efetuados testes para avaliação dos desvios de predição e firmar um compromisso entre estes e os tempos requeridos para treinamento.

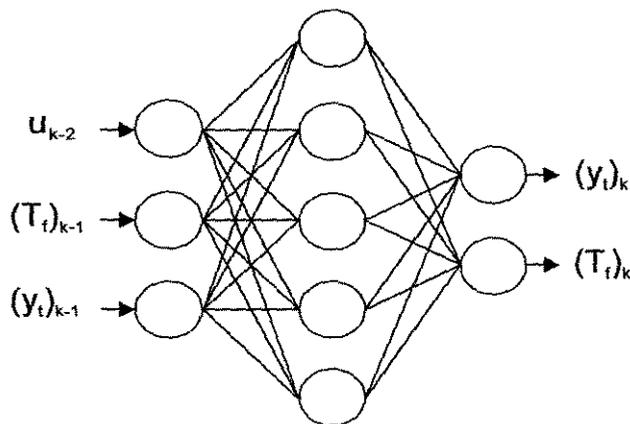


Fig. 4.11 - RNA básica, de topologia (3-Mn-2), representativa da coluna em batelada.

Conforme figura acima, nota-se que o vetor-entrada, I , para treinamento supervisionado, é composto de três elementos básicos: u_{k-2} , $(T_{\text{fundo}})_{k-1}$ e y_{k-1} , ou respectivamente, razão de refluxo, temperatura de fundo e composição de topo dos instantes anteriores de amostragem. Do mesmo modo, o vetor-saída-desejada, d , é constituído por dois elementos: y_k e $(T_{\text{fundo}})_k$, ou composição de topo e temperatura de fundo da amostragem atual.

A cada amostragem efetuada pelo sistema de controle, a RNA acima é treinada através do algoritmo de treinamento GDR, detalhado anteriormente. Depois de ajustados, em linha, os fatores ponderais da rede, a mesma estará apta a ser utilizada no sistema de controle para predição de valores futuros de temperatura de fundo, \hat{T}_{fundo} , e composição de topo, \hat{y} , a partir de valores amostrados e preditos de temperatura e composição, além de valores implementados da variável manipulada, u , e valor “pseudo-otimizado” da mesma, u' (ver figura 4.10).

Para evitar problemas nos cálculos efetuados na etapa de treinamento e permitir previsões fora do intervalo $[0,1]$, restrito pela função sigmoideal, todos os dados retirados do processo e impostos à rede neural devem ser normalizados, assumindo valores entre 0 e 1.

Sendo o controlador baseado em previsões das futuras saídas do processo, obtidas a partir de um modelo, podem surgir “off-set’s” decorrentes de distúrbâncias não previstas pela modelagem. Utilizando-se o mesmo conceito de discrepância do IMC, entre o modelo e o processo ($\text{dic}(k)$), devem-se corrigir as previsões do instante atual a partir da avaliação dos desvios obtidos das previsões no instante imediatamente anterior, ou seja:

$$\text{dic}(k) = \tilde{y}(k) - \hat{y}(k-1) \quad (4.20)$$

onde $\tilde{y}(k)$ é o valor atual amostrado da variável de saída da rede e $\hat{y}(k-1)$ o valor predito para a variável de saída do instante atual k , no instante anterior $k-1$.

Em circunstâncias onde houver ruído significativo nas medições, a estimativa deste “off-set” pode ser filtrada para assegurar estabilidade. O valor resultante da discrepância filtrada é então usado para corrigir as previsões atuais obtidas do modelo:

$$\hat{y}_{\text{corrigido}}(k) = \hat{y}(k) + \text{dic}(k) \quad (4.21)$$

sendo $\hat{y}(k)$ os valores de saída preditos no instante atual.

Trabalhando os blocos de otimização e previsão em conjunto, as previsões efetuadas em linha pela RNA são fornecidas à rotina de otimização que procura minimizar um critério de desempenho, normalmente quadrático, baseada em cálculo do erro entre as previsões e a trajetória desejada da variável controlada. A função objetivo utilizada neste trabalho é dada pela equação (4.22), sugerida tanto por Willis *et al.* (1992) quanto por Hunt *et al.* (1992).

$$J = \sum_{i=1}^{N_p} (\hat{y}(k+i) - \text{sp}(k+i))^2 + \sum_{i=0}^{N_f} q_i (u'(k) - u'(k-1))^2 \quad (4.22)$$

onde N_p é o horizonte de predição, N_c o horizonte de controle e q_i são pesos utilizados para penalizar variações excessivas da variável manipulada.

Para efeito de predição, quando o horizonte de predição ultrapassar o horizonte de controle, deve-se assumir que a variável manipulada permaneça constante. Apesar da rotina de otimização fornecer uma seqüência de ações ótimas de controle, somente a primeira ação é implementada ao processo, assim como na maioria das estratégias de controle preditivo.

De uma forma mais direta, o procedimento para implementação do Controlador Preditivo Não-linear com Modelo RNA é apresentado no algoritmo da figura 4.12.

Na busca da solução ótima para as ações de controle, uma solução analítica da função objetivo seria bastante difícil, senão impossível, devido às não-linearidades da RNA. Portanto para se obter a solução desejada, apesar de mais demorada computacionalmente, adotou-se uma aproximação por otimização numérica.

A maioria dos algoritmos de otimização numérica empregam alguma forma de técnica de busca para efetuar a varredura do espaço possível da função objetivo, até que um ponto extremo seja encontrado (ver Edgar e Himmelblau, 1989). A pesquisa é geralmente guiada por cálculos com a própria função objetivo e/ou as derivadas desta função. De um modo geral, os vários procedimentos disponíveis podem ser classificados como “baseados no gradiente” (métodos indiretos) e “sem gradiente” (métodos diretos).

Segundo Willis *et al.* (1992), devido à complexidade do controlador proposto, as técnicas “sem gradiente” são mais apropriadas, além de conferirem maior flexibilidade quanto à forma do modelo - linear ou não-linear - a ser incorporado no controlador.

Dentre os métodos que não se utilizam das derivadas parciais da função objetivo na determinação da direção de busca, Fletcher (1980) mostrou que o mais eficiente era o desenvolvido por Powell (1964).

Assim, por sua simplicidade de cálculos e conseqüente maior agilidade, apesar de não poder incorporar as restrições do processo, o Método de Powell foi escolhido como base para o controlador preditivo não-linear implementado neste trabalho.

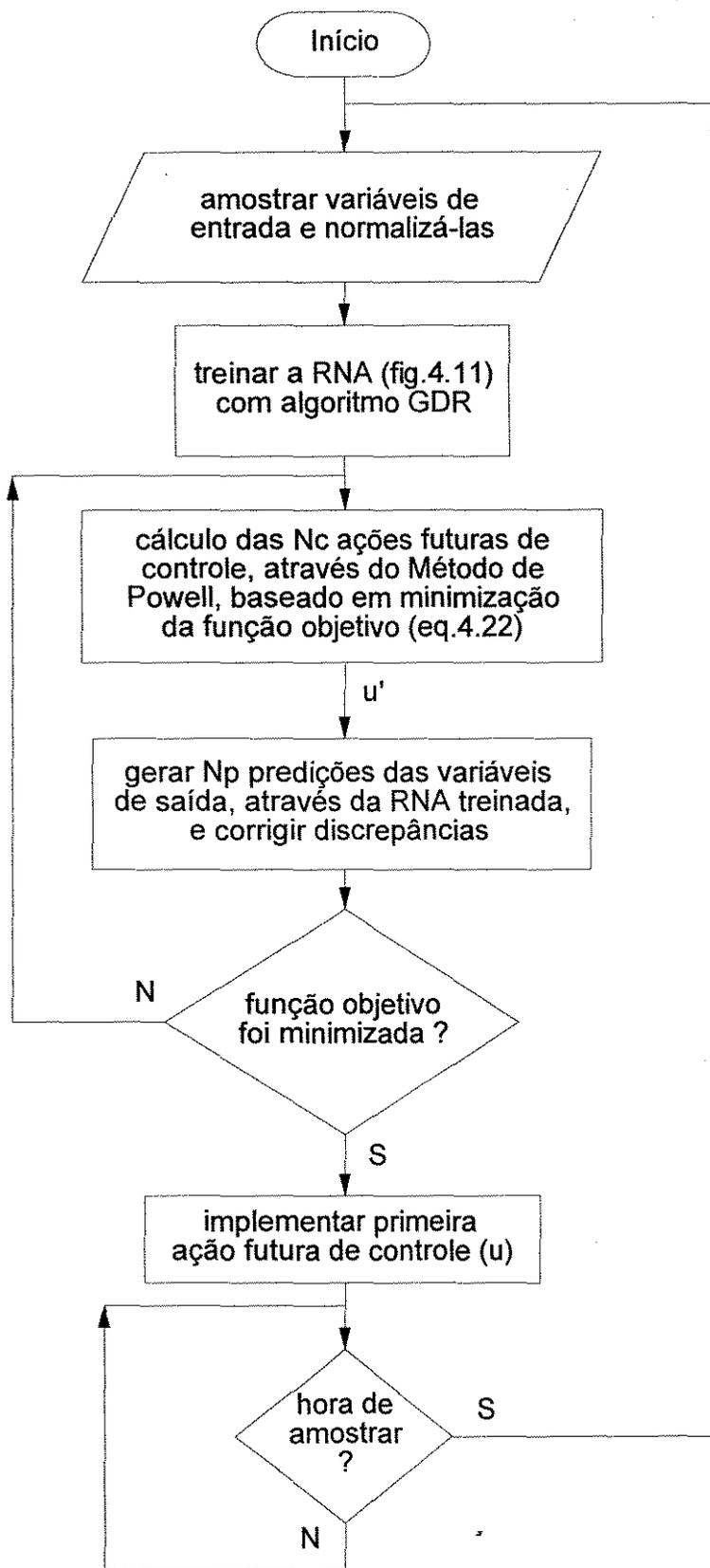


Fig. 4.12 - Algoritmo de implementação do Controlador Preditivo com Modelo RNA.

4.5.1.1 Método de Powell

O Método de Powell localiza o mínimo de uma função f , através de pesquisas unidimensionais seqüenciais, a partir de um ponto inicial (x^0) ao longo de um conjunto de direções conjugadas geradas pelo algoritmo. Com o andamento da busca, novas direções são introduzidas.

Com o objetivo de clarear a aplicação do Método de Powell, uma função de somente duas variáveis é considerada. A busca começa num ponto escolhido arbitrariamente (x_1^0, x_2^0). Duas direções de busca linearmente independentes (s_1^0, s_2^0) são escolhidas; por exemplo, os eixos coordenados ($[1 \ 0], [0 \ 1]$) formam uma escolha inicial adequada. Uma busca unidimensional é então executada para encontrar o ponto ótimo na direção coordenada x_1 :

$$x_1^0 = x_0^0 + \lambda_1^0 s_1^0 \quad (4.23)$$

onde λ_1^0 é o passo ótimo determinado por minimização unidimensional.

Outra busca unidimensional é executada do ponto x_1^0 na direção coordenada x_2 para encontrar de x que minimiza $f(x)$:

$$x_2^0 = x_1^0 + \lambda_2^0 s_2^0 \quad (4.24)$$

Por último, mais um passo, a partir de x_2^0 , de tamanho correspondente ao passo total ($x_2^0 - x_0^0$) iniciado do estágio 0, é tomado, conforme figura 4.13.

O algoritmo relativo ao Método de Powell é mostrado no diagrama de blocos da figura 4.14 e sua implementação é descrita, também em linguagem C, no conjunto de sub-rotinas situado no ANEXO 7 deste trabalho.

Em geral, o k -ésimo estágio do Método de Powell usa n direções de busca linearmente independentes. Considerando-se o início em $x_0^k = x_{n+1}^{k-1}$, o método resume-se nas seguintes etapas descritas abaixo:

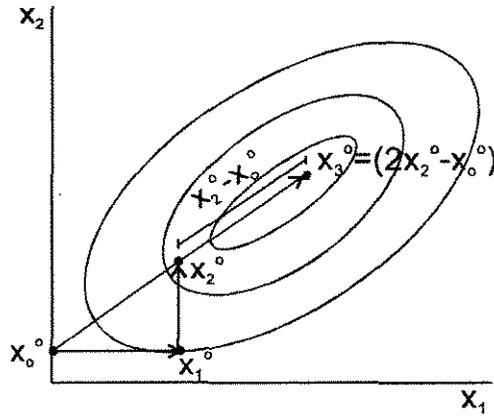


Fig. 4.13 - Busca realizada pelo Método de Otimização de Powell.

1ª etapa: A partir de x_0^k , determinar λ_1^k , através de busca unidimensional na direção s_1^k , onde $f(x_0^k + \lambda_1 s_1^k)$ seja mínimo. Fazer $x_1^k = x_0^k + \lambda_1^k s_1^k$. A partir de x_1^k determinar λ_2^k tal que $f(x_1^k + \lambda_2 s_2^k)$ seja um mínimo. Fazer $x_2^k = x_1^k + \lambda_2^k s_2^k$. A busca deve continuar seqüencialmente em cada direção, começando sempre do último ponto da seqüência até que todos $\lambda_i^k, i=1, \dots, n$, sejam determinados.

2ª etapa: Segundo Powell, a busca descrita na primeira etapa pode levar a direções de busca linearmente dependentes, quando, por exemplo, um dos componentes de s_k for igualado a zero já que nenhum progresso tenha sido feito nesta direção particular. Duas direções de busca podem então se tornar essencialmente colineares. Assim, pode ser insensato substituir a direção antiga pela nova se o conjunto de direções puder se tornar linearmente dependente. Concluindo, Powell afirma que as novas direções de progresso para o k-ésimo estágio, s_k , devem substituir as direções de busca existentes somente se o vetor de substituição aumentar o determinante da matriz das direções de busca normalizadas. Deste modo, depois de minimizar $f(x)$ em cada uma das n direções (1ª etapa), um passo adicional de tamanho $(x_n^k - x_0^k)$ é tomado correspondente ao progresso total do k-ésimo estágio, resultando o ponto $(2x_n^k - x_0^k)$.

3ª etapa: Para verificar se, adicionando-se a nova direção, o determinante das direções de busca cai ou não, um teste deve ser realizado.

Seja a maior redução em $f(x)$, em qualquer direção no k -ésimo estágio, representada por:

$$\Delta^k = \max\{f(x_{i-1}^k) - f(x_i^k)\} \quad i=1, \dots, n \quad (4.25)$$

e seja a direção de busca, correspondente a esta variação máxima em $f(x)$, representada por s_m^k . Para que a notação se torne mais compacta, assume-se:

$$f_1 = f(x_0^k), \quad f_2 = f(x_n^k) \quad \text{e} \quad f_3 = f(2x_n^k - x_0^k), \quad \text{onde} \quad x_0^k = x_n^{k-1} \quad \text{e}$$

$$x_n^k = x_{n-1}^k + \lambda_n^k s_n^k = x_0^k + \sum_{i=1}^n \lambda_i^k s_i^k.$$

Se $f_3 \geq f_1$ e/ou $(f_1 - 2f_2 + f_3)(f_1 - f_2 - \Delta^k)^2 \geq 0.5\Delta^k(f_1 - f_3)^2$ usar as mesmas direções s_1^k, \dots, s_n^k no $(k+1)$ -ésimo estágio, ou seja, $s_i^{k+1} = s_i^k$ para $i=1, \dots, n$ e começar de $x_0^{k+1} = x_n^k$ ou de $x_0^{k+1} = 2x_n^k - x_0^k = x_{n+1}^k$ dependendo de qual destes resultar no menor valor de $f(x)$.

4ª etapa: Se o teste da 3ª etapa não for satisfeito, a direção s^k de x_0^k para x_n^k é pesquisada para o mínimo de $f(x)$, que será utilizado como ponto inicial para o próximo estágio $(k+1)$. O conjunto de direções a ser usado no estágio $k+1$ deve ser o mesmo do k -ésimo estágio, com exceção da direção s_m^k , que é substituída por s^k . Entretanto, s^k é colocada na última coluna da matriz de direções, ao invés do lugar de s_m^k . Consequentemente, as direções a serem usadas no estágio $k+1$ se tornam:

$$[s_1^{k+1} \quad s_2^{k+1} \quad \dots \quad s_n^{k+1}] = [s_1^k \quad s_2^k \quad \dots \quad s_{m-1}^k \quad s_{m+1}^k \quad \dots \quad s_n^k \quad s^k] \quad (4.26)$$

5ª etapa: Terminar a busca no fim de cada estágio no momento em que a variação de cada variável independente for menor que determinada precisão ϵ_i para $i=1, \dots, n$ ou

$$\|x_n^k - x_0^k\| \leq \epsilon.$$

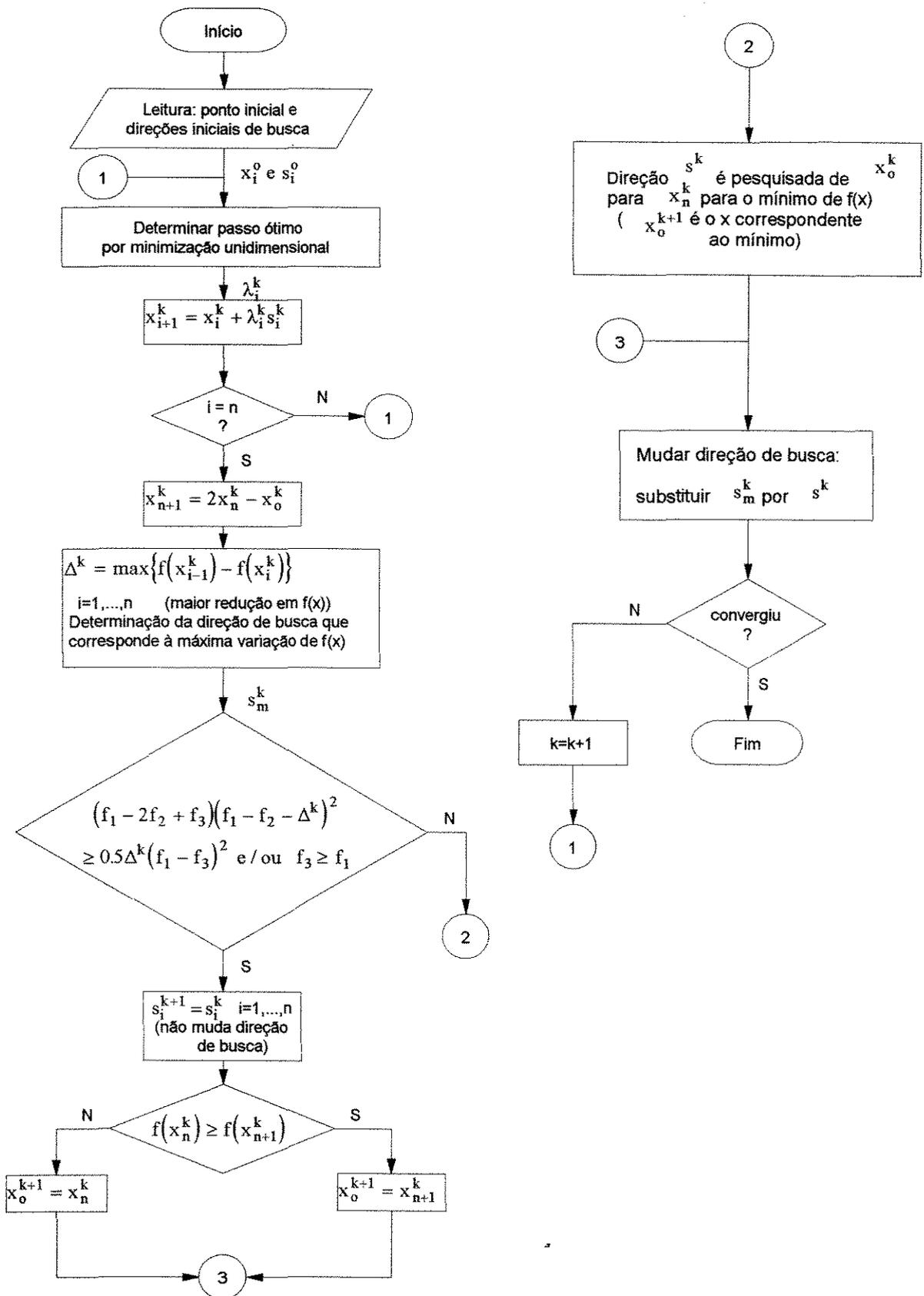


Fig. 4.14 - Algoritmo do Método de Powell, usado para otimização das ações do controlador.

4.6 RESULTADOS E DISCUSSÃO

Para efeito de comparação de resultados entre os controladores adaptativos descritos no capítulo anterior e o controlador preditivo deste capítulo e também visando a exemplificação da escolha dos melhores parâmetros deste último, utilizaram-se alguns casos resumidos na tabela 3.2 do capítulo 3.

O primeiro exemplo escolhido foi o caso 3 da tabela 3.2, onde 100 moles de uma mistura inicial de fração molar 0,30 de n-hexano, 0,45 de benzeno e 0,25 de tolueno são destilados em uma coluna de 6 estágios para a obtenção de duas frações de purezas diferentes: a primeira com fração molar de 0,65 de n-hexano e a segunda de 0,70 de benzeno. A geometria da coluna resume-se na tabela referida acima.

O segundo exemplo a ser tratado neste capítulo é o caso 2 da mesma tabela, onde deseja-se a obtenção somente do segundo componente mais volátil da mistura com fração molar de 0,98, a partir de mistura inicial de 0,15 de n-hexano, 0,75 de benzeno e 0,10 de tolueno. Parte-se de 100 moles da mistura inicial e opera-se em uma coluna de 9 estágios.

4.6.1 Exemplo 1

Para a retirada de mais de um componente com pureza especificada no topo da coluna, o sistema de controle foi igualmente acrescido de um conjunto de condições que apontasse o fim de cada ciclo da batelada. Dotando o sistema de capacidade de decisão baseada no conhecimento do equipamento operado, a condição principal inserida é a fração molar atualmente controlada estar abaixo de seu valor de referência mesmo quando a razão de refluxo já estiver em seu valor máximo. De mesmo modo que no sistema de controle adaptativo, acrescentou-se um filtro para “amaciamento” da curva de dados amostrados para que houvesse a confirmação do decréscimo da curva da variável controlada. Também aqui o sistema de controle deverá se adaptar se necessário: podem ser utilizadas diferentes arquiteturas de redes neurais para previsão de comportamento e diferentes horizontes de predição e controle, assim como diferentes valores dos fatores restritivos de variação excessiva (q_i - eq. 4.22).

Conforme item 4.5.1, descritivo da implementação do controlador preditivo baseado em RNA no equipamento em estudo, a arquitetura básica da RNA (fig.4.11), para o propósito, seria de 3 neurônios de entrada (fração molar do componente controlado no topo e temperatura de fundo atuais e razão de refluxo do instante anterior) e 2 de saída (predições de fração molar do componente controlado no topo e temperatura de fundo).

Conferindo características dinâmicas à rede e acomodando os atrasos de transportes inerentes ao processo, altera-se a mesma para que incorpore uma janela móvel de dados de entrada. Para tanto são testadas separadamente as seguintes configurações: (3-Mn-2), (6-Mn-2), (9-Mn-2) e (12-Mn-2), onde Mn é o n^o de neurônios da camada oculta.

Também discutido no item 4.5.1, o número de neurônios a ser utilizado na camada oculta deve ser obtido de testes para avaliação dos desvios de predição e tempo requerido para treinamento da mesma. Atribuíram-se os valores de 0,9 para a taxa de aprendizado, η , e de 0,7 para o coeficiente de "momentum", α , utilizados pelo algoritmo de treinamento da rede (equações 4.15 e 4.16).

Realizando-se diversas simulações da coluna em batelada em malha aberta e amostrando-se, a cada 6 segundos, os dados necessários para treinamento da rede em paralelo ao simulador, resumem-se os resultados obtidos nas figuras 4.15, 4.16 e 4.17. As figuras 4.15 e 4.16 mostram os desvios quadráticos médios obtidos de predições em linha da fração molar de topo do componente mais volátil em relação ao valor obtido pelo simulador rigoroso, respectivamente um instante e dois instantes de amostragem futuros. Nestas figuras pode-se verificar que a arquitetura (6-18-2) foi a que apresentou menor desvio médio de predição, apesar dos bons resultados obtidos em todas variantes: desvios entre a 4^a e 5^a casas decimais de valores de frações molares. Quanto ao compromisso em relação ao tempo requerido para treinamento da rede, não houve restrições já que todos os treinamentos efetuados com as diversas topologias não excederam 1,5 segundos num PC-486 DX2 de 66MHz.

Escolhida a utilização da rede de arquitetura (6-18-2), a figura 4.17 mostra a variação do desvio quadrático da predição de uma amostragem futura durante a operação da coluna em malha aberta, juntamente com a variação imposta à razão de refluxo. Pode-se verificar que o maior desvio ocorre quando há uma inversão no sentido da futura variável manipulada da malha de controle de topo.

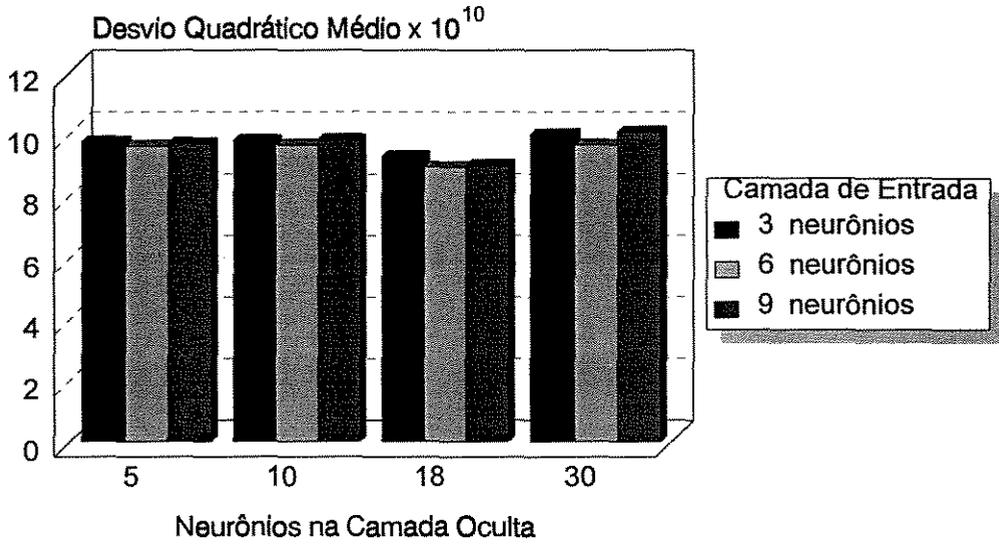


Fig. 4.15 - Desvios Quadráticos Médios de predição (um instante de amostragem futuro) da fração molar de topo, do componente mais volátil, em relação ao simulador rigoroso.

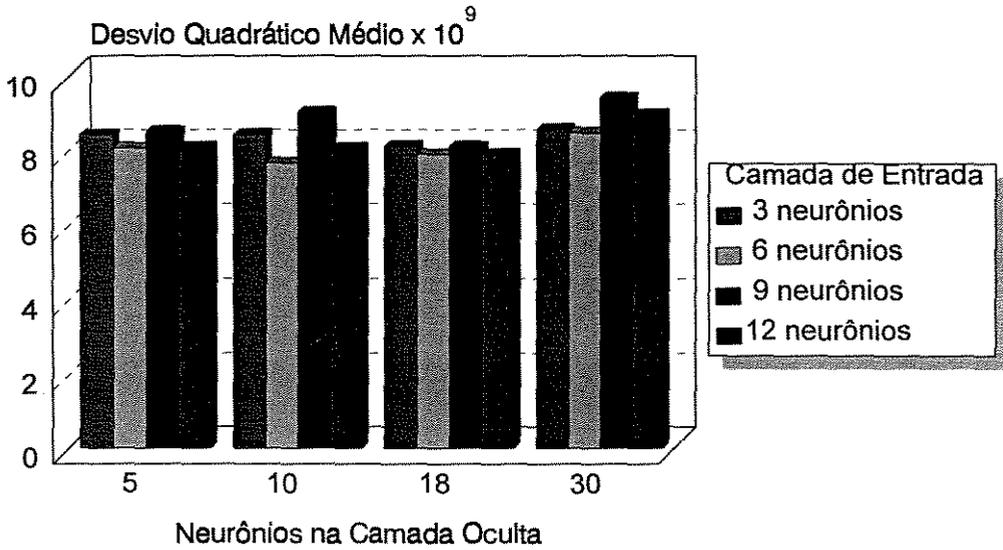


Fig. 4.16 - Desvios Quadráticos Médios de predição (dois instantes futuros de amostragem) da fração molar de topo, do componente mais volátil, em relação ao simulador rigoroso.

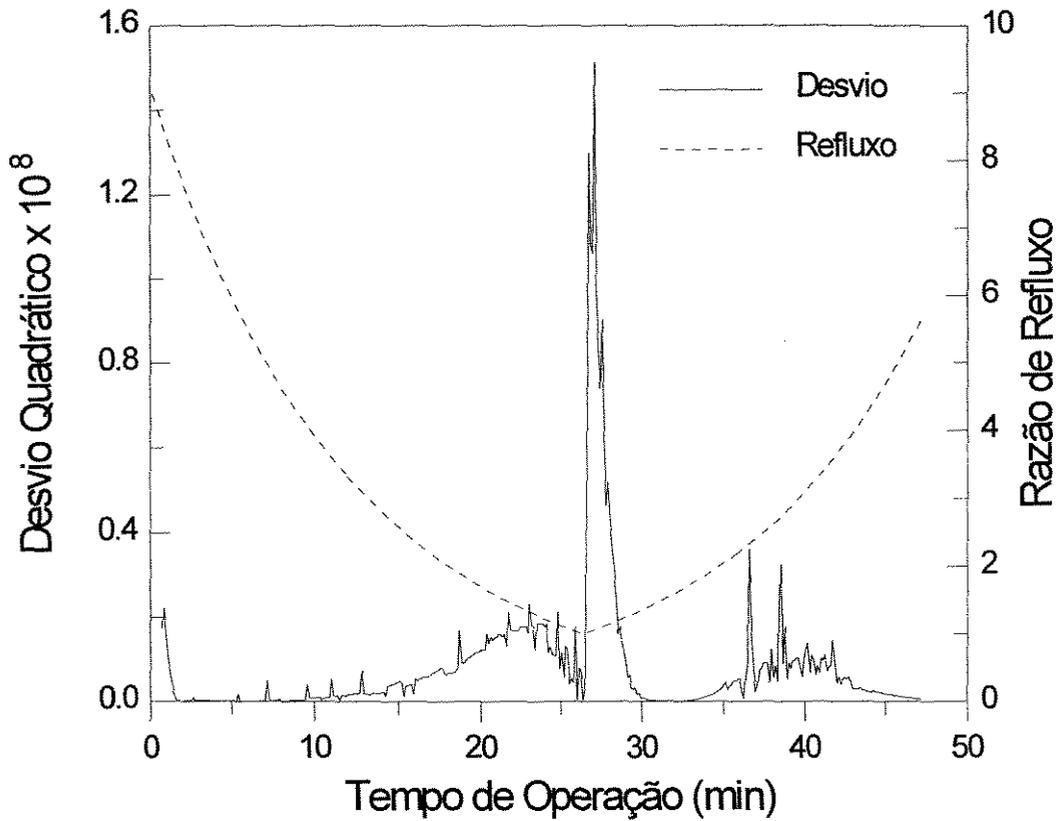


Fig. 4.17 - Variação do desvio quadrático de predição via RNA (6-18-2), um instante futuro de amostragem, da fração molar de topo do componente mais volátil.

Com este procedimento pode-se determinar a estrutura mais recomendada para o sistema em questão, porém deve-se ter em mente que os desvios poderão assumir valores bastante diferentes quando o equipamento estiver sob controle, onde as variações poderão mudar o sentido freqüentemente e quando o intervalo de amostragem requerido pode aumentar.

Determinada a melhor arquitetura da rede neural artificial que proverá ao sistema de controle a previsão de comportamento do equipamento em estudo, a cada tentativa de otimização da razão de refluxo dentro do intervalo de amostragem, segue-se para a implementação do controle preditivo propriamente dito.

Fazendo uso da função objetivo descrita pela equação 4.22 e do método de otimização detalhado no item 4.5.1.1, implementa-se o esquema do controlador preditivo representado no diagrama de blocos da figura 4.10 e no algoritmo ilustrado pela figura 4.12.

As figuras 4.18a e 4.18b ilustram os comportamentos obtidos em malha fechada para as variáveis controlada e manipulada de topo do exemplo 1, respectivamente.

É claro que a escolha dos horizontes de predição (N_p) e controle (N_c) e dos fatores restritivos (q_i) afeta diretamente na qualidade da resposta do equipamento sob controle, conforme se observa no equacionamento da função objetivo. Portanto, os resultados mostrados nas figuras 4.18 são os melhores encontrados de uma série de simulações em malha fechada da coluna destilando o sistema ternário deste exemplo, deixando esta discussão para o próximo exemplo.

Utilizando-se a capacidade de decisão inserida no programa de controle para troca de variável controlada, observa-se que a mesma é efetuada quando aproximadamente 46% da carga inicial fora obtida como produto de topo. Comparando-se este resultado com os obtidos dos controladores adaptativos (fig. 3.7a), verifica-se um melhor desempenho do controlador preditivo em relação à retirada do componente mais volátil. Além de resultar "off-set" menor, no intervalo de 36 minutos - em que o STR/Dahlin recuperou apenas 52% do n-hexano da carga inicial - recuperaram-se 73% do mesmo. A recuperação total obtida pelo controlador preditivo chegou a 78% em 48 minutos.

Porém, em relação ao controle do segundo componente mais volátil (benzeno), um índice menor de recuperação foi encontrado: 58% quando se mantinham os horizontes de predição e controle e 50% modificando-se os mesmos, contra os 66% obtidos com o controlador auto-ajustável STR/Dahlin. Os desvios encontrados para a fração molar do benzeno também foram maiores, mas podendo estes fatos ser explicados pela grande quantidade obtida como produto de topo do primeiro componente definido como prioritário.

Como o esgotamento do n-hexano da coluna foi quase total no primeiro ciclo, pouca impureza sobrou para que o benzeno se mantivesse com uma pequena pureza de 70% no topo. Isto pode ser melhor observado na figura 4.18b, onde as duas curvas de razão de refluxo tomam direções opostas (aproximadamente aos 68% de destilado acumulado) e na figura 4.18a a fração molar de benzeno aumenta para qualquer uma das duas estratégias seguidas.

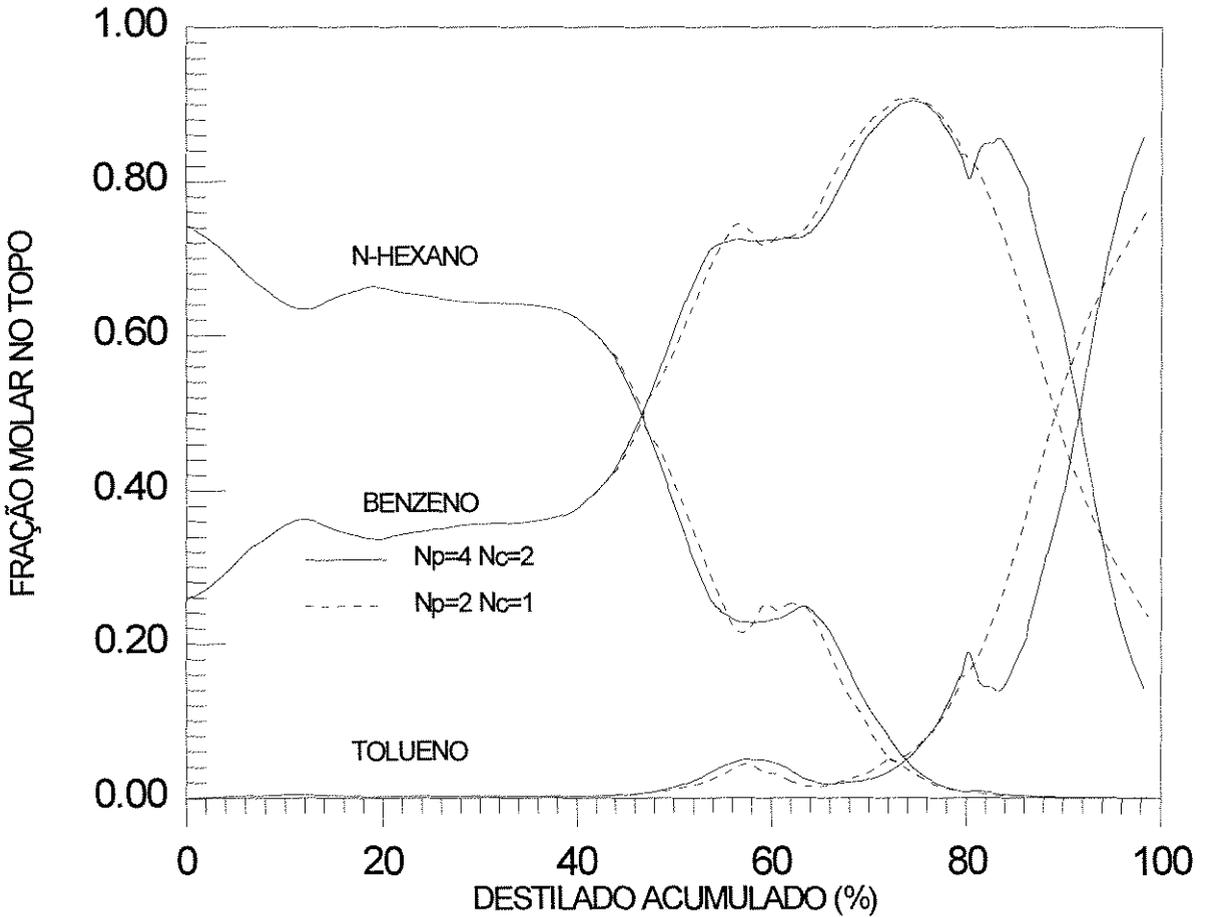


Fig. 4.18a - Composição de topo do exemplo1 sob Controle Preditivo baseado em RNA (6-18-2), com horizonte de predição (N_p) = 4 e de controle (N_c) = 2 para a retirada do n-hexano e N_p e N_c variando conforme legenda para a retirada do benzeno.

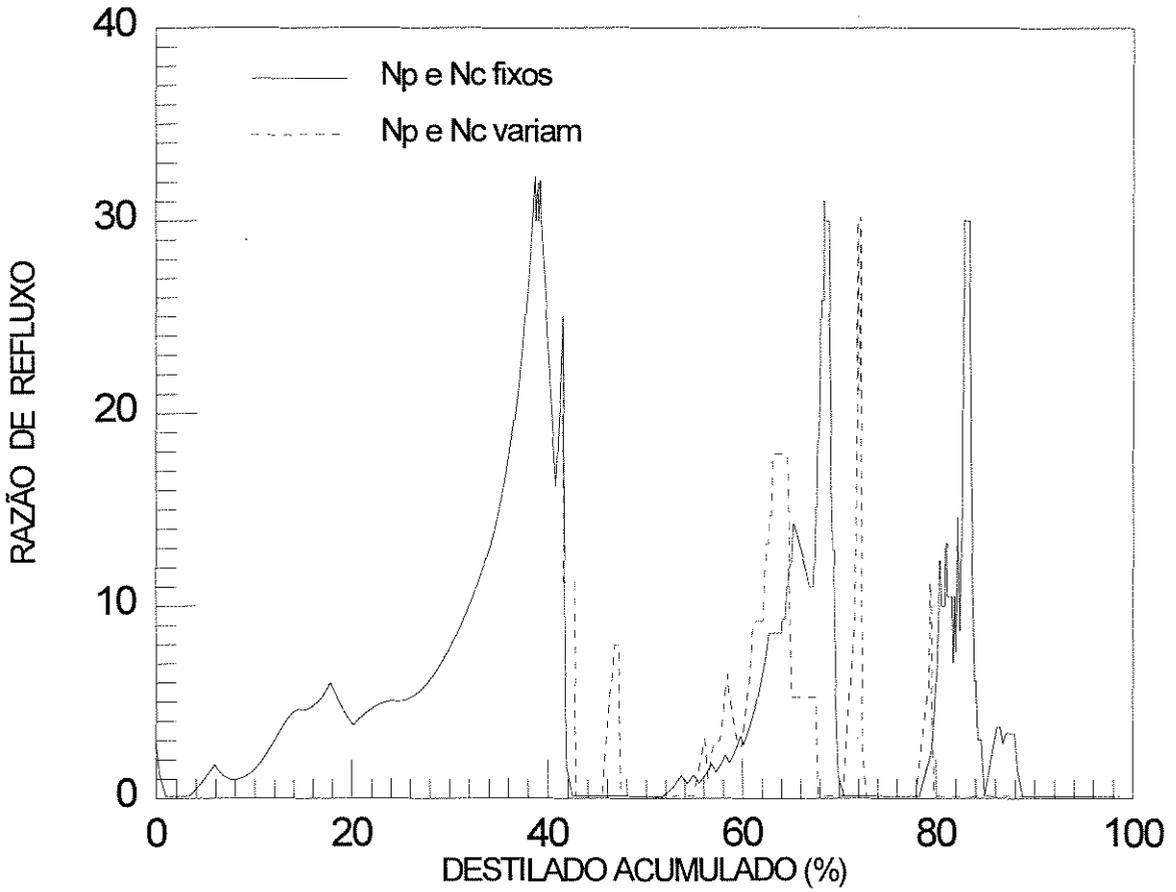


Fig. 4.18b - Comportamento da variável manipulada da malha de topo para o exemplo 1.

4.6.2 Exemplo 2

Também estudado no capítulo que trata de controladores adaptativos, o exemplo 2 se resume no caso 2 da tabela 3.2 daquele capítulo. A prioridade neste caso é a obtenção do segundo componente mais volátil com uma pureza de 0,98 em fração molar a partir de uma mistura de 0,15 de n-hexano, 0,75 de benzeno e 0,10 de tolueno.

Similar ao exemplo 1, realizaram-se diversas simulações da coluna em batelada em malha aberta e amostrando-se os dados necessários para treinamento da rede em paralelo ao simulador, analisaram-se os desvios quadráticos médios obtidos de predições em linha da fração molar de topo do componente de interesse (benzeno), em relação aos valores obtidos pelo simulador rigoroso.

Para a etapa de treinamento da rede foram utilizados $\eta=3$, $\alpha=0,7$ e desvios permitidos de 10^{-5} para os valores obtidos dos dois neurônios de saída da rede no treinamento supervisionado.

Mais uma vez bons resultados foram obtidos das arquiteturas utilizadas: desvios entre a 4^a e 5^a casas decimais de valores de frações molares para os dois primeiros intervalos de amostragem futuros e entre a 3^a e 4^a casas entre $N_p=3$ e $N_p=6$. Verificou-se então que a arquitetura (6-15-2) foi a que apresentou menores desvios médios de predição, apesar dos bons resultados globais obtidos.

A média do tempo requerido para treinamento da rede com esta estrutura foi de 0,5s no mesmo PC-486 DX2 de 66MHz. Não sendo objeto de estudo neste trabalho, constatou-se que a utilização de diferentes valores de taxa de aprendizado, η , interfere diretamente nos tempos de treinamento requeridos pelo algoritmo de Retropropagação utilizado (GDR). Com valor de $\eta=3$ o treinamento efetua-se mais rapidamente, mas, dependendo da tolerância requerida para os valores de saída, a convergência pode não ser atingida.

Conforme mencionado no exemplo anterior, a escolha dos horizontes de predição (N_p) e controle (N_c) e dos fatores restritivos (q_i) afeta diretamente a qualidade da resposta em malha fechada do equipamento sob controle preditivo. Portanto, diversas simulações em malha fechada foram realizadas com o intuito de se estabelecerem alguns critérios que pudessem auxiliar na escolha destes parâmetros.

Deve-se lembrar aqui que o melhor resultado obtido para este caso com os controladores adaptativos foi a recuperação de 60% de benzeno da carga inicial em 21 minutos, utilizando-se o STR/Dahlin com $\mu=10$.

Como primeira série de tentativas, manteve-se o horizonte de controle fixo na unidade ($N_c=1$) e para diferentes fatores restritivos, q , foram variados os horizontes de predição, N_p . Através da figura 4.19 verifica-se a obtenção de comportamentos adequados para a variável controlada, com boas taxas de recuperação para todos horizontes de predição utilizados. Dentre estas, a melhor é de 65% de benzeno da carga inicial recuperados em 21 minutos de operação. Com a observação das curvas tracejadas, nota-se que para este horizonte de controle fixo, o horizonte de predição maior ($N_p=6$) permitiu uma variação inicial mais brusca da razão de refluxo e o horizonte de predição menor não permitiu variação necessária, resultando em controle mais lento. O valor inicial estimado por tentativa-e-erro para a razão de refluxo é 3 e ao redor deste valor se mantém a variável manipulada quando se utiliza $N_p=2$, para o respectivo valor do fator restritivo que gerou os resultados da figura 4.19 ($q=-1,0$).

Mantendo-se $N_c=1$ e utilizando-se um valor absoluto de restrição menor que o da figura 4.19, a figura 4.20 ($q=-0,01$) revela melhor taxa de recuperação quando da utilização de $N_p=2$, indicando um efeito compensatório na grande variação permitida por este valor de $|q|$. Ou seja, quando se utiliza a combinação N_p grande e $|q|$ pequeno, são permitidos saltos muito elevados na variável controlada (curva $N_p=6$ - fig.4.20), resultando recuperação pobre e demorada já que a retirada do segundo componente com pureza especificada é bastante sensível a pequenas variações da razão de refluxo. Esta sensibilidade se deve à presença, na coluna, de componentes tanto mais voláteis como menos voláteis que o componente desejado.

A compensação de valores dos parâmetros ajustáveis, objetivando melhores resultados, pode ser melhor observada nas figuras 4.21 e 4.22. Mantendo-se os horizontes de predição (N_p) e de controle (N_c) fixos, o efeito da variação do fator restritivo é considerado nestas figuras. Através da figura 4.22 confirma-se um melhor resultado quando $|q|$ maior (curva $q=-1,0$) restringe o tamanho da variação permitida pelo valor de $N_p=6$. De maneira inversa, observa-se na figura 4.21 um resultado melhor com a utilização de $|q|$ pequeno ($q=-0,01$) para $N_p=2$. Na mesma figura, diminuindo-se ainda mais $|q|$,

grandes variações foram permitidas, assim como para a curva ($q=-0,01$) na figura 4.22, indicando que valores equilibrados devem ser procurados entre $|q|$ e N_p .

Observando-se a função objetivo utilizada (eq. 4.22), pode-se notar que com a utilização de horizontes de predição (N_p) maiores, o primeiro somatório desta função assume valores maiores devido à transitoriedade do processo e para que este seja anulado pelo segundo somatório - para a minimização da função objetivo - as mudanças da variável manipulada de topo teriam que ser bastante grandes (talvez até desestabilizadoras) se não fosse pela ajuda do fator restritivo q (daí também se explica a utilização de valores negativos para q).

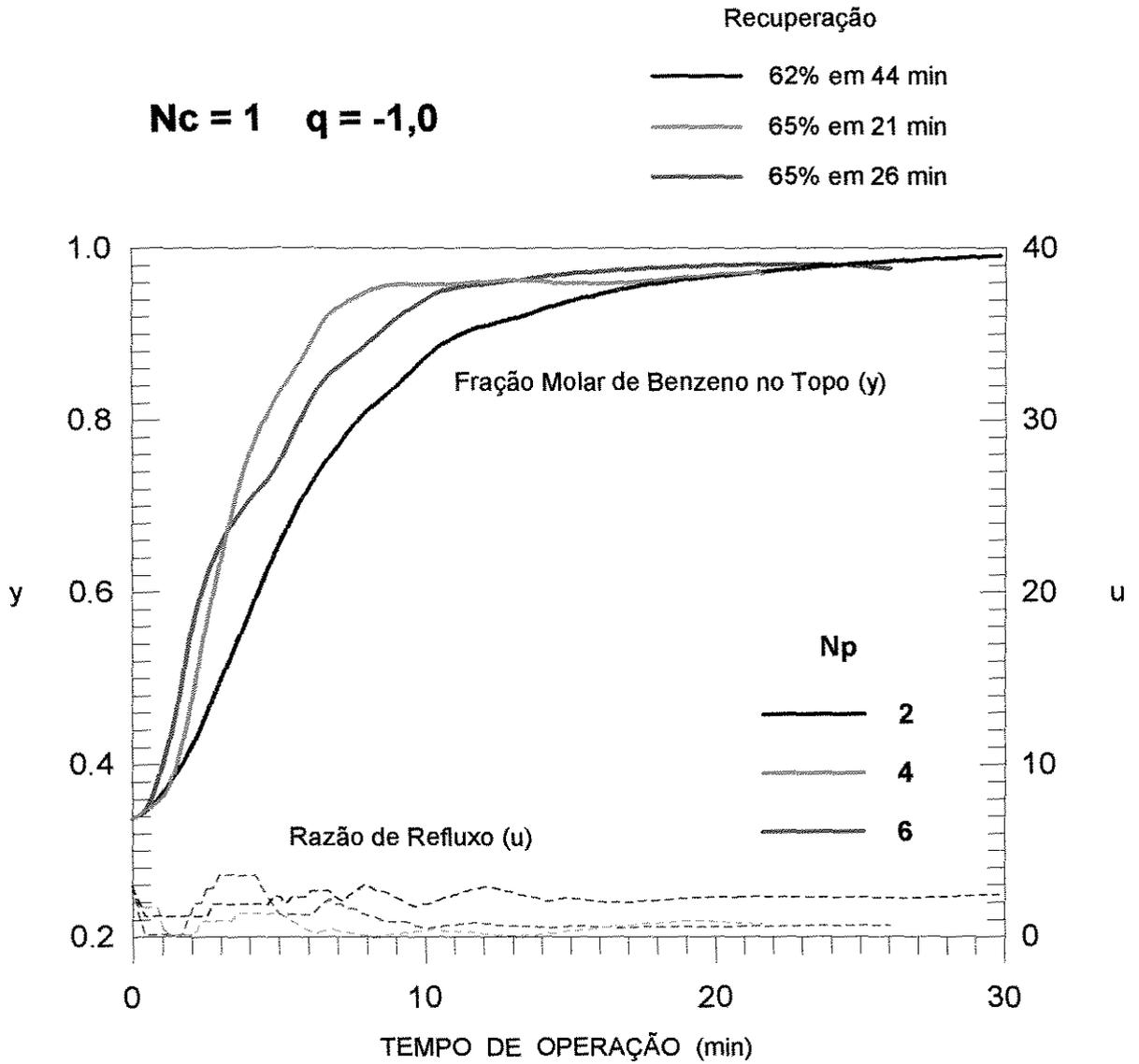


Fig. 4.19 - Influência do Horizonte de Predição sobre a recuperação de benzeno 98% (exemplo 2), quando somente uma ação de controle é otimizada ($N_c=1$) e não se restringe a variação da razão de refluxo ($q=-1,0$).

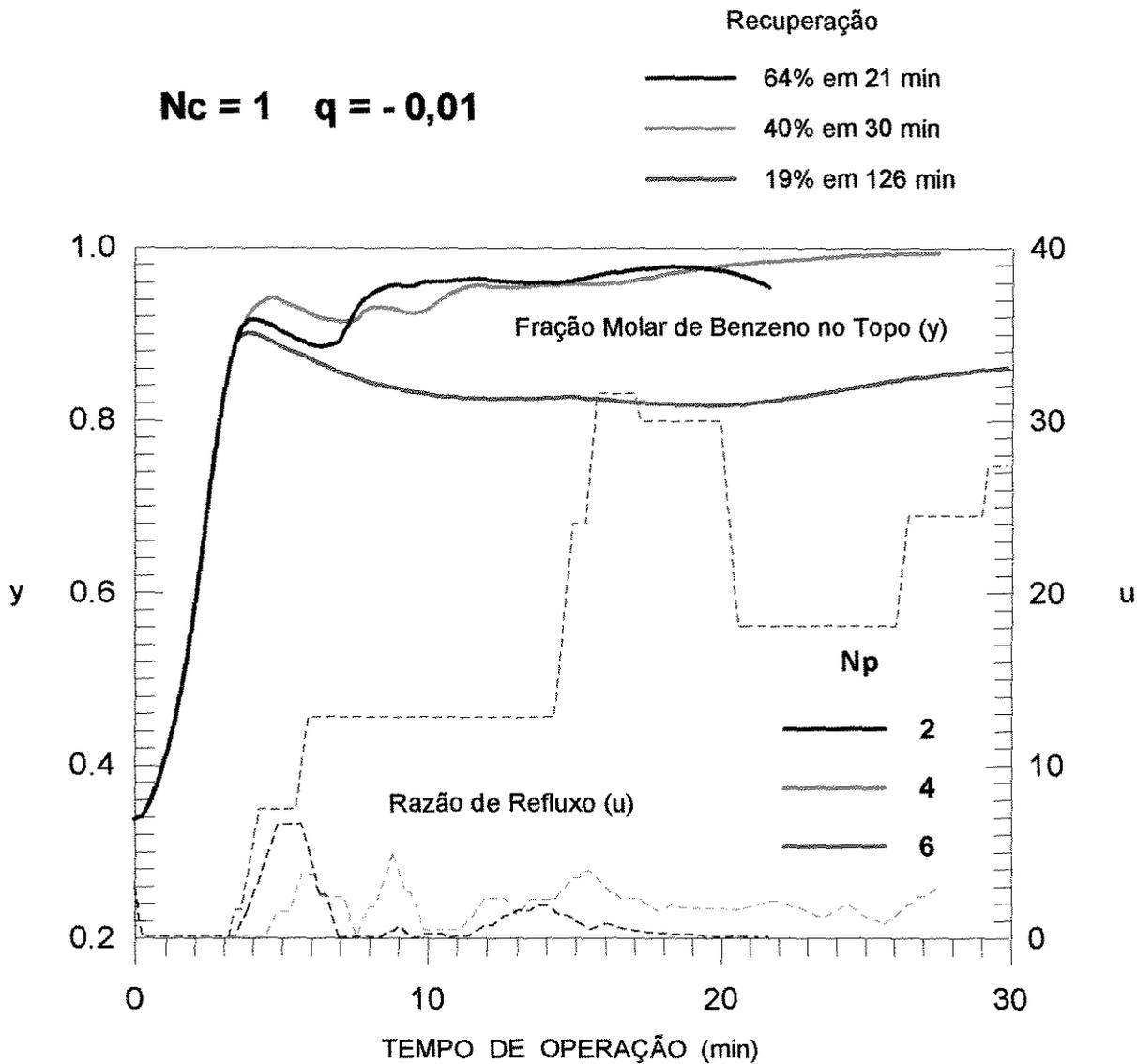


Fig. 4.20 - Influência do Horizonte de Predição sobre a recuperação de benzeno 98% (exemplo 2), quando somente uma ação de controle é otimizada ($N_c=1$) e se restringe a variação da razão de refluxo através de $q=-0,01$.

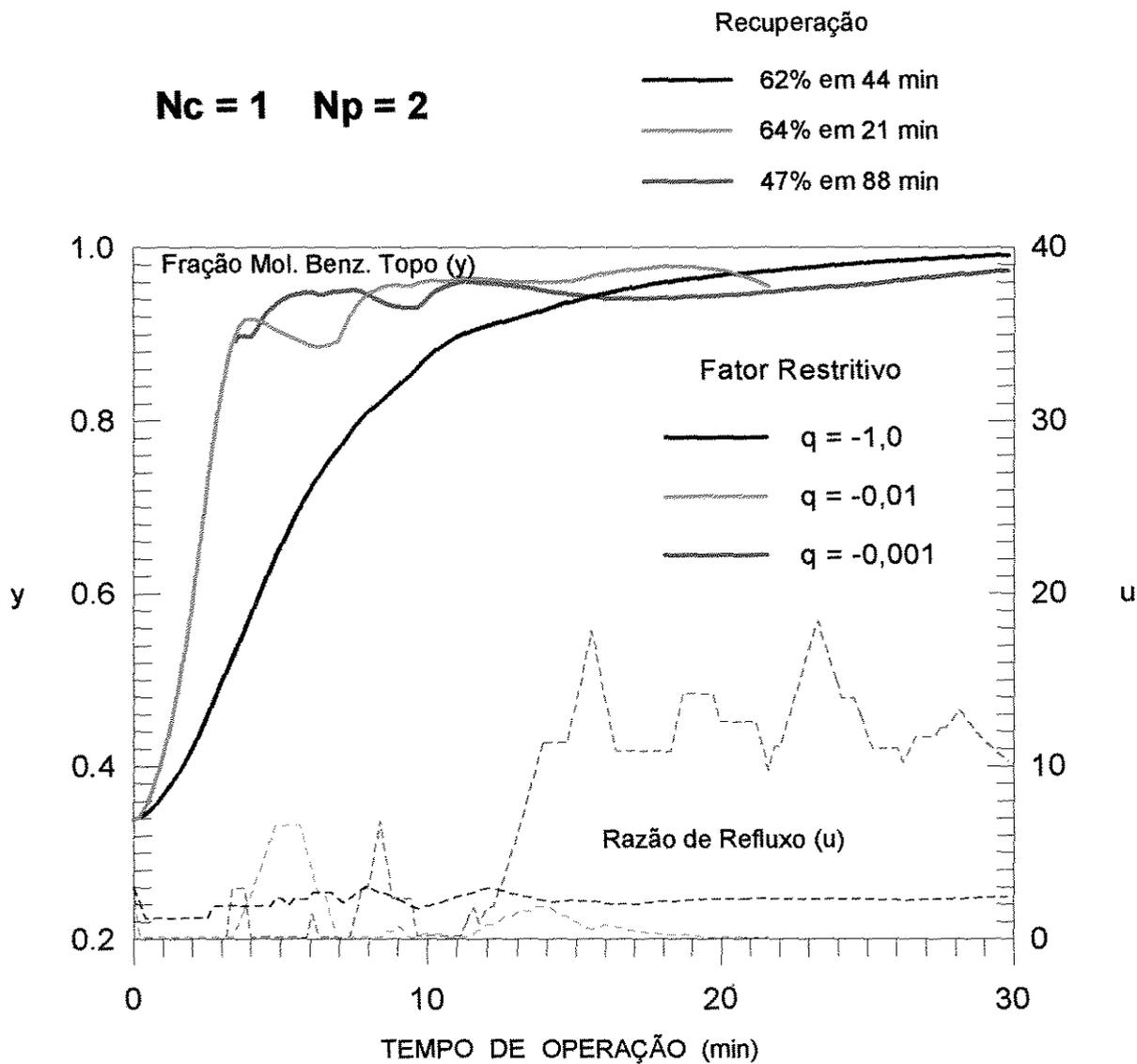


Figura 4.21 - Influência do Fator Restritivo, da variação de razão de refluxo, sobre a recuperação de benzeno 98% (exemplo 2), quando somente uma ação de controle é otimizada ($N_c=1$) e previsões dois instantes futuros são efetuadas ($N_p=2$).

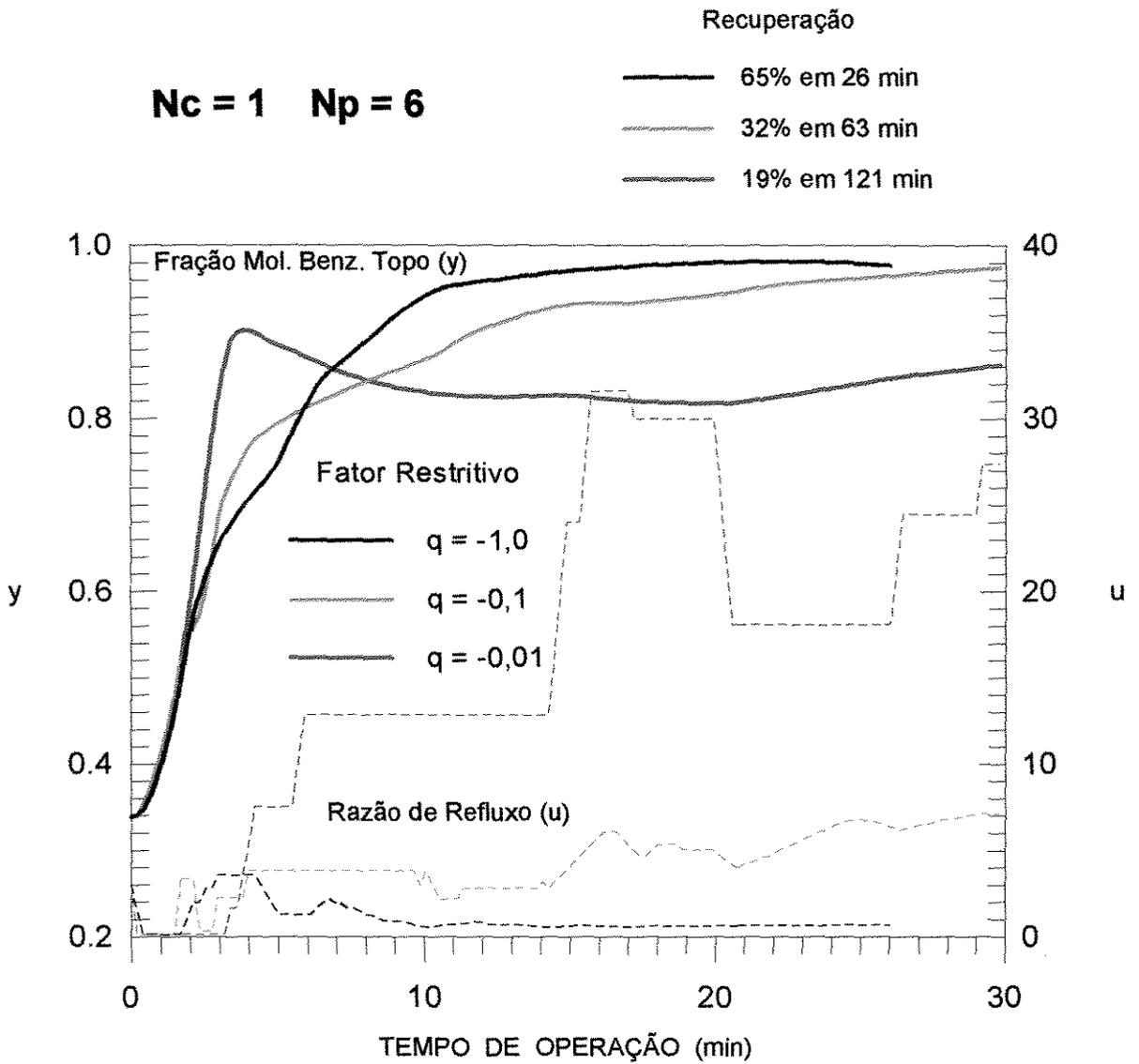


Figura 4.22 - Influência do Fator Restritivo, da variação de razão de refluxo, sobre a recuperação de benzeno 98% (exemplo 2), quando somente uma ação de controle é otimizada ($N_c=1$) e predições seis instantes futuros são efetuadas ($N_p=6$).

A utilização de função objetivo descrita pela equação 4.19 também foi testada. Entretanto, as mudanças na variável manipulada se manifestam somente no valor predito da composição de topo, $\hat{y}(k+1)$, e não tendo o método de otimização utilizado como incorporar restrições, as variações se mostraram absurdas e não implementáveis, provando a importância do segundo termo da equação 4.22 e dos respectivos fatores restritivos neste caso.

Horizontes de controle mais elevados na equação 4.22 também poderiam anular o efeito de horizontes de predição grandes já que aumenta-se o valor do segundo somatório. Sem restrições na variação do refluxo ($q_1=q_2=-1,0$), foram testados na fig. 4.23, valores diferentes para o horizonte de controle, otimizando-se agora ações de controle em dois instantes de amostragem futuros ($N_c=2$). Pode-se então verificar movimentos bastante suaves na ação de controle efetivamente implementada (ação do 1º inst. futuro otimizada).

Numa tentativa de diminuir o tamanho da variação da segunda ação de controle otimizada em relação à do primeiro instante futuro de amostragem e verificar as conseqüências com o horizonte de predição fixo, utilizou-se um valor absoluto de q_2 aproximadamente dez vezes superior a q_1 . Alguns resultados para seis predições futuras ($N_p=6$) se concentram na figura 4.24, onde pode-se observar que a maior liberdade da primeira ação futura não resultou em melhores recuperações de benzeno dentro de especificação quando N_p é grande. A ação acentuada dos primeiros instantes - quando ainda se tem a presença do componente mais volátil, n-hexano - acaba retardando a retirada do componente de interesse, ou seja, benzeno. Como conseqüência das grandes variações, observa-se também a "saturação" do controlador que insiste em manter a razão de refluxo em valores negativos e que o atuador implementa como vazão de refluxo nula.

Com o horizonte de predição menor ($N_p=3$), a figura 4.25 mostra a influência dos fatores restritivos quando $N_c=2$. Utilizando-se de valores absolutos de q menores para compensar N_p pequeno, o controlador se torna praticamente um controlador "on-off", levando a variável manipulada sempre aos seus limites superior e inferior. Apesar da recuperação aceitável de 53% de benzeno da carga inicial, em 21 minutos, da curva de $q_1=-0,0001$ e $q_2=-0,001$, este efeito é totalmente indesejado devido ao desgaste dos elementos finais de controle, além do que estas variações bruscas talvez sejam ineficazes, dependendo do atuador utilizado.

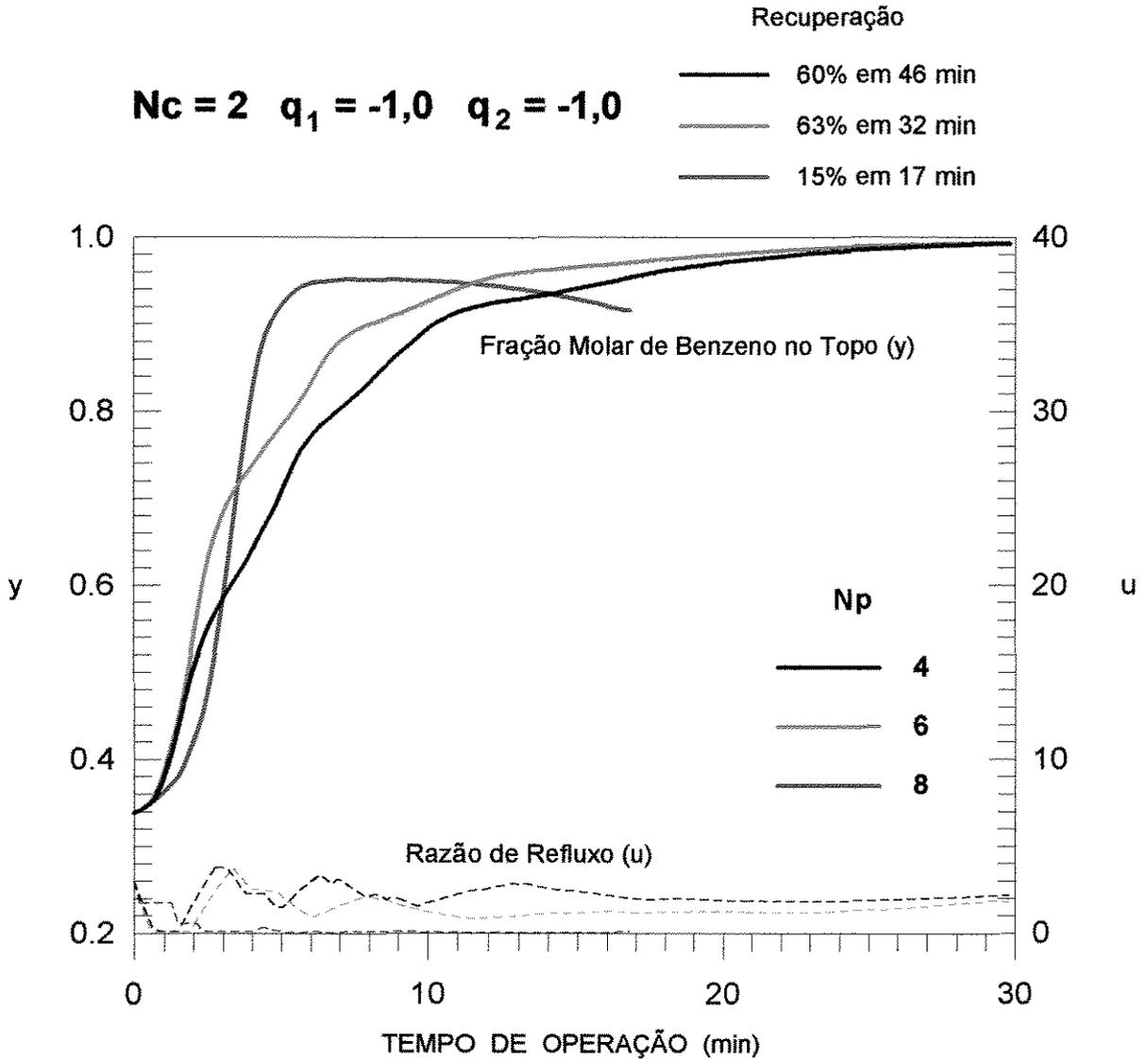


Figura 4.23 - Influência do Horizonte de Predição sobre a recuperação de benzeno 98% (exemplo 2), quando duas ações de controle são otimizadas ($N_c=2$) e não se restringe a variação da razão de refluxo ($q=-1,0$).

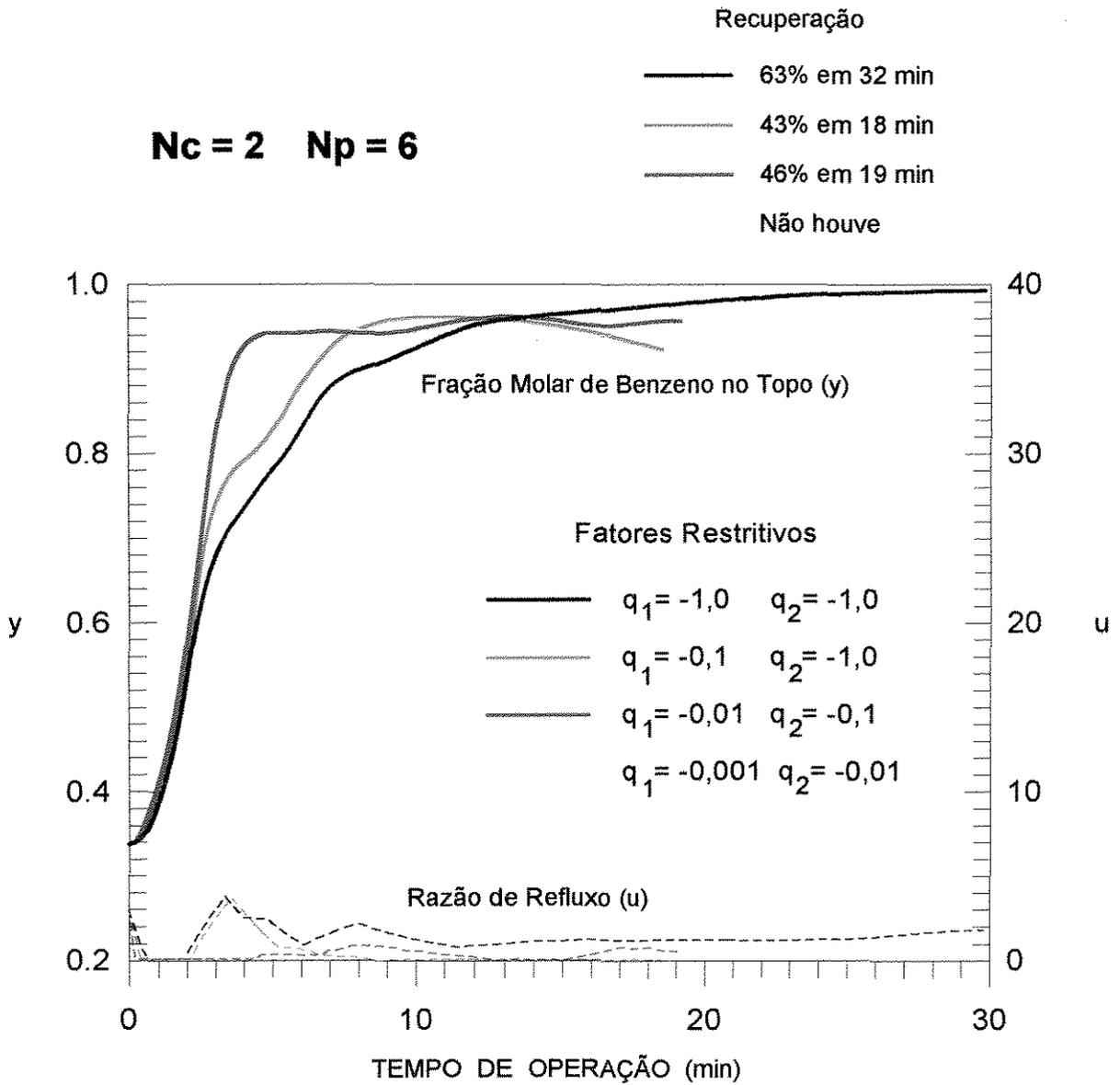


Figura 4.24 - Influência do Fatores Restritivos das duas ações de controle futuras otimizadas ($N_c=2$), sobre a recuperação de benzeno 98% (exemplo 2), quando predições seis instantes futuros são efetuadas ($N_p=6$).

Nc = 2 Np = 3

Fatores Restritivos

Recuperação

—	$q_1 = -0,001$	$q_2 = -0,01$	—	46% em 30 min
—	$q_1 = -0,0001$	$q_2 = -0,001$	—	53% em 21 min
—	$q_1 = -0,00001$	$q_2 = -0,0001$	—	41% em 120 min

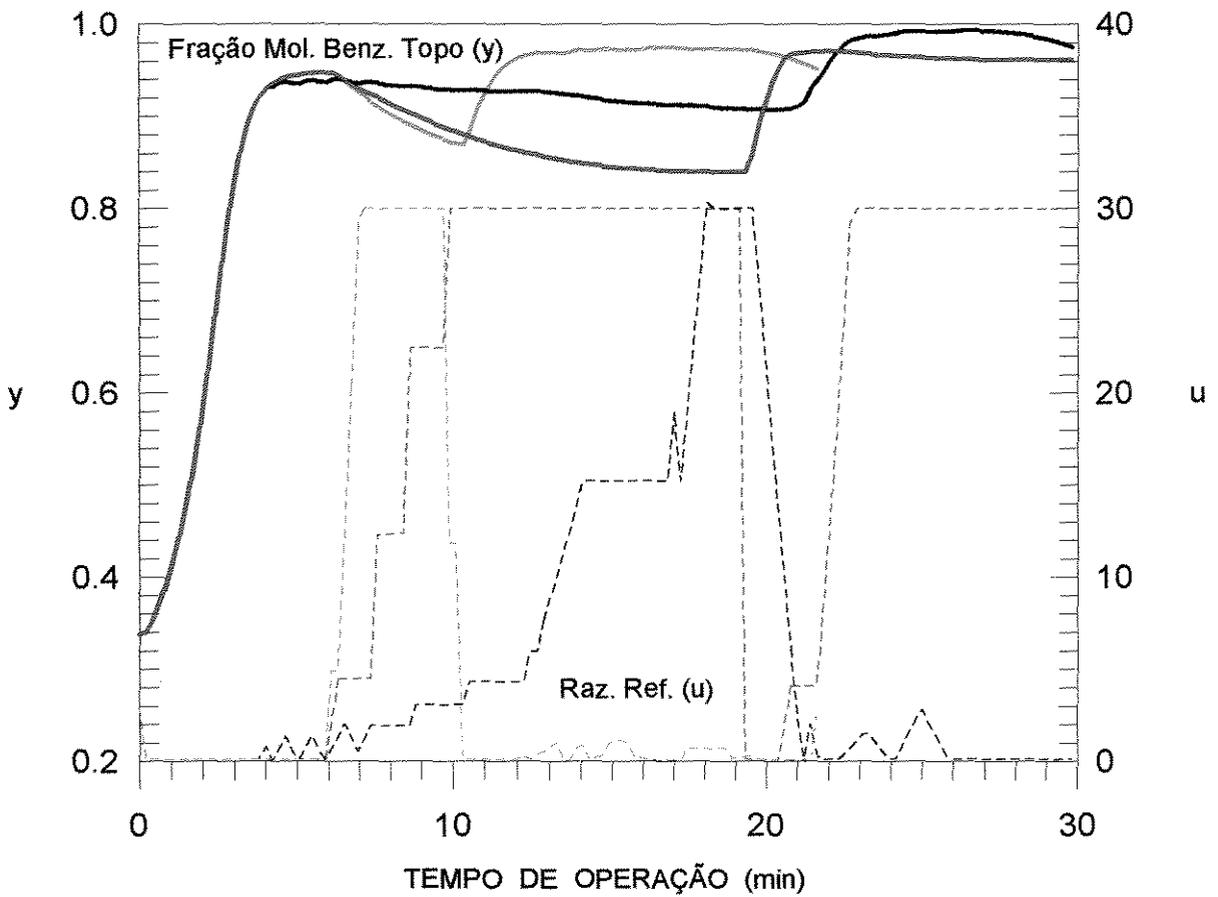


Figura 4.25 - Influência do Fatores Restritivos das duas ações de controle futuras otimizadas ($N_c=2$), sobre a recuperação de benzeno 98% (exemplo 2), quando predições três instantes futuros são efetuadas ($N_p=3$).

Utilizando-se $N_p=N_c=2$, os resultados não foram comparáveis aos anteriormente obtidos com horizontes de predição maiores. Resultados da utilização de $N_c=2$ e $N_p=4$ foram bastante dispersos para este exemplo, não apresentando tendências conclusivas a respeito da melhor escolha dos fatores restritivos e quase sempre resultando desvios finais que levavam a fração molar de benzeno próxima da unidade. Apesar da fração molar estar dentro do desvio permitido para o caso (ver tabela 3.2), razões de refluxo altas como as obtidas com estes valores aumentariam muito o tempo total de operação.

O melhor resultado obtido com a otimização de ações de controle em dois instantes futuros de amostragem ($N_c=2$) foi o então apresentado na figura 4.24 pela curva dos fatores $q_1=q_2=-1,0$, com uma recuperação de 63% em 32 minutos, mas ainda não comparável aos mais fácil e rapidamente obtidos resultados quando $N_c=1$.

4.7 ANÁLISE

No presente capítulo, a modelagem de processos via RNA's foi detalhada com o objetivo de implementação do modelo dentro de um esquema de controle preditivo. A viabilidade de utilização das mesmas na identificação em linha de processos e previsão de comportamento foi demonstrada através de exemplos, onde o tempo computacional gasto pela etapa de treinamento foi levado em consideração, provando não constituir empecilho quando uma escolha criteriosa é feita para a arquitetura da rede a ser utilizada. Com o treinamento efetuado, a RNA realiza simulações praticamente instantâneas do processo, cuja precisão depende quase que exclusivamente do valor de desvio utilizado para o treinamento supervisionado da rede.

Na escolha da arquitetura da rede para representar o processo em estudo, cada variável indicativa das condições operacionais do mesmo, que seja mensurável ou inferenciável, deve ser representada por pelo menos um neurônio artificial na camada de entrada da rede. Dentre estas devem-se encontrar as variáveis controladas e manipuladas, sendo que as primeiras também devem fazer parte da camada de saída quando o objetivo for implementação de sistemas de controle preditivo.

Em casos onde o horizonte de predição seja maior que apenas um instante futuro de amostragem, informações preditas pela rede devem alimentar ela própria, tornando-a recursiva, e portanto o número de neurônios da camada de saída fica atrelado ao número de básico de neurônios de entrada menos o número de variáveis controladas, que são obtidas do método de otimização. Se o processo operar em regime transiente, um histórico deste conjunto básico de informações deve ser incorporado na camada de entrada, constituindo uma janela móvel de informações de entrada que acomoda também os atrasos inerentes ao processo.

O número de neurônios ocultos deve ser estabelecido através da análise dos resultados de testes de funcionamento do equipamento real ou simulações do processo em malha aberta e comparação de desvios resultantes e tempos requeridos para treinamento da rede.

Se a rede que apresentar melhores resultados levar tempo muito grande de treinamento em relação ao intervalo de amostragem, podem-se utilizar valores diferentes para a taxa de aprendizado (η) e para o coeficiente de momentum (α), verificando sempre se com estes valores os desvios de saída permitidos podem ser alcançados e a convergência obtida.

Para a escolha dos horizontes de predição e controle, testes preliminares do equipamento ou simulações em malha fechada têm que ser realizadas e analisadas, onde alguns conselhos práticos podem ser utilizados como critério para uma estimativa inicial, em casos nos quais se utilizam métodos de otimização que não incorporem restrições de processo:

- é fundamental que seja utilizada uma função objetivo que permita restringir variações excessivas das variáveis manipuladas;
- partindo-se de um modelo de predição confiável e desejando-se utilizar horizonte de predição grande, devem-se elevar os valores absolutos dos fatores restritivos, q_i , para penalizar excessos;
- se a confiança no modelo não for grande e portanto a utilização de horizontes de predição pequenos for efetuada, devem-se utilizar valores absolutos menores para os fatores restritivos para que as variações não sejam tão lentas;

- a utilização de N_c maiores que a unidade leva o método de otimização a cálculos mais exaustivos, tornando-o computacionalmente mais lento. Além disso, a seleção de parâmetros ajustáveis fica mais complexa, dificultando a sintonia.

Tratando de uma maneira simples o problema de controle como ele realmente se apresenta no equipamento real - sendo afetado tanto pelas variações de refluxo como pelas variações na composição de fundo - a recuperação do componente prioritário se mostra mais efetiva com a utilização do Controle Preditivo (MPC) baseado em Redes Neurais do que com os controladores adaptativos tratados no capítulo 3.

Os melhores índices obtidos de recuperação do MPC em relação aos controladores adaptativos, com a destilação sujeita às mesmas condições de taxa de evaporação (mantida pela malha de controle de aquecimento na base) comprovam um melhor aproveitamento energético do processo em estudo quando submetido ao sistema de controle preditivo. Sendo os processos de destilação caracterizados por elevado consumo de energia e diante da atual importância da busca de maior eficiência energética, o MPC mostra-se uma ferramenta bastante importante na busca deste objetivo.

CAPÍTULO 5 - CONCLUSÕES E SUGESTÕES.

5.1 CONCLUSÕES

Diante da proposta de tese apresentada após análise crítica da literatura (item 2.6), verifica-se o cumprimento dos objetivos de projeção e implementação de sistemas de controle digitais que ainda não tivessem sido utilizados em destilação batelada. Estes deveriam principalmente não fixar seus parâmetros de sintonia devido ao comportamento intrinsecamente transiente do processo em questão.

Seguindo a mesma ordem descrita de implementação dos tipos de controladores sugeridos, analisa-se em primeiro lugar o Controlador Adaptativo Programável. Sendo uma das maneiras mais simples de se efetuarem as adaptações necessárias no controlador, os cálculos efetuados em linha são extremamente rápidos e permitem que um simples processador do tipo PC-XT acoplado à coluna seja suficiente para implementação do mesmo. Entretanto, o tempo consumido para projeto do sistema, no qual determinam-se parâmetros para diversas condições operacionais do processo através de simulações exaustivas, e a necessidade de conhecimento prévio das propriedades dos componentes da mistura a ser processada, o tornam uma escolha aceitável somente em casos onde uma determinada coluna é utilizada sempre para destilação de mistura bastante conhecida e de composição de carga invariável. Desde que adequadamente planejado, o desempenho do PAC, inclusive para misturas multicomponentes, mostrou-se bastante satisfatório.

Visando eliminar a dependência do projeto do controlador do conhecimento das propriedades da mistura a ser processada e das exaustivas simulações "off-line", e respeitando o caráter multiproduto comum às colunas em batelada, desenvolveu-se o projeto do Controlador Auto-ajustável. Com sua implementação, todo trabalho de identificação do processo passou a ser realizada recursivamente em paralelo à planta. Para executar esta análise regressiva, especificou-se a relação estrutural entre entradas e saídas do processo por um modelo do tipo polinomial obtido da discretização de uma função de transferência de primeira ordem com tempo morto, suficiente para representar o processo em estudo. Sua estrutura permitiu também variar as leis de controle. Assim sendo, o algoritmo digital equivalente ao PI analógico, ajustado em linha por critério de sintonia de Cohen-Coon, e uma lei de controle obtida do Algoritmo de Dahlin foram individualmente

testados. Das simulações realizadas no capítulo três, o algoritmo PI se mostrou mais sensível, podendo levar o processo à instabilidade, quando submetido a estimativas iniciais inadequadas fornecidas ao método matemático utilizado para localização de raízes na etapa de identificação do processo. Em relação à quantidade recuperada de produto dentro de especificação, o algoritmo de Dahlin foi superior em todos os casos, desde que seus parâmetros de projeto fossem adequadamente supridos através de testes preliminares do equipamento (ou simulações) em malha fechada.

Apesar da maior complexidade de cálculos a serem efetuados em linha em relação ao Controlador Programável, desde que se forneçam estimativas iniciais adequadas e se limitem as mudanças das variáveis manipuladas para garantia de convergência mais rápida do método de identificação do processo, o Controlador Auto-ajustável pode ser normalmente implementado através de um PC-386 acoplado ao processo.

Procurando-se por uma representação também não-fenomenológica, porém não-linear e auto-ajustável do processo, projetou-se um controlador baseado em técnica de Inteligência Artificial cuja utilização em Controle de Processos vem crescendo exponencialmente nos últimos anos, as RNA's. A viabilidade de utilização das mesmas na identificação em linha de processos e previsão de comportamento foi demonstrada através de exemplos, onde o tempo computacional gasto pela etapa de treinamento foi levado em consideração, provando não constituir empecilho quando uma escolha criteriosa é feita para a arquitetura da rede a ser utilizada. Com o treinamento efetuado, a RNA realiza simulações praticamente instantâneas do processo, cuja precisão depende quase que exclusivamente do valor de desvio utilizado para o treinamento supervisionado da rede.

Fornecendo então, as RNA's treinadas, previsões de comportamento da planta em determinado horizonte de predição, pode-se implementar um sistema de controle preditivo não-linear baseado em modelo. Através das simulações realizadas, obtiveram-se alguns conselhos de ordem prática para a escolha da arquitetura da rede a ser utilizada e para a escolha dos horizontes de predição e controle. Estas escolhas envolvem testes preliminares com o equipamento em malha aberta (ou simulações), assim como a maioria dos controladores avançados.

Tratando de uma maneira simples o problema de controle como ele realmente se apresenta no equipamento real - sendo afetado tanto pelas variações de

refluxo como pelas variações na composição de fundo - a recuperação do componente prioritário se mostra mais efetiva com a utilização do Controle Preditivo (MPC) baseado em Redes Neurais do que com os controladores adaptativos tratados no capítulo 3.

Os melhores índices obtidos de recuperação do MPC em relação aos controladores adaptativos, com a destilação sujeita às mesmas condições de taxa de evaporação (mantida pela malha de controle de aquecimento na base) comprovam um melhor aproveitamento energético do processo em estudo quando submetido ao sistema de controle preditivo. Sendo os processos de destilação caracterizados por elevado consumo de energia e diante da atual importância da busca de maior eficiência energética, o MPC mostra-se uma ferramenta bastante importante na busca deste objetivo.

Estes resultados positivos do Controlador Preditivo com RNA confirmam adequadamente a habilidade de adaptação e de incorporação de não-linearidades do modelo que imita o comportamento inteligente do cérebro humano, no qual se baseia. Além de representar o processo de maneira efetiva, sem descrever os fenômenos envolvidos e portanto sem necessidade de conhecimento das características da mistura processada e do processo em si, as RNA's apresentam baixa sensibilidade a ruídos e informações deturpadas vindas dos sensores uma vez que cada neurônio artificial de entrada incorpora uma parte muito pequena do problema, ao contrário de modelos empíricos ou ajustados.

Da maneira com que foi projetado no presente trabalho, o Controlador Auto-ajustável também pode ser aplicado a outros processos que não a destilação em batelada, desde que este processo possa ser representado por uma função de transferência semelhante à utilizada. Este sistema de controle (STR) foi projetado apenas para atuar como controlador regulatório do processo e sua configuração determinada para problema SISO..

Em contrapartida, o "software" de implementação do Controlador Preditivo com RNA é totalmente genérico, podendo ser utilizado para qualquer equipamento, qualquer problema de controle (supervisório ou regulatório) e número de variáveis controladas e manipuladas (SISO/MISO/MIMO), desde que seja inserida a nova função objetivo do problema e seja fornecida a melhor estrutura da rede pela escolha das variáveis mais significativas do processo e sejam realizados testes do processo a ser controlado em malha aberta.

Conforme descrito no item que trata da implementação do Controlador Preditivo baseado em RNA's, dois algoritmos de otimização são acionados a cada amostragem: um para treinamento da rede (Regra de Delta Generalizada) e outro para otimização das ações de controle. Seguindo conselho da literatura objetivando agilização de cálculos, para otimização das ações de controle utilizou-se um método independente do cálculo da função gradiente da função objetivo e nenhuma restrição foi incorporada (Método de Powell).

O tempo máximo gasto pelos cálculos efetuados "on-line" não pode ser avaliado de maneira exata por verificar-se que o mesmo depende da não-idealidade do sistema destilado, da estrutura da RNA, do momento em que se encontra a operação e das tolerâncias e parâmetros utilizados nas otimizações. Porém, com as configurações utilizadas para controle no capítulo quatro, foi possível a utilização de intervalos de amostragem de seis segundos num equipamento computacional do tipo PC-486 DX2 de 66 MHz.

Com a possibilidade de atuação conjunta regulatória e supervisória sobre o processo, o MPC pode ser utilizado com a mesma instrumentação básica proposta (fig. 2.6) para a implementação de uma estratégia ótima preestabelecida, tanto de composição de topo quanto de energia fornecida ao refeedor, para misturas de propriedades conhecidas. Quando possível, este é sem dúvida um caminho direto para a busca incessante por processos energeticamente mais eficientes.

5.2 SUGESTÕES

Como primeira proposta para complementação do presente trabalho, sugere-se a utilização de um método de otimização de ações do controlador que possa incorporar todas as restrições do processo e dos elementos finais de controle, na implementação do Controle Preditivo baseado em Redes Neurais Artificiais. Com a implementação de um método mais rígido, deve-se verificar principalmente o grau de comprometimento do tempo gasto em cada intervalo de amostragem frente ao possível benefício atingido através de cálculos mais exatos das futuras ações de controle.

Quando houver casos em que a estrutura escolhida para a Rede Neural a ser utilizada no Controlador Preditivo, assim com o método de otimização de ações de controle, levarem tempo muito elevado para treinamento em relação ao intervalo de amostragem, deve ser realizado um estudo de utilização de diferentes valores de taxa de aprendizado e/ou coeficiente de “momentum”. Pode-se utilizar também uma variante do método de ajuste de pesos na qual ocorre uma mudança dinâmica da taxa de aprendizado de acordo com o comportamento da função residual na camada de saída da rede. Durante o treinamento, se o resíduo diminuir, aumenta-se o valor da taxa de aprendizado e se o resíduo aumentar, desacelera-se o mesmo.

Visando ainda a redução do tempo de treinamento da RNA, para simulação de processos dinâmicos, ao invés de se utilizar maior número de neurônios artificiais na camada de entrada, sugere-se a alternativa de incorporação da dinâmica do processo na própria transmissão entre os neurônios através da passagem de sinais por transformadas de Laplace do tipo polinomiais.

Para implementação do sistema de controle de melhor desempenho em equipamento real, sugere-se ainda a otimização prévia de funções-objetivo de produtividade ou lucro, sempre que as características da mistura forem bem conhecidas.

Também para a implementação prática, sugere-se o estudo da possibilidade de utilização de Redes Neurais Artificiais como “soft-sensores”, ou seja, uma alternativa não-convencional de inferenciação de variáveis de difícil medição (composições de topo e fundo da coluna), com objetivo de aumentar a confiabilidade de sistemas de controle onde a inferenciação seja necessária.

**ANEXO 1 - SIMULADOR DA COLUNA DE DESTILAÇÃO
MULTICOMPONENTE EM BATELADA.**

```

/* *****
* Simulação de Colunas de Destilação Multicomponente em Batelada (COLBAT.C) *
***** */

```

/*Suposições:

- 1) Fluxos molares "não" são constantes;
- 2) Somente o refeedor é considerado estágio ideal;
- 3) Refluxo total é considerado para estimativa do perfil de composição inicial;
- 4) Hold-up constante ao longo da coluna e considerado na estimativa inicial;
- 5) Hold-up de vapor desprezível;
- 6) Hold-up de líquido variável nos pratos;
- 7) Hold-up constante e suficientemente pequeno no condensador para que se possa considerar a temp. do recipiente igual a do prim. prato;
- 8) Condensador total;
- 9) Eficiência constante nos pratos;
- 10) Queda de pressão desprezível;
- 11) Calor fornecido ao refeedor varia para manter $V[n+1]$ constante; (ou seja, deve haver controle no sistema de aquecimento);
- 12) Pressão deve ser baixa ou moderada devido ao uso da eq. virial truncada no segundo termo e eq. de antoine. */

```

#include "math.h"
#include "stdio.h"

```

```

void somatx();
void somaty();
void ental();
void entvap();
void bolha();
void orvalho();
void uniq();
void viri();
void fugpad();
void CONTROLE();

```

```

int i,j,n,c,bo=1,it,polar[9],PH[9];
float x[9][18],y[9][18],E,gam[9],fi[9],fug0[9],K[9],t=0.0;
float Pc[9],Tc[9],cpl[9],cpv[9],Teb[9],To,P,alfa;
float h[18],H[18],he[18],Hvapr,Hvapc,Hvpteb[9],Hvap[9],Qc,Qr,T[18],hant[18];
float mdp[9],at[9],bt[9],omega[9],B[9][9]; /* virial (viri) */
float Vc[9],Aanto[9],Banto[9],Canto[9]; /* Antoine (fugpad) */
float unr[9],unq[9],una[9][9]; /* Parametros Uniquac (uniq) */

```

```

void main(void)

{

char arq[30];
float bas,taux;
float *pR,pRant,*px[20],pxan,*pxref[20];
int iter,ii,ced=0,ent,aux1=0,aux2=1,ctem=0,ctem0=4,cont=0,contr=1,Vfix,na=1,pert;
float MANIP[23],CONTRO[23],TAUX=0.0,TAUX2=0.0;
float fun[9][18],k1[9][18],k2[9][18],k3[9][18],k4[9][18];
float xant[9][18],xsp[9],xmed[10],xc[18],xan[18],corr;
float Mc,So,Mrt,ro[18],d[18],fe,Ap,Lw,hv,rom; /* Holdup's */
float Raz,Lo,D,Dacum=0.0,Sini=0.0,pda=0.0,pb=-2.0;
float pas,tf,timp1=0.0,timp2=0.0,tx=1.e-20,soma,kc,Ti,Iant=0.0,I=0.0;
float desv,dif,fun1,V1,erant=0.0;
float xo[9],totcl[18],carg[18],somcl[18],som[18],Q[18],*LUC[20],delta[18];
float km1[18],km2[18],km3[18],km4[18],func[18],L[18],M[18],Mant[18],V[18];
float somcom[9],somdes[9],compmed[9];
FILE *a1;
FILE *fp1;
FILE *fp2;
FILE *fp3;
FILE *fp4;
if((fp1=fopen("topo.dat","w+b"))==NULL) {
    printf ("\n  Cannot open file\n\n");
    goto fim; }
if((fp2=fopen("percom.dat","w+b"))==NULL) {
    printf ("\n  Cannot open file\n\n");
    goto fim; }
if((fp3=fopen("pertem.dat","w+b"))==NULL) {
    printf ("\n  Cannot open file\n\n");
    goto fim; }
if((fp4=fopen("parproc.dat","w+b"))==NULL) {
    printf ("\n  Cannot open file PARPROC.DAT \n\n");
    goto fim; }
printf ("\n\n          COLUNA DE DESTILACAO BATELADA \n\n\n");
printf ("\n Digite o num. de compon. e o num. de pratos:\n");
scanf ("%d,%d",&c,&n);
printf ("\n De o nome do arquivo de entrada de dados: ");
printf ("\n\n cuidado com as constantes de antoine -log,mmHg,Celsius ! \n");
scanf ("%s",arq);
if((a1=fopen(arq,"r")) == NULL) {
    printf ("\n\n  Nao existe o arquivo %-s !!!!!\n\n",arq);
    goto fim; }

fscanf(a1,"%f %f %f %f",&Ap,&Lw,&hv,&E);
fscanf(a1,"%f %f %f",&kc,&Ti,&desv);

```

```

fscanf(a1, "%f %f %f %d %f %f", &So, &Mc, &P, &Vfix, &V[n+1], &Qr);
fscanf(a1, "%f %f %f", &pas, &tf, &To);
for (i=0; i<c; i++) /* le primeiro todos Pc, Vc, Tc, Teb p/ cada comp. */
{ fscanf(a1, "%f %f %f %f", &Pc[i], &Vc[i], &Tc[i], &Teb[i]);
}
for (i=0; i<c; i++)
{ fscanf(a1, "%f %f %f %f", &cpl[i], &cpv[i], &d[i], &Hvap[i]);
}
for (i=0; i<c; i++)
{ fscanf(a1, "%d %d %f %f %f", &polar[i], &PH[i], &mdp[i], &at[i], &bt[i]);
}
for (i=0; i<c; i++)
{ fscanf(a1, "%f %f %f %f", &omega[i], &Aanto[i], &Banto[i], &Canto[i]);
}
for (i=0; i<c; i++)
{ fscanf(a1, "%f %f %f %f", &xo[i], &xsp[i], &unr[i], &unq[i]);
}
for (i=0; i<c; i++)
{ for (ii=0; ii<c; ii++)
{ fscanf(a1, "%f ", &una[i][ii]);
}
}
fclose(a1);

/* Cálculo do hold-up máximo de cada prato (refluxo total - L[j]=V[n+1]) */
fe = 9.345e-3; /* constante para eq. Francis */
rom = 0;
for (i=0; i<c; i++)
{ rom = rom + (d[i] * xo[i]); /* massa específica molar da mistura inicial */
}
printf ("\n Vn+1 = %f\n", V[n+1]);
Mrt = Ap * rom * (hv + (fe * pow((V[n+1]/(rom * Lw)), (2./3.)))); /* Eq. Francis */
printf ("\n Mrt = %f\n", Mrt);
Sini = So - (Mrt * n) - Mc;
M[n+1] = Sini;

/* iniciacao */

for (j=1; j<=n; j++) {
M[j] = Mrt; /* Considera-se hold-up inicial igual p/ todos pratos */
}
To = 273.15 + To;
P = P * 101.325;
for (i=0; i<c; i++) {
Tc[i] = 273.17 + Tc[i]; /* Transformação das T em C para K */
Teb[i] = 273.15 + Teb[i];
Pc[i] = Pc[i] * 101.325; /* Conversão da Pc de atm para kPa */
}

```

```

Q[i] = 0.0;
*LUC[i] = 0.0;
somcom[i] = 1.e-5;
somdes[i] = 1.e-5;
}

/* Estimativa da concentração inicial no condensador */

printf("\n Estimativa inicial da concentracao no condensador: \n");

for (i=0;i<c;i++) {
    printf("\n x[i][0] = ");
    scanf ("%f",&x[i][0]);
    y[i][1] = x[i][0];
    xc[i] = x[i][0];
    xan[i] = 0.0;
}

/* cálculo das concentrações iniciais prato a prato a partir da estimativa anterior */

calc;;
printf("\n xcond[0,1] = %f %f, k[0,1] referv. = %f %f",xc[0],xc[1],K[0],K[1]);
for (j=1;j<=(n+1);j++)
    { orvalho); /* Devolverá os verdadeiros K[i]'s e x[i][j]'s */
      if (j!=(n+1))
          { for (i=0;i<c;i++) y[i][j+1] = x[i][j];
            }
        }

for (i=1;i<c;i++)
{ soma = 0;
  somcl[i] = 0;
  for (j=1;j<=n;j++) somcl[i] = somcl[i] + x[i][j];
  fun1 = (x[i][n+1]*Sini)/(xo[i]*So - x[i][0]*Mc - somcl[i]*Mrt) - 1.0;
  printf ("\n fun1 = %f",fun1);
  if (fabs(fun1)<0.1) continue; /* Convergiu, vai para outro componente */
  xan[i] = x[i][0];
  if ((So*xo[i]-Mc*x[i][0]-Mrt*somcl[i])>0.000000)
    { corr = (So*xo[i] - Mc*x[i][0] - Mrt*somcl[i])/(Sini*x[i][n+1]);
      printf ("\n coef. de correcao de x[%d][0] = %f",i,corr);
      x[i][0] = corr*x[i][0];
    }
  else { dif = (x[i][n+1]*Sini - xo[i]*So + Mc*x[i][0] + Mrt*somcl[i]);
        if (dif < 0) x[i][0] = 1.10 * x[i][0];
        else x[i][0] = x[i][0]/1.10;
      }
}
for (i=0;i<c;i++) soma = soma + x[i][0];

```

```

for (i=0;i<c;i++)
  { x[i][0] = x[i][0] / soma;
    xc[i] = x[i][0];
    y[i][1] = x[i][0];
  }
goto calc;
}

/* Cálculo do calor lat.mol. de vap. de cada comp. a temp. normal de ebulição */

for (i=0;i<c;i++)
  { Hvpteb[i] = (Teb[i] * 2.17 * (log(Pc[i]/101.325) - 1))/(0.930 - Teb[i]/Tc[i]);
  } /* Teb em Kelvin, Pc em atm, Hvpteb em cal/mol */

/* Estimativa inicial dos fluxos molares de liq. e vapor */

for (j=1;j<=(n+1);j++)
  { bolha();
    ental();
  }
entvap();
if (Vfix==1) Qr = V[n+1] * Hvapr; /* calor fornec. ao ref. para manter fluxo V[n+1] cte. */
else V[n+1] = Qr/Hvapr; /* vazão de vapor do refeedor se Qr for cte. e não V[n+1] */
L[n] = V[n+1];
for (j=n;j>0;j--)
  { V[j] = (V[j+1] * (h[j] - H[j+1]))/(h[j-1] - H[j]);
    L[j-1] = V[j];
  }
Qc = V[1] * Hvapc; /* calor retirado no cond. p/ vapor se tornar líquido sat. */
printf ("\n Perfil de conc. inicial dos comp. 1,2,3 e 4 e de temp.:\n");
for (j=0;j<=(n+1);j++) printf ("%f %f %f %f %f\n",x[0][j],x[1][j],x[2][j],x[3][j],T[j]);

bo = 2;
L[n+1] = 0.0;
V[0] = 0.0;

Raz = 1.e-5;

do {
  pR = &Raz;
  for (i=0;i<c;i++)
    { px[i] = &x[i][0];
      pxref[i] = &x[i][n+1];
    }

  V1 = 0.97 * V[1];
  taux = TAUX;

```

```

CONTROLE(&na,pR,&pRant,px,LUC,&pxan,pxref,c,xsp,desv,&ced,&aux1,&aux2,&ctem,
&ctem0,&contr,&cont,xmed,&tx,MANIP,CONTRO,&TAUX,&TAUX2,&pert,&kc,&Ti,
&Iant,&I,&V1,&erant); /* chamada do gerenciador de controle */

```

```

Raz = *pR;
L[0] = (Raz/(Raz+1)) * V[1];
if (L[0]<1.e-20)
  { Raz = 1.e-20;
    L[0] = 1.e-20;
  }
if (L[0]>(0.99*V[1])) L[0] = 0.99*V[1];
D = V[1] - L[0];

```

```

somcom[ced] = somcom[ced] + x[ced][0]*D;
somdes[ced] = somdes[ced] + D;
compmed[ced] = somcom[ced]/somdes[ced];

```

```

if (x[ced][0]>(xsp[ced]-desv)) Q[ced] = Q[ced] + (x[ced][0] * D * pas);

```

```

if (t>=taux) printf("\n Composicao media no destilado = %f\n",compmed[ced]);

```

```

/* gravação em arquivo */

```

```

if (t>=timp1) { /* grava comp.de todos comp. no cond. */
fprintf (fp1,"%f %f %f %f %f %f ",t,L[0],pda,Qr,Qc,(Q[ced]*100.0/(So*xo[ced]));
for (i=0;i<c;i++)
  { if (i==(c-1)) fprintf (fp1,"%-5.4f %-5.4f\r\n",x[c-1][n+1],x[c-1][0]);
    else fprintf (fp1,"%-5.4f %-5.4f ",x[i][n+1],x[i][0]);
  }
timp1 = timp1 + (10. * pas);
}

```

```

if (t>=timp2) {
fprintf (fp2,"%f ",t); /* grava o tempo e o perfil de todos os compon. */
for (i=0;i<c;i++)
  { for (j=0;j<=(n+1);j++)
    { if (j==(n+1) && i==(c-1)) fprintf (fp2,"%-5.4f\r\n",x[i][n+1]);
      else fprintf (fp2,"%-5.4f ",x[i][j]);
    }
  }
fprintf (fp3,"%f ",t); /* grava o perfil de temper. e vaz.liq. p/ varios t */
for (j=0;j<=(n+1);j++)
  { if (j==(n+1)) fprintf (fp3,"%-6.3f %-6.4f %-6.4f\r\n",(T[n+1]-
273.15),L[n+1],V[n+1]);
    else fprintf (fp3,"%-6.3f %-6.4f %-6.4f ",(T[j]-273.15),L[j],V[j]);
  }
}

```

```

/* impressão na tela do perfil da coluna */

printf("\nTempo  Conc.0  Conc.1  Conc.2  Temper.  Vaz.vap.  Vaz.liq.  Hold-up
\n");
M[0] = Mc;
for (j=0;j<=(n+1);j++) printf ("%6.2f  %5.4f  %5.4f  %5.4f  %6.3f  %4.3f  %4.3f
%6.3f\n",t,x[0][j],x[1][j],x[2][j],T[j],V[j],L[j],M[j]);

    timp2 = timp2 + (10.0 * pas);
    }
t = t + pas;
Dacum = Dacum + D*pas;
pda = (Dacum/Sini) * 100.0;

/* Integração por Runge-Kutta de ordem 4 - BALANÇOS MÁSSICOS */

iter = 1;
do {
    for (i=0;i<c;i++) {
        /* Condensador - Composição */
        fun[i][0] = V[1] * (y[i][1] - x[i][0])/Mc;
        /* Refervedor - Composição */
        fun[i][n+1]=(L[n]*(x[i][n]-x[i][n+1])+V[n+1]*(x[i][n+1]-y[i][n+1]))/M[n+1];
        }
        /* Refervedor - Hold-up */
        func[n+1] = L[n] - V[n+1];
        /* Pratos */
        for (j=1;j<=n;j++)
            { /* Hold-up */
                func[j] = L[j-1] + V[j+1] - L[j] - V[j];
                for (i=0;i<c;i++) {
                    /* Composição */
                    fun[i][j]=(V[j+1]*(y[i][j+1]-x[i][j])+V[j]*(x[i][j]-y[i][j])+L[j-1]*(x[i][j-1]-
                    x[i][j]))/M[j];
                }
            }

        switch(iter) {

            case 1:
                for (j=0;j<=(n+1);j++) {
                    ro[j] = 0.0;
                    for (i=0;i<c;i++) {
                        k1[i][j] = fun[i][j];
                        xant[i][j] = x[i][j];
                        x[i][j] = x[i][j] + (0.5 * pas * k1[i][j]);
                    }
                }
            }
        }
    }

```

```

ro[j] = ro[j] + (x[i][j] * d[i]); /* mas.esp.mol.média no prato j */
    }
    if (j!=0)
    { Mant[j] = M[j];
      km1[j] = func[j];
      M[j] = M[j] + (0.5 * pas * km1[j]);
      if (j!=(n+1))
        { bas = (1./fe) * (M[j]/(Ap*ro[j]) - hv);
          if (bas<1.e-10) L[j] = 0.0;
          else L[j] = ro[j] * Lw * pow(bas,1.5);
        }
      bolha(); /* Eq. Francis */
      ental();
    }
    somatx();
    somaty();
  }
  entvap(); /* refervedor */
  if (Vfix==1) Qr = V[n+1] * Hvapr; /* Consid.V[n+1] cte, Qr tem que variar */
  else V[n+1] = Qr/Hvapr; /* Com Qr cte. quem varia e V[n+1] */

  /* Balanços de Energia */
  for (j=n;j>=1;j--) /* pratos */
  { delta[j] = (h[j] - hant[j])/pas; /* aprox.por dif.finitas de dh/dt */
    V[j] = ((M[j] * delta[j]) + L[j-1] * (h[j] - h[j-1]) + V[j+1] * (h[j] - H[j+1]))/(h[j] -
H[j]);
  }
  Qc = V[1] * Hvapc; /* Condensador */
  break;

case 2:
  for (j=0;j<=(n+1);j++) {
    ro[j] = 0.0;
    for (i=0;i<c;i++) {
      k2[i][j] = fun[i][j];
      x[i][j] = x[i][j] + (0.5 * pas * k2[i][j]);
      ro[j] = ro[j] + (x[i][j] * d[i]); /* mas.esp.mol.media no prato j */
    }
  }
  if (j!=0)
  { km2[j] = func[j];
    M[j] = M[j] + (0.5 * pas * km2[j]);
    if (j!=(n+1))
      { bas = (1./fe) * (M[j]/(Ap*ro[j]) - hv);
        if (bas<1.e-10) L[j] = 0.0;
        else L[j] = ro[j] * Lw * pow(bas,1.5);
      }
    bolha();
  }

```

```

    ental();
  }
  somatx();
  somaty();
}
entvap(); /* refervedor */
if (Vfix==1) Qr = V[n+1] * Hvapr; /* Consid.V[n+1] cte, Qr tem que variar */
else V[n+1] = Qr/Hvapr; /* Com Qr cte. quem varia e V[n+1] */

/* Balancos de Energia */
for (j=n;j>=1;j--) /* pratos */
{ delta[j] = (h[j] - hant[j])/pas; /* aprox.por dif.finitas de dh/dt */
  V[j] = ((M[j] * delta[j]) + L[j-1] * (h[j] - h[j-1]) + V[j+1] * (h[j] - H[j+1]))/(h[j] -
H[j]);
}
Qc = V[1] * Hvapc; /* Condensador */
break;

case 3:
for (j=0;j<=(n+1);j++) {
  ro[j] = 0.0;
  for (i=0;i<c;i++) {
    k3[i][j] = fun[i][j];
    x[i][j] = x[i][j] + (pas * k3[i][j]);
    ro[j] = ro[j] + (x[i][j] * d[i]); /* mas.esp.mol.media no prato j */
  }
  if (j!=0)
  { km3[j] = func[j];
    M[j] = M[j] + (pas * km3[j]);
    if (j!=(n+1))
    { bas = (1./fe)*(M[j]/(Ap*ro[j])-hv);
      if (bas<1.e-10) L[j] = 0.0;
      else L[j] = ro[j] * Lw * pow(bas,1.5);
    }
    bolha();
    ental();
  }
  somatx();
  somaty();
}
entvap(); /* refervedor */
if (Vfix==1) Qr = V[n+1] * Hvapr; /* Consid.V[n+1] cte, Qr tem que variar */
else V[n+1] = Qr/Hvapr; /* Com Qr cte. quem varia e V[n+1] */

/* Balancos de Energia */
for (j=n;j>=1;j--) /* pratos */
{ delta[j] = (h[j] - hant[j])/pas; /* aprox.por dif.finitas de dh/dt */

```

```

    V[j] = ((M[j] * delta[j]) + L[j-1] * (h[j] - h[j-1]) + V[j+1] * (h[j] - H[j+1]))/(h[j] -
H[j]);
    }
    Qc = V[1] * Hvapc; /* Condensador */
    break;

case 4:
for (j=0;j<=(n+1);j++) {
    ro[j] = 0.0;
    for (i=0;i<c;i++) {
        k4[i][j] = fun[i][j];
        x[i][j]=xant[i][j]+pas/6.0*(k1[i][j]+2.0*k2[i][j]+2.0*k3[i][j]+k4[i][j]);
        ro[j] = ro[j] + (x[i][j] * d[i]); /* mas.esp.mol.media no prato j */
    }
    if (j!=0)
    { km4[j] = func[j];
      M[j] = Mant[j] + pas/6.0 * (km1[j]+2.0*km2[j]+2.0*km3[j]+km4[j]);
      if (j!=(n+1))
      { bas = (1./fe) * (M[j]/(Ap*ro[j]) - hv);
        if (bas<1.e-10) L[j] = 0.0;
        else L[j] = ro[j] * Lw * pow(bas,1.5);
      }
      bolha();
      ental();
    }
    somatx();
    somaty();
    }
    entvap(); /* refervedor */
    if (Vfix==1) Qr = V[n+1] * Hvapr; /* Consid.V[n+1] cte, Qr tem que variar */
    else V[n+1] = Qr/Hvapr; /* Com Qr cte. quem varia e V[n+1] */

/* Balancos de Energia */
for (j=n;j>=1;j--) /* pratos */
    { delta[j] = (h[j] - hant[j])/pas; /* aprox.por dif.finitas de dh/dt */
      V[j] = ((M[j] * delta[j]) + L[j-1] * (h[j] - h[j-1]) + V[j+1] * (h[j] - H[j+1]))/(h[j] -
H[j]);
    }
    Qc = V[1] * Hvapc; /* Condensador */
    break;
}

iter = iter + 1;

} while (iter<=4);

```

```

if (pb<0.00001)
{ if (x[c-1][0]>=xsp[c-1])
    { pb = t;
      printf ("\n VAI SAIR !!!\n");
      goto fim;
    }
  } /* Poderia parar a batelada e recolher c-1 no fundo da coluna */

} while (t<=tf && M[n+1]>V[n+1] && V[n+1]>0.0000000000000001);

fim:
printf ("\nxo={%-3.2f,%-3.2f,%-3.2f,%-3.2f}, ced=%d, desv=%-5.4f\n",
xo[0],xo[1],xo[2],xo[3],ced+1,desv);
printf ("passo=%-5.4f\n",pas);
printf ("xsp={%-3.2f,%-3.2f,%-3.2f,%-3.2f}, aux1=%d, aux2=%d\n",
xsp[0],xsp[1],xsp[2],xsp[3],aux1,aux2);
printf ("\n %% moles acima espec.no dest.={%-4.2f,%-4.2f,%-4.2f,%-4.2f}, tempo
total=%-6.3f\n",
(Q[0]*100.0/(So*xo[0]),(Q[1]*100.0/(So*xo[1]),(Q[2]*100.0/(So*xo[2]),(Q[3]*100.0/(
So*xo[3])),t);
printf ("\n\n Batelada poderia ter sido finalizada no tempo %f !!!",pb);
printf ("\n (O componente c-1 seria recolhido no fundo da coluna)");
fclose(fp1);
fclose(fp2);
fclose(fp3);
fclose(fp4);
}

void somatx()
{
float somx=0.0;
for (i=0;i<c;i++) {
    somx = somx + x[i][j]; }
if (fabs(1.0-somx)>0.01) {
    printf ("\n somx = %f no prato %d no tempo %f\n",somx,j,t);
    printf ("  CUIDADO !!!!!!!!!!!!! \n");
    return; }
}

void somaty()
{
float somy=0.0;
if (j!=0) {
for (i=0;i<c;i++) {
    somy = somy + y[i][j]; }
}
}

```

```

if (fabs(1.0-somy)>0.01) {
    printf ("\nsomy = %f no prato %d no tempo %f\n",somy,j,t);
    printf ("  CUIDADO !!!!!!!!!!!!! \n");
    return;    }
}
}

```

```

void ental()
{
float hid[18],hcom[9][18],Hcom[9][18];
/* Cálculo das entalpias necessárias para balanço de energia -
   Entalpias do líquido e do vapor nos pratos */

hid[j] = 0;
H[j] = 0;
for (i=0;i<c;i++) /* cp em cal/mol.K - To em K */
{ hcom[i][j] = cpl[i] * (To - T[j]);
  /* ent.mol.de i puro liq. a temp.T do pr. j (cal/mol) */
  Hcom[i][j] = Hvap[i] + cpv[i] * (T[j] - To);
  /* ent.mol.de i puro gas. a temp.T do pr. j (cal/mol) */
  hid[j] = hid[j] + (x[i][j] * hcom[i][j]);
  /* ent.mol.de solucao liq.ideal no pr. j (cal/mol) */
  H[j] = H[j] + (y[i][j] * Hcom[i][j]);
  /* ent.total molar da fase gasosa no pr.j (cal/mol) */
}
hant[j] = h[j];
h[j] = hid[j] + he[j]; /* ent. excesso he calculada em uniq() */
/* entalpia total molar da fase liq. no prato j (cal/mol) */
}

```

```

void entvap()
{
float Hvpc[9],Hvpr[9];
/* Cálculo das entalpias de vaporização para referv. e condens. */

Hvapc = 0;
Hvapc = 0;
T[0] = T[1]; /* temp. no cond. = temp. prato 1 */
for (i=0;i<c;i++)
{ Hvpc[i] = Hvpteb[i] * pow(((1 - T[0]/Tc[i])/(1 - Teb[i]/Tc[i])),0.38);
  /* cal.lat.mol.vap.de i puro a temp.do condensador */
  Hvpr[i] = Hvpteb[i] * pow(((1 - T[n+1]/Tc[i])/(1 - Teb[i]/Tc[i])),0.38);
  /* cal.lat.mol.vap.de i puro a temp. do refervedor */
  Hvapc = Hvapc + (y[i][1] * Hvpc[i]);
  /* cal.lat.mol.vap.da solucao liq. no cond. */
}

```

```

    Hvapr = Hvapr + (x[i][n+1] * Hvpr[i]);
        /* cal.lat.mol.vap.da solucao liq. no referv. */
    }
}

void orvalho()
{
/* Caálculo da temperatura de orvalho e da razão de equilíbrio p/ estimativa
do perfil inicial de temperatura e composição da coluna */

float T1=0,S,Sant,SS,fun1,fun2,dfdt;
int orv;

/* Estimativa inicial */
for (i=0;i<c;i++)
    { T1 = T1 + ((Teb[i]+10.0) * y[i][j]);
    }
T[j] = T1;

vir:
it = 2;
orv = 1;
viri();
fugpad();

quac:
if (orv==1)
    { Sant = 0.0;
    for (i=0;i<c;i++) gam[i] = 1.0; /* fase líq. ideal na prim. iteração */
    }
else { uniq();
/* Sant = S; */
}
S = 0.0;
for (i=0;i<c;i++)
    { K[i] = (gam[i] * fug0[i])/(fi[i] * P);
    x[i][j] = y[i][j]/K[i];
    S = S + x[i][j];
    }
if (fabs(Sant - S) > 0.001) /* os somatórios de x ainda nao se tornaram ctes. */
    { for (i=0;i<c;i++)
        { x[i][j] = x[i][j]/S; /* normalização dos x */
        }
    Sant = S;
    orv = orv + 1;
    goto quac;
}

```

```

}
fun1 = log(S);
if (fabs(fun1) > 0.001) /* somatório de x ainda não é = 1 */
{ T1 = T1 + 1.0; /* correção da temp. por Newton-Raphson */
  T[j] = T1;
  uniq();
  viri();
  fugpad();
  SS = 0;
  for (i=0;i<c;i++)
    { K[i] = (gam[i] * fug0[i])/(fi[i] * P);
      x[i][j] = y[i][j]/K[i];
      SS = SS + x[i][j];
    }
  fun2 = log(SS);
  T1 = T1 - 1.0;

  dfdt = (fun2 - fun1)/1.0; /* aprox. por dif. finitas */

  T[j] = T1 - (fun1/dfdt);
  T1 = T[j];
  goto vir;
}
}

```

```
void bolha()
```

```

{
/* Cálculo da temperatura de cada estágio
- Método iterativo de Newton-Raphson - */
/* Cálculo da composição de equilíbrio do vapor */

```

```
float S,SS,Sant,T1=0,fun1,fun2,dfdt,yeq[9][18];
```

```
/* Estimativa inicial */
```

```

for (i=0;i<c;i++)
  { T1 = T1 + (x[i][j] * Teb[i]); /* T EM KELVIN !!!!! */
  } /* chute bom para pressões próximas a atmosférica */
T[j] = T1;

```

```
uniquac:
```

```

it = 1;
uniq();

```

```
virial:
```

```

if (it==1)
  { Sant = 0.0;

```

```

    viri(); /* não serão calculados os fi, mas sim os Bik */
    fugpad();
    for (i=0; i<c; i++) fi[i] = 1.0; /* fase gasosa ideal na primeira iteração */
}
else { viri();
      fugpad();
      /* Sant = S; */
}
S = 0.0;
for (i=0; i<c; i++)
  { K[i] = (gam[i] * fug0[i]) / (fi[i] * P);
    y[i][j] = K[i] * x[i][j];
    S = S + y[i][j];
  }
if (fabs(Sant - S) > 0.001) /* os somatórios de y ainda não se tornaram ctes. */
  { for (i=0; i<c; i++)
    { y[i][j] = y[i][j] / S; /* normalização dos y */
    }
    Sant = S;
    it = it + 1;
    goto virial;
  }

fun1 = log(S);
if (fabs(fun1) > 0.001) /* somatório de y ainda não é = 1 */
  { T1 = T1 + 1.0; /* correção da temperatura por Newton-Raphson */
    T[j] = T1;
    uniq();
    viri(); /* calcula fi considerando a mesma composição da temp. anterior */
    fugpad();
    SS = 0;
    for (i=0; i<c; i++)
      { K[i] = (gam[i] * fug0[i]) / (fi[i] * P);
        y[i][j] = K[i] * x[i][j];
        SS = SS + y[i][j];
      }
    fun2 = log(SS);
    T1 = T1 - 1.0;

    dfdt = (fun2 - fun1) / 1.0; /* aproximação por dif. finitas */

    T[j] = T1 - (fun1 / dfdt);
    T1 = T[j];
    goto uniuac;
  }

for (i=0; i<c; i++)

```

```

    { yeq[i][j] = y[i][j];
      if (j!=(n+1) && bo!=1) { /* refervedor é estagio ideal e bo=1 também */
        y[i][j] = E/100.0 * (yeq[i][j] - y[i][j+1]) + y[i][j+1];
      }
    }
  }

void viri()
{
  /* Cálculo do coeficiente de fugacidade fi[i] pela Equação Virial; */

  float mdpr[9], Tr, term, Tcm, Pcm, omegam, f0, f1, f2, Bm, som;
  int k;

  /* Estimativa dos seg. coef. vir. puros e cruzados da mistura gasosa pelas
  correlações de Tsonopoulos */

  for (i=0; i<c; i++)
  { if (polar[i]==1) /* gás polar */
    { if (PH[i]==0)
      { /* não forma ponte de hidrogênio */
        mdpr[i] = (986.4 * pow(mdp[i],2.0) * Pc[i])/pow(Tc[i],2.0);
        /* momento dipolar reduzido - Pc em kPa - Tc em K */
        at[i] = (-2.14e-4 * mdpr[i]) - (4.308e-21 * pow(mdpr[i],8.0));
        bt[i] = 0.0;
      }
      /* Se for polar e formar pte. de H, at e bt devem ter sido lidos no início */
    }
    /* Se não for polar, estes cálculos não são necessários */
  }
  for (i=0; i<c; i++)
  { for (k=0; k<c; k++) /* k é componente */
    { if (i==k) Tr = T[j]/Tc[i];
      else { term = pow(Vc[i],(1./3.)) + pow(Vc[k],(1./3.));
            Tcm = sqrt(Tc[i]*Tc[k]) * (Vc[i] * Vc[k])/pow((0.5*term),6.0);
            Pcm = 4.0*Tcm * (Pc[i]*Vc[i]/Tc[i] + Pc[k]*Vc[k]/Tc[k])/pow(term,3.0);
            /* Volume crítico em m3/kmol */
            omegam = 0.5 * (omega[i] * omega[k]);
            Tr = T[j]/Tcm;
          }
      f0 = 0.1445 - 0.33/Tr - 0.1385/pow(Tr,2.0) - 0.0121/pow(Tr,3.0) - 6.07e-
4/pow(Tr,8.0);
      f1 = 0.0637 + 0.331/pow(Tr,2.0) - 0.423/pow(Tr,3.0) - 0.008/pow(Tr,8.0);

      if (polar[i]==1)
        { if (i==k) f2 = at[i]/pow(Tr,6.0) - bt[i]/pow(Tr,8.0);

```

```

        else if (polar[k]==1) /* componentes diferentes mas ambos polares */
            { f2 = 0.5*(at[i]+at[k])/pow(Tr,6.0) - 0.5*(bt[i]+bt[k])/pow(Tr,8.0);
              }
        }
    else f2 = 0.0;
    if (i==k) B[i][i] = (8.31439*Tc[i]/Pc[i]) * (f0 + omega[i]*f1 + f2);
    else B[i][k] = (8.31439*Tcm/Pcm) * (f0 + omegam*f1 + f2);
        B[k][i] = B[i][k];
    }
}
if (it!=1)
{ Bm = 0.0;
  for (i=0;i<c;i++)
    { for (k=0;k<c;k++) /* Segundo coeficiente virial da mistura gasosa */
      { Bm = Bm + (y[i][j] * y[k][j]) * B[i][k];
        } /* Bm em m3/kmol */
    }
  for (i=0;i<c;i++)
    { som = 0;
      for (k=0;k<c;k++)
        { som = som + y[k][j]*B[i][k];
          }
      fi[i] = P * ((2.0 * som) - Bm)/(8.31439 * T[j]);
      fi[i] = exp(fi[i]); /* coef. de fugacidade do comp. i */
    }
}
}
}

```

void fugpad()

```

{
float Pv[9],fisat[9];
/* Cálculo da fugacidade no estado padrão fug0[i] */

/* Cálculo das pressões de saturação pela eq. empírica de Antoine */
for (i=0;i<c;i++)
    { T[j] = T[j] - 273.15; /* Pv em mmHg - T em Celsius - CUIDADO */
      Pv[i] = Aanto[i] - (Banto[i]/(T[j] + Canto[i]));
        /* Pv em mmHg - T em K - Cuidado com as ctes.!!!! */
    } /* Pv[i] = exp(Pv[i]);
    CUIDADO LOGARITMO NEBERIANO */
    Pv[i] = pow(10.0,Pv[i]); /* CUIDADO LOG NA BASE 10 */
    T[j] = T[j] + 273.15; /* volta T[j] para Kelvin */
    Pv[i] = Pv[i] * 0.13332237; /* Pv de mmHg para kPa */
        /* o cálculo dos coef. vir. nao muda */
    fisat[i] = (Pv[i] * B[i][i])/(8.31439 * T[j]);
    fisat[i] = exp(fisat[i]); /* Virial para comp i puro */
}

```

```

/* printf("\n B[%d][%d] = %f, fisat[%d] = %f, T[%d]=%f",i,i,B[i][i],i,fisat[i],j,T[j]); */
  fug0[i] = fisat[i] * Pv[i]; /* fugac. do liq. no est. padrão */
}
}

```

```
void uniq0
```

```
{
/* Cálculo do coeficiente de atividade pelo modelo de UNIQUAC
com parâmetros UNIFAC */

```

```
float som,somqx,somrx,somki,somsk,somdiv,somln,somtt,soma;
float gamC[9],gamR[9],l[9],theta[9],unfi[9],tal[9][9];
int k,s;
```

```
/* Contribuição combinatorial (gamC e na verdade ln gamC) */
```

```
somqx = 0.0;
somrx = 0.0;
for (k=0;k<c;k++)
{ somqx = somqx + (unq[k] * x[k][j]);
  somrx = somrx + (unr[k] * x[k][j]);
  l[k] = 5.0 * (unr[k] - unq[k]) - (unr[k] - 1.0);
  for (s=0;s<c;s++)
    { tal[k][s] = exp(-una[k][s]/(T[j]*1.98721));
      } /* una[i][k] em cal/mol, R=1.98721 cal/mol.K */
}

```

```
som = 0.0;
```

```
for (k=0;k<c;k++)
```

```
{ som = som + (x[k][j] * l[k]);
}

```

```
for (i=0;i<c;i++)
```

```
{ theta[i] = (unq[i] * x[i][j])/somqx;
  unfi[i] = (unr[i] * x[i][j])/somrx;
  gamC[i] = log(unfi[i]/x[i][j]) + 5.0*unq[i]*log(theta[i]/unfi[i]) + l[i] -
(unfi[i]/x[i][j])*som;
}

```

```
/* Contribuição residual (gamR e na verdade ln gamR) */
```

```
for (i=0;i<c;i++)
```

```
{ somki = 0;
  somdiv = 0;
  for (k=0;k<c;k++)
    { somki = somki + (theta[k] * tal[k][i]);
      somsk = 0;
      for (s=0;s<c;s++)
        { somsk = somsk + (theta[s] * tal[s][k]);

```

```

    }
    somdiv = somdiv + (theta[k]*tal[i][k])/somsk;
    }
    gamR[i] = unq[i] * (1.0 - log(somki) - somdiv);
}

for (i=0;i<c;i++) /* Coef. de ativ. do comp. i no prato j */
{ gam[i] = exp(gamC[i]+gamR[i]);
}

/* Cálculo da entalpia de excesso do prato j */
soma = 0;
for (i=0;i<c;i++)
{ somln = 0;
  somtt = 0;
  for (k=0;k<c;k++)
  { somln = somln + (theta[k]*tal[k][i]*log(tal[k][i]));
    somtt = somtt + (theta[k] * tal[k][i]);
  }
  soma = soma + unq[i]*x[i][j]*(somln/somtt);
}
he[j] = -1.9872 * T[j] * soma;
/* entalpia em cal/mol, temp em K, R em cal/mol.K */
}

```

**ANEXO 2 - OBTENÇÃO DO MODELO DISCRETO
DO PROCESSO VIA TRANSFORMADA-Z.**

Sendo a função transferência de pulso dada pela equação (3.6) do capítulo 3, efetuando-se as multiplicações e rearranjando-se a mesma, pode-se obter:

$$G_H G_p(z) = Z \left\{ \frac{K_p}{\tau_p} \left[\frac{e^{-st_d}}{s(s+1/\tau_p)} - \frac{e^{-sT} \cdot e^{-st_d}}{s(s+1/\tau_p)} \right] \right\} \quad (A2.1)$$

Devido à linearidade da operação de transformação-z, há uma propriedade que afirma que a transformada de uma soma é igual à soma das transformadas e portanto a equação (A2.1) se torna:

$$G_H G_p(z) = \frac{K_p}{\tau_p} \left[Z \left\{ \frac{e^{-t_d s}}{s(s+1/\tau_p)} \right\} - Z \left\{ \frac{e^{-Ts} \cdot e^{-t_d s}}{s(s+1/\tau_p)} \right\} \right] \quad (A2.2)$$

Da própria definição da transformada-z, tem-se que:

$$e^{-sT} = z^{-1} \quad (A2.3)$$

Aplicando-se esta definição à equação (A2.2) e rearranjando, obtém-se:

$$G_H G_p(z) = \frac{K_p}{\tau_p} (1 - z^{-1}) Z \left\{ \frac{e^{-t_d s}}{s(s+1/\tau_p)} \right\} \quad (A2.4)$$

Sabendo-se, da implementação do PAC, que o tempo morto associado ao processo é bastante inferior ao valor de amostragem a ser utilizado, este sistema deve ser manuseado por uma extensão das transformadas-z conhecida por transformada-z modificada, onde:

$$Z_m \{G(s)\} = Z \{G(s) \cdot e^{-t_d s}\} \quad (A2.5)$$

Modificando-se a equação (A2.4) através do conceito da equação (A2.5) para posterior resolução através da transformada-z modificada, tem-se:

$$G_H G_p(z, m) = \frac{K_p}{\tau_p} (1 - z^{-1}) Z_m \left\{ \frac{1}{s(s+1/\tau_p)} \right\} \quad (A2.6)$$

Considerando-se $t_d = T - mT$, ou rearranjando:

$$m = \frac{T - t_d}{T} \quad (A2.7)$$

sendo m uma variável auxiliar utilizada na transformada-z modificada, e aplicando-se esta transformada através de tabelas disponíveis em literatura, obtém-se de (A2.6) e (A2.7):

$$G_H G_p(z, m) = \frac{K_p}{\tau_p} (1 - z^{-1}) \tau_p \left[\frac{1}{z-1} - \frac{e^{-(T-t_d)/\tau_p}}{z - e^{-T/\tau_p}} \right] \quad (A2.8)$$

ou, dividindo-se o numerador e o denominador por z :

$$G_H G_p(z, m) = K_p (1 - z^{-1}) z^{-1} \left[\frac{1}{1 - z^{-1}} - \frac{e^{-(T-t_d)/\tau_p}}{1 - z^{-1} e^{-T/\tau_p}} \right] \quad (A2.9)$$

Rearranjando-se a equação (A2.9), obtém-se:

$$\frac{\hat{y}(z, m)}{\hat{u}(z, m)} = G_H G_p(z, m) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1}} \quad (A2.10)$$

onde $b_1 = K_p [1 - e^{-(T-t_d)/\tau_p}]$ (A2.11a)

$b_2 = K_p [e^{-(T-t_d)/\tau_p} - e^{-T/\tau_p}]$ (A2.11b)

$a_1 = e^{-T/\tau_p}$ (A2.11c)

Portanto, para a obtenção do modelo discreto entre a variável controlada (y) e a manipulada (u), efetuam-se as multiplicações na equação (A2.10), de modo a obter:

$$\hat{y}(z, m)(1 - a_1 z^{-1}) = \hat{u}(z, m)(b_1 z^{-1} + b_2 z^{-2}) \quad (\text{A2.12})$$

Isolando-se a variável controlada do instante atual, tem-se:

$$\hat{y}(z, m) = a_1 z^{-1} \hat{y}(z, m) + b_1 z^{-1} \hat{u}(z, m) + b_2 z^{-2} \hat{u}(z, m) \quad (\text{A2.13})$$

Aplicando-se a transformada inversa Z_m^{-1} , obtém-se finalmente o modelo discreto do processo:

$$y_i = a_1 y_{i-1} + b_1 u_{i-1} + b_2 u_{i-2} \quad (\text{A2.14})$$

com a_1 , b_1 e b_2 dados pelo conjunto de equações (A2.11).

**ANEXO 3 - OBTENÇÃO ANALÍTICA DO GRADIENTE
DA FUNÇÃO-OBJETIVO.**

Substituindo-se as equações (3.7) e (3.8) na equação (3.9) do capítulo 3, obtém-se a seguinte função-objeto:

$$P = \frac{1}{I} \sum_{i=1}^I \left[\tilde{y}_i - e^{-T/\tau_p} \tilde{y}_{i-1} - K_p \left(1 - e^{-(T-t_d)/\tau_p} \right) \tilde{c}_{i-1} - K_p \left(e^{-(T-t_d)/\tau_p} - e^{-T/\tau_p} \right) \tilde{c}_{i-2} \right]^2 \quad (\text{A3.1})$$

onde o superscrito \sim indica valores amostrados e o subscrito i representa o instante de amostragem.

Sendo condição necessária do método escolhido para otimização da função objetivo, igualar a zero o gradiente da mesma, obtém-se analiticamente as seguintes derivadas parciais de P em relação aos seus parâmetros desconhecidos:

$$f_1 = \frac{\partial P}{\partial K_p} = \sum_{i=1}^I \left\{ \left[\tilde{y}_i - e^{-T/\tau_p} \tilde{y}_{i-1} - K_p \left(1 - e^{-(T-t_d)/\tau_p} \right) \tilde{c}_{i-1} - K_p \left(e^{-(T-t_d)/\tau_p} - e^{-T/\tau_p} \right) \tilde{c}_{i-2} \right] \cdot \left[- \left(1 - e^{-(T-t_d)/\tau_p} \right) \tilde{c}_{i-1} - \left(e^{-(T-t_d)/\tau_p} - e^{-T/\tau_p} \right) \tilde{c}_{i-2} \right] \right\} = 0 \quad (\text{A3.2})$$

$$f_2 = \frac{\partial P}{\partial \tau_p} = \sum_{i=1}^I \left\{ \left[\tilde{y}_i - e^{-T/\tau_p} \tilde{y}_{i-1} - K_p \left(1 - e^{-(T-t_d)/\tau_p} \right) \tilde{c}_{i-1} - K_p \left(e^{-(T-t_d)/\tau_p} - e^{-T/\tau_p} \right) \tilde{c}_{i-2} \right] \cdot \left[\frac{-T}{\tau_p^2} e^{-T/\tau_p} \tilde{y}_{i-1} + K_p \frac{(T-t_d)}{\tau_p^2} e^{-(T-t_d)/\tau_p} (\tilde{c}_{i-1} - \tilde{c}_{i-2}) + K_p \frac{T}{\tau_p^2} e^{-T/\tau_p} \tilde{c}_{i-2} \right] \right\} = 0 \quad (\text{A3.3})$$

$$f_3 = \frac{\partial P}{\partial t_d} = \sum_{i=1}^I \left\{ \left[\tilde{y}_i - e^{-T/\tau_p} \tilde{y}_{i-1} - K_p \left(1 - e^{-(T-t_d)/\tau_p} \right) \tilde{c}_{i-1} - K_p \left(e^{-(T-t_d)/\tau_p} - e^{-T/\tau_p} \right) \tilde{c}_{i-2} \right] \right. \\ \left. \left[\frac{K_p}{\tau_p} e^{-(T-t_d)/\tau_p} (\tilde{c}_{i-1} - \tilde{c}_{i-2}) \right] \right\} = 0 \quad (A3.4)$$

Efetuada-se as multiplicações e agrupando-se os termos de amostragens, obtém-se de (A3.2), (A3.3) e (A3.4), respectivamente:

$$f_1 = \sum_{i=1}^I [Ae^{-(T-t_d)/\tau_p} - J + Ee^{-T/\tau_p} - Be^{-(2T-t_d)/\tau_p} - Fe^{-2T/\tau_p} + 2DK_p e^{-(T-t_d)/\tau_p} + HK_p \\ - 2DK_p e^{-2(T-t_d)/\tau_p} + 2DK_p e^{-(2T-t_d)/\tau_p} + HK_p e^{-2T/\tau_p} - 2GK_p e^{-T/\tau_p}] = 0 \quad (A3.5)$$

$$f_2 = \sum_{i=1}^I \left[-M \frac{T}{\tau_p^2} e^{-T/\tau_p} + AK_p \frac{(T-t_d)}{\tau_p^2} e^{-(T-t_d)/\tau_p} + EK_p \frac{T}{\tau_p^2} e^{-T/\tau_p} + \right. \\ \left. + B \frac{K_p}{\tau_p^2} e^{-(2T-t_d)/\tau_p} (t_d - 2T) + N \frac{T}{\tau_p^2} e^{-2T/\tau_p} - 2FK_p \frac{T}{\tau_p^2} e^{-2T/\tau_p} + \right. \\ \left. + LK_p^2 \frac{(T-t_d)}{\tau_p^2} e^{-(T-t_d)/\tau_p} + (L+D)K_p^2 \frac{(t_d-T)}{\tau_p^2} e^{-2(T-t_d)/\tau_p} - \right. \\ \left. - GK_p^2 \frac{T}{\tau_p^2} e^{-T/\tau_p} + HK_p^2 \frac{T}{\tau_p^2} e^{-2T/\tau_p} + DK_p^2 \frac{(2T-t_d)}{\tau_p^2} e^{-(2T-t_d)/\tau_p} \right] = 0 \quad (A3.6)$$

$$f_3 = \sum_{i=1}^I \left[A \frac{K_p}{\tau_p} e^{-(T-t_d)/\tau_p} - B \frac{K_p}{\tau_p} e^{-(2T-t_d)/\tau_p} + L \frac{K_p^2}{\tau_p} e^{-(T-t_d)/\tau_p} + \right. \\ \left. + C^2 \frac{K_p^2}{\tau_p} e^{-2(T-t_d)/\tau_p} + D \frac{K_p^2}{\tau_p} e^{-(2T-t_d)/\tau_p} \right] = 0 \quad (A3.7)$$

sendo:

$$A = (\tilde{c}_{i-1} - \tilde{c}_{i-2})\tilde{y}_i$$

$$B = (\tilde{c}_{i-1} - \tilde{c}_{i-2})\tilde{y}_{i-1}$$

$$C = \tilde{c}_{i-1} - \tilde{c}_{i-2}$$

$$D = (\tilde{c}_{i-1} - \tilde{c}_{i-2})\tilde{c}_{i-2}$$

$$E = \tilde{c}_{i-2}\tilde{y}_i + \tilde{c}_{i-1}\tilde{y}_{i-1}$$

$$F = \tilde{c}_{i-2}\tilde{y}_{i-1}$$

$$G = \tilde{c}_{i-1}\tilde{c}_{i-2}$$

$$H = \tilde{c}_{i-2}^2$$

$$J = \tilde{c}_{i-1}\tilde{y}_i$$

$$L = (\tilde{c}_{i-2} - \tilde{c}_{i-1})\tilde{c}_{i-1}$$

$$M = \tilde{y}_{i-1}\tilde{y}_i$$

$$N = \tilde{y}_{i-1}^2$$

**ANEXO 4 - OBTENÇÃO ANALÍTICA DO JACOBIANO
DA FUNÇÃO-OBJETIVO.**

Para que se possa utilizar o Método de Newton-Raphson para localização das raízes (k_p , τ_p e t_d) no conjunto de equações algébricas não-lineares formadas pelas equações (A3.5), (A3.6) e (A3.7), obtidas no anexo 3 a partir das equações (3.7), (3.8), (3.9) e (3.10) do capítulo 3, é necessário que se encontrem as derivadas parciais de cada uma destas equações algébricas em relação aos mesmos parâmetros desconhecidos.

Assim sendo, derivando-se analiticamente a equação (A3.5) do anexo 3, em relação a K_p , τ_p e t_d , obtém-se respectivamente:

$$\frac{\partial f_1}{\partial K_p} = \sum_{i=1}^I [2De^{-(T-t_d)/\tau_p} + H - 2De^{-2(T-t_d)/\tau_p} + 2De^{-(2T-t_d)/\tau_p} + He^{-2T/\tau_p} - 2Ge^{-T/\tau_p}] \quad (A4.1)$$

$$\begin{aligned} \frac{\partial f_1}{\partial \tau_p} = \sum_{i=1}^I [& A \frac{(T-t_d)}{\tau_p^2} e^{-(T-t_d)/\tau_p} + E \frac{T}{\tau_p^2} e^{-T/\tau_p} - B \frac{(2T-t_d)}{\tau_p^2} e^{-(2T-t_d)/\tau_p} - F \frac{2T}{\tau_p^2} e^{-2T/\tau_p} + \\ & + 2DK_p \frac{(T-t_d)}{\tau_p^2} e^{-(T-t_d)/\tau_p} - 4DK_p \frac{(T-t_d)}{\tau_p^2} e^{-2(T-t_d)/\tau_p} + 2DK_p \frac{(2T-t_d)}{\tau_p^2} e^{-(2T-t_d)/\tau_p} + \\ & + HK_p \frac{2T}{\tau_p^2} e^{-2T/\tau_p} - 2GK_p \frac{T}{\tau_p^2} e^{-T/\tau_p}] \end{aligned} \quad (A4.2)$$

$$\begin{aligned} \frac{\partial f_1}{\partial t_d} = \sum_{i=1}^I [& \frac{A}{\tau_p} e^{-(T-t_d)/\tau_p} - \frac{B}{\tau_p} e^{-(2T-t_d)/\tau_p} + \frac{2DK_p}{\tau_p} e^{-(T-t_d)/\tau_p} - \\ & - \frac{4DK_p}{\tau_p} e^{-2(T-t_d)/\tau_p} + \frac{2DK_p}{\tau_p} e^{-(2T-t_d)/\tau_p}] \end{aligned} \quad (A4.3)$$

Da equação (A3.6), tem-se da mesma forma:

$$\begin{aligned} \frac{\partial f_2}{\partial K_p} = \sum_{i=1}^I & \left[A \frac{(T-t_d)}{\tau_p^2} e^{-(T-t_d)/\tau_p} + E \frac{T}{\tau_p^2} e^{-T/\tau_p} + \frac{B}{\tau_p^2} e^{-(2T-t_d)/\tau_p} (t_d - 2T) - 2F \frac{T}{\tau_p^2} e^{-2T/\tau_p} + \right. \\ & + 2LK_p \frac{(T-t_d)}{\tau_p^2} e^{-(T-t_d)/\tau_p} + 2(L+D)K_p \frac{(T-t_d)}{\tau_p^2} e^{-2(T-t_d)/\tau_p} - 2GK_p \frac{T}{\tau_p^2} e^{-T/\tau_p} + \\ & \left. + 2HK_p \frac{T}{\tau_p^2} e^{-2T/\tau_p} + 2DK_p \frac{(2T-t_d)}{\tau_p^2} e^{-(2T-t_d)/\tau_p} \right] \end{aligned} \quad (A4.4)$$

$$\begin{aligned} \frac{\partial f_2}{\partial \tau_p} = \sum_{i=1}^I & \left[(EK_p T - MT - GK_p^2 T) e^{-T/\tau_p} \left(\frac{T}{\tau_p^4} - \frac{2}{\tau_p^3} \right) + \right. \\ & + (BK_p - DK_p^2) (t_d - 2T) e^{-(2T-t_d)/\tau_p} \left[\frac{(2T-t_d)}{\tau_p^4} - \frac{2}{\tau_p^3} \right] + \\ & + (AK_p + LK_p^2) (T-t_d) e^{-(T-t_d)/\tau_p} \left[\frac{(T-t_d)}{\tau_p^4} - \frac{2}{\tau_p^3} \right] + \\ & + (NT - 2FK_p T + HK_p^2 T) e^{-2T/\tau_p} \left(\frac{2T}{\tau_p^4} - \frac{2}{\tau_p^3} \right) + \\ & \left. + (L+D)K_p^2 (t_d - T) e^{-2(T-t_d)/\tau_p} \left[\frac{2(T-t_d)}{\tau_p^4} - \frac{2}{\tau_p^3} \right] \right] \end{aligned} \quad (A4.5)$$

$$\begin{aligned} \frac{\partial f_2}{\partial t_d} = \sum_{i=1}^I & \left[\left(A \frac{K_p}{\tau_p^2} + L \frac{K_p^2}{\tau_p^2} \right) e^{-(T-t_d)/\tau_p} \left[\frac{(T-t_d)}{\tau_p} - 1 \right] + \right. \\ & + \left(D \frac{K_p^2}{\tau_p^2} - B \frac{K_p}{\tau_p} \right) e^{-(2T-t_d)/\tau_p} \left[\frac{(2T-t_d)}{\tau_p} - 1 \right] + \\ & \left. + (L+D) \frac{K_p^2}{\tau_p^2} e^{-2(T-t_d)/\tau_p} \left[\frac{2(t_d - T)}{\tau_p} + 1 \right] \right] \end{aligned} \quad (A4.6)$$

Finalmente, da equação (A3.7), obtém-se:

$$\begin{aligned} \frac{\partial f_3}{\partial K_p} = \sum_{i=1}^I & \left[\frac{A}{\tau_p} e^{-(T-t_d)/\tau_p} - \frac{B}{\tau_p} e^{-(2T-t_d)/\tau_p} + \frac{2LK_p}{\tau_p} e^{-(T-t_d)/\tau_p} + \right. \\ & \left. + \frac{2C^2 K_p}{\tau_p} e^{-2(T-t_d)/\tau_p} + \frac{2DK_p}{\tau_p} e^{-(2T-t_d)/\tau_p} \right] \end{aligned} \quad (A4.7)$$

$$\begin{aligned} \frac{\partial f_3}{\partial \tau_p} = \sum_{i=1}^I & \left[(AK_p + LK_p^2) e^{-(T-t_d)/\tau_p} \left[\frac{(T-t_d)}{\tau_p^3} - \frac{1}{\tau_p^2} \right] + \right. \\ & \left. + (DK_p^2 - BK_p) e^{-(2T-t_d)/\tau_p} \left[\frac{(2T-t_d)}{\tau_p^3} - \frac{1}{\tau_p^2} \right] + \right. \\ & \left. + C^2 K_p^2 e^{-2(T-t_d)/\tau_p} \left[2 \frac{(T-t_d)}{\tau_p^3} - \frac{1}{\tau_p^2} \right] \right] \end{aligned} \quad (A4.8)$$

$$\begin{aligned} \frac{\partial f_3}{\partial t_d} = \sum_{i=1}^I & \left[A \frac{K_p}{\tau_p^2} e^{-(T-t_d)/\tau_p} - B \frac{K_p}{\tau_p^2} e^{-(2T-t_d)/\tau_p} + L \frac{K_p^2}{\tau_p^2} e^{-(T-t_d)/\tau_p} + \right. \\ & \left. + 2C^2 \frac{K_p^2}{\tau_p^2} e^{-2(T-t_d)/\tau_p} + D \frac{K_p^2}{\tau_p^2} e^{-(2T-t_d)/\tau_p} \right] \end{aligned} \quad (A4.9)$$

sendo: $A = (\tilde{c}_{i-1} - \tilde{c}_{i-2}) \tilde{y}_i$

$$B = (\tilde{c}_{i-1} - \tilde{c}_{i-2})\tilde{y}_{i-1}$$

$$C = \tilde{c}_{i-1} - \tilde{c}_{i-2}$$

$$D = (\tilde{c}_{i-1} - \tilde{c}_{i-2})\tilde{c}_{i-2}$$

$$E = \tilde{c}_{i-2}\tilde{y}_i + \tilde{c}_{i-1}\tilde{y}_{i-1}$$

$$F = \tilde{c}_{i-2}\tilde{y}_{i-1}$$

$$G = \tilde{c}_{i-1}\tilde{c}_{i-2}$$

$$H = \tilde{c}_{i-2}^2$$

$$J = \tilde{c}_{i-1}\tilde{y}_i$$

$$L = (\tilde{c}_{i-2} - \tilde{c}_{i-1})\tilde{c}_{i-1}$$

$$M = \tilde{y}_{i-1}\tilde{y}_i$$

$$N = \tilde{y}_{i-1}^2$$

**ANEXO 5 - IMPLEMENTAÇÃO DA IDENTIFICAÇÃO DE
PROCESSOS POR ANÁLISE REGRESSIVA.**

```

void NEWT(TA,COEF,X,cna,ced)

float TA, COEF[13][23], X[10][5];
int *cna,*ced;

/* TA = intervalo de amostragem
   COEF = coeficientes obtidos de amostragens (anexo B)
   X = raízes das equações B.5, B.6 e B.7 do anexo B (Kp, Tp, td)
   *ced = ponteiro que indica o componente atualmente controlado
   *cna = número de amostragens para a análise regressiva */

{
  /* Método de Newton-Raphson para resolução do problema dos mínimos-quadrados
    de estimação on-line dos parâmetros do processo */

int NTRIAL=10000,ii,kk,jj,N=3,com=0;
  /* N=3 porque são 3 raízes procuradas (Kp,Tp,td) e 3 equações */
float errf,errx,TOLX=1.e-6,TOLF=1.e-6;
float ALPHA[5][5],BETA[5],INDX[5],X0[5],ALPHAVE[5][5],dX[5];

/* BETA = {f1, f2, f3}

      df1/dKp df1/dTp df1/dtd
ALPHA = df2/dKp df2/dTp df2/dtd
      df3/dKp df3/dTp df3/dtd */

for (jj=1;jj<=N;jj++)
  { dX[jj] = 0.1 * X[*ced][jj];
  }

for (jj=1;jj<=N;jj++)
  { X0[jj] = X[*ced][jj];
  }
comeco:
com = com + 1;

if (com==2) /* estimativas iniciais diferentes usadas quando a primeira não levar à convergência */
  { for (jj=1;jj<=N;jj++)
    { X[*ced][jj] = X0[jj] * 1.15;
    }
  }

if (com==3)
  { for (jj=1;jj<=N;jj++)
    { X[*ced][jj] = X0[jj] * 0.85;
    }
  }

if (com==4)
  { for (jj=1;jj<=N;jj++)
    { X[*ced][jj] = X0[jj] * 1.30;
    }
  }

if (com==5)
  { for (jj=1;jj<=N;jj++)
    { X[*ced][jj] = X0[jj] * 0.70;
    }
  }
}

```

```

if (com==6)
  { for (jj=1;jj<=N;jj++)
    { X[*ced][jj] = X0[jj] * 1.50;
    }
  }
if (com==7)
  { for (jj=1;jj<=N;jj++)
    { X[*ced][jj] = X0[jj] * 0.50;
    }
  }
for (kk=1;kk<=NTRIAL;kk++)
  {USRFUN1(X,ALPHA,BETA,N,cna,COEF,TA,ced);
  /* subrotina para montagem do sistema de equações cujas raízes deverão ser determinadas */

  errf=1.e-20;
  for (jj=1;jj<=N;jj++) errf = errf + (fabs(BETA[jj])/1000.0);
  /* BETA havia sido multiplicado por 1000 */

  if (errf<=TOLF)
    { goto fim; /* As raízes estão corretas */
    }
  for (ii=1;ii<=N;ii++)
    { for (jj=1;jj<=N;jj++)
      { ALPHAVE[ii][jj] = ALPHA[ii][jj];
      }
    }

  LUDCMP(ALPHA,INDX,N);
  /* decomposição LU para encontrar incrementos de Newton-Raphson (DX) */

  MPROVE(ALPHAVE,ALPHA,N,INDX,BETA,dX);
  /* junto com LUBKSB resolve sistema linear de equações a partir de matriz decomposta por LUDCMP */

  errx=1.e-20;
  for (jj=1;jj<=N;jj++)
    { errx = errx + fabs(dX[jj]);
      X[*ced][jj] = X[*ced][jj] + 0.05*dX[jj]; /* Correções para o chute inicial */
    }
  if (errx<=TOLX)
    { goto fim; /* As raízes estão corretas */
    }
  if (kk==NTRIAL) printf ("\n\n NÚMERO DE TENTATIVAS INSUFICIENTE \n\n");
  }

fim:
if (X[*ced][1]<1.e-05 || X[*ced][2]<1.e-05 || X[*ced][3]<1.e-05)
  { if (com<7) goto comeco; /* volta e procura raízes em outra região */
  }
if (X[*ced][1]>10.0 || X[*ced][2]>300.0 || X[*ced][3]>10.0)
  /* limites aceitáveis para os parâmetros Kp, Tp, td */
  { if (com<7) goto comeco; /* volta e procura raízes em outra região */
  }

printf("\nNEWT com=%d Raízes corretas: Kp=%f(min/mol) Tp=%f (min) td=%f (min)",
com,X[*ced][1],X[*ced][2],X[*ced][3]);
}

```

```

void LUDCMP(ALPHA,INDX,N)

float ALPHA[5][5], *INDX;
int N;

{

/* Decomposição LU da matriz das equações algébricas lineares (eqs. 3.12),
   geradas por Newton-Raphson, para encontrar os incrementos DX */

float VV[100],AAMAX,sum,dum,TINY=1.0e-20;
int D=1,ii,jj,kk,IMAX;

for (ii=1;ii<=N;ii++)
  { AAMAX = 1.e-20;
    for (jj=1;jj<=N;jj++)
      { if (fabs(ALPHA[ii][jj])>AAMAX) AAMAX = fabs(ALPHA[ii][jj]);
        }
    if (fabs(AAMAX)<=1.e-20) printf("\n\n MATRIZ SINGULAR \n\n");
    VV[ii] = 1./AAMAX;
  }

for (jj=1;jj<=N;jj++)
  { for (ii=1;ii<=(jj-1);ii++)
    { sum = ALPHA[ii][jj];
      for (kk=1;kk<=(ii-1);kk++)
        { sum = sum - (ALPHA[ii][kk]* ALPHA[kk][jj]);
          }
      ALPHA[ii][jj] = sum;
    }
  }

AAMAX=1.e-20;

for (ii=jj;ii<=N;ii++)
  { sum = ALPHA[ii][jj];
    for (kk=1;kk<=(jj-1);kk++)
      { sum = sum - (ALPHA[ii][kk]* ALPHA[kk][jj]);
        }
    ALPHA[ii][jj] = sum;
    dum = VV[ii]*fabs(sum);
    if (dum>=AAMAX)
      { IMAX=ii;
        AAMAX=dum;
      }
  }

if (jj!=IMAX)
  { for (kk=1;kk<=N;kk++)
    { dum = ALPHA[IMAX][kk];
      ALPHA[IMAX][kk] = ALPHA[jj][kk];
      ALPHA[jj][kk] = dum;
    }
    D = -D;
    VV[IMAX] = VV[jj];
  }
}

```

```

INDX[jj] = IMAX;

if(fabs(ALPHA[jj][jj])<=1.e-20) ALPHA[jj][jj] = TINY;

if (jj!=N)
  { dum=1./ ALPHA[jj][jj];
    for (ii=(jj+1);ii<=N;ii++)
      { ALPHA[ii][jj]= ALPHA[ii][jj]*dum;
        }
    }
}
}
}

```

```
void LUBKSB(ALPHA,N,INDX,R)
```

```
float ALPHA[5][5], *INDX, *R;
int N;
```

```
/* R = BETA */
```

```
{
```

```
/* Resolução do sistema linear de eqs. 3.12, a partir de matriz decomposta em LUDCMP */
```

```
int I,ii=0,ll,jj;
float sum;
```

```
for (I=1;I<=N;I++)
  { ll= INDX[I];
    sum= R[ll];
    R[ll]= R[I];
    if (ii!=0)
      { for (jj=ii;jj<=(I-1);jj++)
          { sum = sum - (ALPHA[I][jj]* R[jj]);
            }
        }
    else { if (sum!=0) ii=I;
          }
    R[I]=sum;
  }

```

```
for (I=N;I>=1;I--)
  { sum = R[I];
    if (I<N)
      { for (jj=(I+1);jj<=N;jj++) sum = sum - ALPHA[I][jj]* R[jj];
        }
    R[I] = sum/ ALPHA[I][I];
  }
}
}

```

```

void USRFUN1(X,ALPHA,BETA,N,cna,COEF,TA,ced)

float X[10][5], ALPHA[5][5], *BETA, TA, COEF[13][23];
int N, *cna, *ced;

{
/* Subrotina para montar o sistema de equações de Newton-Raphson, onde
X={Kp, Tp, td}, BETA={f1, f2, f3} e


$$\text{ALPHA} = \begin{matrix} df_1/dKp & df_1/dTp & df_1/dtd \\ df_2/dKp & df_2/dTp & df_2/dtd \\ df_3/dKp & df_3/dTp & df_3/dtd \end{matrix} \quad */$$


int ii, jj;
float PARBETA[5], PARALPHA[5][5]; /* parcelas de BETA e ALPHA */

for (jj=1; jj<=N; jj++)
{ BETA[jj] = 1.e-20;
for (ii=1; ii<=N; ii++)
{ ALPHA[jj][ii] = 1.e-20;
}
}

for (ii=1; ii<=*cna; ii++)
{ PARBETA[1] = COEF[1][ii]*exp(-(TA- X[*ced][3])/ X[*ced][2]) - COEF[9][ii] +
COEF[5][ii]*exp(-TA/ X[*ced][2]) - COEF[2][ii]*exp(-(2.0*TA- X[*ced][3])/ X[*ced][2]) -
COEF[6][ii]*exp(-2.0*TA/ X[*ced][2]) + 2.0* COEF[4][ii]* X[*ced][1]*
exp(-(TA- X[*ced][3])/ X[*ced][2]) + COEF[8][ii]* X[*ced][1] - 2.0* COEF[4][ii]* X[*ced][1]*
exp(-2.0*(TA- X[*ced][3])/ X[*ced][2]) + 2.0* COEF[4][ii]* X[*ced][1]*
exp(-(2.0*TA- X[*ced][3])/ X[*ced][2]) + COEF[8][ii]* X[*ced][1]*exp(-2.0*TA/ X[*ced][2]) -
2.0* COEF[7][ii]* X[*ced][1]*exp(-TA/ X[*ced][2]); /* equação B.5 */

BETA[1] = BETA[1] + PARBETA[1];

PARBETA[2] = - COEF[11][ii]*(TA/pow(X[*ced][2],2.))*exp(-TA/ X[*ced][2]) +
COEF[1][ii]* X[*ced][1]*(TA- X[*ced][3])/pow(X[*ced][2],2.)*exp(-(TA- X[*ced][3])/
X[*ced][2]) + COEF[5][ii]* X[*ced][1]*TA/pow(X[*ced][2],2.)*exp(-TA/ X[*ced][2]) +
COEF[2][ii]* X[*ced][1]/pow(X[*ced][2],2.)*exp(-(2.0*TA-X[*ced][3])/X[*ced][2])
*(X[*ced][3]-2.*TA) + COEF[12][ii]*TA/pow(X[*ced][2],2.)*exp(-2.0*TA/ X[*ced][2])-2.0*
COEF[6][ii]*X[*ced][1]*TA/pow(X[*ced][2],2.)*exp(-2.0*TA/ X[*ced][2]) + COEF[10][ii]*
pow(X[*ced][1],2.)*(TA- X[*ced][3])/pow(X[*ced][2],2.)*exp(-(TA- X[*ced][3])/ X[*ced][2]) +
(COEF[10][ii] + COEF[4][ii])*pow(X[*ced][1],2.)*(X[*ced][3]-TA)/pow(X[*ced][2],2.)*
exp(-2.0*(TA- X[*ced][3])/ X[*ced][2]) - COEF[7][ii]*pow(X[*ced][1],2.0)*
TA/pow(X[*ced][2],2.)*exp(-TA/ X[*ced][2]) + COEF[8][ii]*pow(X[*ced][1],2.)*
TA/pow(X[*ced][2],2.)*exp(-2.0*TA/ X[*ced][2]) + COEF[4][ii]*pow(X[*ced][1],2.)*
(2.0*TA- X[*ced][3])/pow(X[*ced][2],2.)*exp(-(2.0*TA- X[*ced][3])/ X[*ced][2]);
/* equação B.6 */

BETA[2] = BETA[2] + PARBETA[2];

PARBETA[3] = COEF[1][ii]* X[*ced][1]/ X[*ced][2]*exp(-(TA- X[*ced][3])/ X[*ced][2]) -
COEF[2][ii]*X[*ced][1]/ X[*ced][2]*exp(-(2.0*TA- X[*ced][3])/ X[*ced][2])+
COEF[10][ii]*(pow(X[*ced][1],2.)/ X[*ced][2]*exp(-(TA- X[*ced][3])/ X[*ced][2]) +

```

```

pow(COEF[3][ii],2.)*pow( X[*ced][1],2.)/ X[*ced][2]*exp(-2.*(TA- X[*ced][3])/ X[*ced][2]) +
COEF[4][ii]*pow(X[*ced][1],2.)/ X[*ced][2]*exp(-2.0*TA- X[*ced][3])/ X[*ced][2]);
/* equação B.7 */

```

```
BETA[3] = BETA[3] + PARBETA[3];
```

```
}
```

```

for (jj=1;jj<=N;jj++)
{ BETA[jj] = - BETA[jj];
}

```

```

for (ii=1;ii<=*cna;ii++)
{ PARALPHA[1][1] = 2.0* COEF[4][ii]*exp(-(TA- X[*ced][3])/ X[*ced][2]) + COEF[8][ii] -
2.0* COEF[4][ii]*exp(-2.0*(TA- X[*ced][3])/ X[*ced][2]) + 2.0* COEF[4][ii]*
exp(-2.0*TA- X[*ced][3])/ X[*ced][2]) + COEF[8][ii]*exp(-2.0*TA/ X[*ced][2]) -
2.0* COEF[7][ii]*exp(-TA/ X[*ced][2]); /* equação C.1 */

```

```
ALPHA[1][1] = ALPHA[1][1] + PARALPHA[1][1];
```

```

PARALPHA[1][2] = COEF[1][ii]*(TA- X[*ced][3])/pow(X[*ced][2],2.)*exp(-(TA- X[*ced][3])/
X[*ced][2]) + COEF[5][ii]*TA/pow( X[*ced][2],2.)*exp(-TA/ X[*ced][2]) - COEF[2][ii]*
(2.0*TA- X[*ced][3])/pow( X[*ced][2],2.)*exp(-2.*TA- X[*ced][3])/ X[*ced][2]) -
COEF[6][ii]*2.0*TA/pow( X[*ced][2],2.0)*exp(-2.0*TA/ X[*ced][2]) + 2.* COEF[4][ii]*
X[*ced][1]*(TA- X[*ced][3])/pow(X[*ced][2],2.)*exp(-(TA- X[*ced][3])/ X[*ced][2]) - 4.0*
COEF[4][ii]*X[*ced][1]*(TA- X[*ced][3])/pow( X[*ced][2],2.)*exp(-2.0*(TA- X[*ced][3])/
X[*ced][2]) + 2.0* COEF[4][ii]* X[*ced][1]*(2.0*TA- X[*ced][3])/pow(X[*ced][2],2.)*
exp(-2.*TA- X[*ced][3])/ X[*ced][2]) + COEF[8][ii]* X[*ced][1]*2.0*TA/pow( X[*ced][2],2.)*
exp(-2.0*TA/ X[*ced][2]) - 2.0* COEF[7][ii]* X[*ced][1]*TA/pow( X[*ced][2],2.0)*
exp(-TA/ X[*ced][2]); /* equação C.2 */

```

```
ALPHA[1][2] = ALPHA[1][2] + PARALPHA[1][2];
```

```

PARALPHA[1][3] = COEF[1][ii]/ X[*ced][2]*exp(-(TA- X[*ced][3])/ X[*ced][2]) -
COEF[2][ii]/ X[*ced][2]*exp(-2.0*TA- X[*ced][3])/ X[*ced][2]) + 2.0* COEF[4][ii]*
X[*ced][1]/ X[*ced][2]*exp(-(TA- X[*ced][3])/ X[*ced][2]) - 4.0* COEF[4][ii]* X[*ced][1]/
X[*ced][2]*exp(-2.0*(TA- X[*ced][3])/ X[*ced][2]) + 2.0* COEF[4][ii]* X[*ced][1]/ X[*ced][2]*
exp(-2.0*TA- X[*ced][3])/ X[*ced][2]); /* equação C.3 */

```

```
ALPHA[1][3] = ALPHA[1][3] + PARALPHA[1][3];
```

```

PARALPHA[2][1] = COEF[1][ii]*(TA- X[*ced][3])/pow( X[*ced][2],2.)*exp(-(TA- X[*ced][3])/
X[*ced][2]) + COEF[5][ii]*TA/pow( X[*ced][2],2.)*exp(-TA/ X[*ced][2]) + COEF[2][ii]/pow
(X[*ced][2],2.)*exp(-2.0*TA- X[*ced][3])/ X[*ced][2])*( X[*ced][3]-2.0*TA) - 2.0*
COEF[6][ii]*TA/pow( X[*ced][2],2.)*exp(-2.0*TA/ X[*ced][2]) + 2.0* COEF[10][ii]*
X[*ced][1]*(TA- X[*ced][3])/pow( X[*ced][2],2.0)*exp(-(TA- X[*ced][3])/ X[*ced][2]) +
2.0*(COEF[10][ii] + COEF[4][ii])* X[*ced][1]*( X[*ced][3]-TA)/pow( X[*ced][2],2.0)*
exp(-2.0*(TA- X[*ced][3])/ X[*ced][2]) - 2.0* COEF[7][ii]* X[*ced][1]*TA/
pow(X[*ced][2],2.)*exp(-TA/ X[*ced][2]) + 2.* COEF[8][ii]* X[*ced][1]*TA/
pow( X[*ced][2],2.0)*exp(-2.*TA/ X[*ced][2]) + 2.0* COEF[4][ii]* X[*ced][1]*(2.0*TA-
X[*ced][3])/pow( X[*ced][2],2.)*exp(-2.*TA- X[*ced][3])/ X[*ced][2]);
/* equação C.4 */

```

```
ALPHA[2][1] = ALPHA[2][1] + PARALPHA[2][1];
```

```

PARALPHA[2][2] = (-COEF[11][ii]*TA + COEF[5][ii]* X[*ced][1]*TA - COEF[7][ii]*
pow( X[*ced][1],2.0)*TA)*exp(-TA/ X[*ced][2])*(TA/pow( X[*ced][2],4.0) - 2.0/pow

```

```
(X[*ced][2],3.)) + (COEF[2][ii]* X[*ced][1] - COEF[4][ii]*pow( X[*ced][1],2.)) *
( X[*ced][3]-2.*TA)*exp(-2.*TA- X[*ced][3])/ X[*ced][2])*((2.*TA- X[*ced][3])/pow
(X[*ced][2],4.) - 2./pow( X[*ced][2],3.)) + (COEF[1][ii]* X[*ced][1] + COEF[10][ii]*
pow( X[*ced][1],2.0))* (TA- X[*ced][3])*exp(-(TA- X[*ced][3])/ X[*ced][2]) * ((TA-
X[*ced][3])/pow(X[*ced][2],4.) - 2.0/pow( X[*ced][2],3.0)) + (COEF[12][ii]*TA-2.* COEF[6][ii]*
X[*ced][1]*TA+ COEF[8][ii]*pow( X[*ced][1],2.)*TA)*exp(-2.0*TA/ X[*ced][2])*
(2.0*TA/pow( X[*ced][2],4.0) - 2.0/pow( X[*ced][2],3.0)) + (COEF[10][ii] + COEF[4][ii])
*pow( X[*ced][1],2.)*( X[*ced][3]-TA)* exp(-2.*(TA- X[*ced][3])/ X[*ced][2])*(2.0*(TA-
X[*ced][3])/pow( X[*ced][2],4.) - 2.0/pow( X[*ced][2],3.)); /* equação C.5 */
```

```
ALPHA[2][2] = ALPHA[2][2] + PARALPHA[2][2];
```

```
PARALPHA[2][3] = (COEF[1][ii]* X[*ced][1]/pow( X[*ced][2],2.) + COEF[10][ii]*
pow( X[*ced][1],2.)/pow( X[*ced][2],2.))*exp(-(TA- X[*ced][3])/ X[*ced][2])*
((TA- X[*ced][3])/ X[*ced][2]-1.) + (COEF[4][ii]*pow( X[*ced][1],2.)/pow( X[*ced][2],2.) -
COEF[2][ii]*X[*ced][1]/pow( X[*ced][2],2.))*exp(-2.*TA- X[*ced][3])/ X[*ced][2])*((2.*TA-
X[*ced][3])/ X[*ced][2]-1.) + (COEF[10][ii] + COEF[4][ii])*pow( X[*ced][1],2.)/pow
(X[*ced][2],2.)*exp(-2.*(TA- X[*ced][3])/ X[*ced][2])*(2.0*( X[*ced][3]-TA)/ X[*ced][2] + 1.);
/* equação C.6 */
```

```
ALPHA[2][3] = ALPHA[2][3] + PARALPHA[2][3];
```

```
PARALPHA[3][1] = COEF[1][ii]/ X[*ced][2]*exp(-(TA- X[*ced][3])/ X[*ced][2]) - COEF[2][ii]/
X[*ced][2]*exp(-2.*TA- X[*ced][3])/ X[*ced][2]) + 2.* COEF[10][ii]* X[*ced][1]/ X[*ced][2]*
exp(-(TA- X[*ced][3])/ X[*ced][2]) + 2.0*pow(COEF[3][ii],2.)* X[*ced][1]/ X[*ced][2]*
exp(-2.*(TA- X[*ced][3])/ X[*ced][2]) + 2.* COEF[4][ii]* X[*ced][1]/ X[*ced][2]*
exp(-2.*TA- X[*ced][3])/ X[*ced][2]); /* equação C.7 */
```

```
ALPHA[3][1] = ALPHA[3][1] + PARALPHA[3][1];
```

```
PARALPHA[3][2] = (COEF[1][ii]* X[*ced][1] + COEF[10][ii]*pow( X[*ced][1],2.0))*
exp(-(TA- X[*ced][3])/ X[*ced][2])*((TA- X[*ced][3])/pow( X[*ced][2],3.) - 1.0/pow
(X[*ced][2],2.)) + (COEF[4][ii]*pow( X[*ced][1],2.) - COEF[2][ii]* X[*ced][1])*exp(-2.*TA-
X[*ced][3])/ X[*ced][2])* ((2.*TA- X[*ced][3])/pow( X[*ced][2],3.) - 1./pow( X[*ced][2],2.)) +
pow(COEF[3][ii],2.)*pow( X[*ced][1],2.)*exp(-2.*(TA- X[*ced][3])/ X[*ced][2])*(2.*(TA-
X[*ced][3])/pow( X[*ced][2],3.) - 1./pow( X[*ced][2],2.)); /* equação C.8 */
```

```
ALPHA[3][2] = ALPHA[3][2] + PARALPHA[3][2];
```

```
PARALPHA[3][3] = COEF[1][ii]* X[*ced][1]/pow( X[*ced][2],2.)*exp(-(TA- X[*ced][3])/ X[*ced][2])
- COEF[2][ii]* X[*ced][1]/pow( X[*ced][2],2.)*exp(-2.*TA- X[*ced][3])/ X[*ced][2]) +
COEF[10][ii]*pow( X[*ced][1],2.)/pow( X[*ced][2],2.)*exp(-(TA- X[*ced][3])/ X[*ced][2]) +
2.*pow(((COEF[3][ii]* X[*ced][1]/ X[*ced][2],2.)*exp(-2.*(TA- X[*ced][3])/ X[*ced][2]) +
COEF[4][ii]* pow(( X[*ced][1]/ X[*ced][2],2.)* exp(-2.*TA- X[*ced][3])/ X[*ced][2]));
/* equação C.9 */
```

```
ALPHA[3][3] = ALPHA[3][3] + PARALPHA[3][3];
```

```
}
```

```
printf("\n\n USRFUN1 - ALPHA={11 %f,12 %f,13 %f}",ALPHA[1][1],ALPHA[1][2],ALPHA[1][3]);
printf("\n USRFUN1 - ALPHA={21 %f,22 %f,23 %f}",ALPHA[2][1],ALPHA[2][2],ALPHA[2][3]);
printf("\n USRFUN1 - ALPHA={31 %f,32 %f,33 %f}",ALPHA[3][1],ALPHA[3][2],ALPHA[3][3]);
printf("\n BETA={ %10.9f, %10.9f, %10.9f}",BETA[1],BETA[2],BETA[3]);
```

```
/* ALPHA e BETA contêm números muito pequenos:
multiplicar por 1000 para Newton-Raphson convergir */
```

```
for (ii=1;ii<=N;ii++)
```

```

{ BETA[ii] = BETA[ii] * 1000.0;
  for (jj=1;jj<=N;jj++)
    { ALPHA[ii][jj] = ALPHA[ii][jj] * 1000.0;
    }
}
}

```

```
void USRFUN2(MANIP,CONTRO,COEF,cna,na)
```

```
float *MANIP, *CONTRO, COEF[13][23];
int *cna, *na;
```

```
/* MANIP = vetor histórico da variável (desvio) manipulada
   CONTRO = vetor histórico amostrado da variável (desvio) controlada */
{
```

```
/* Subrotina chamada na subrotina principal de controle para cálculo dos coeficientes COEF
   utilizados nas funções a serem zeradas (anexo B) */
```

```
int ii, jj;
```

```
if (*na>17)
{ for (ii=2;ii<=*cna;ii++)
  { for (jj=1;jj<=12;jj++)
    { COEF[jj][ii-1]=COEF[jj][ii];
    }
  }
}

```

```
COEF[1][*cna] = MANIP[*cna-1] * CONTRO[*cna] - MANIP[*cna-2] * CONTRO[*cna]; /* A */
```

```
COEF[2][*cna] = MANIP[*cna-1] * CONTRO[*cna-1] - MANIP[*cna-2] * CONTRO[*cna-1]; /* B */
```

```
COEF[3][*cna] = MANIP[*cna-1] - MANIP[*cna-2]; /* C */
```

```
COEF[4][*cna] = (MANIP[*cna-1] - MANIP[*cna-2]) * MANIP[*cna-2]; /* D */
```

```
COEF[5][*cna] = MANIP[*cna-2] * CONTRO[*cna] + MANIP[*cna-1] * CONTRO[*cna-1]; /* E */
```

```
COEF[6][*cna] = MANIP[*cna-2] * CONTRO[*cna-1]; /* F */
```

```
COEF[7][*cna] = MANIP[*cna-1] * MANIP[*cna-2]; /* G */
```

```
COEF[8][*cna] = MANIP[*cna-2] * MANIP[*cna-2]; /* H */
```

```
COEF[9][*cna] = MANIP[*cna-1] * CONTRO[*cna]; /* J */
```

```
COEF[10][*cna] = MANIP[*cna-1] * MANIP[*cna-2] - MANIP[*cna-1] * MANIP[*cna-1]; /* L */
```

```
COEF[11][*cna] = CONTRO[*cna-1] * CONTRO[*cna]; /* M */
```

```
COEF[12][*cna] = CONTRO[*cna-1] * CONTRO[*cna-1]; /* N */
```

```

if (*cna==3)
{ for (jj=1;jj<=12;jj++)
  { COEF[jj][1] = COEF[jj][3];
    COEF[jj][2] = COEF[jj][3];
  }
}
}

void MPROVE(ALPHAVE,ALPHA,N,INDX,BETA,dX)

float ALPHA[5][5],ALPHAVE[5][5],*INDX,*BETA,*dX;
int N;

{

/* sub-rotina para melhorar o vetor-solução dos incrementos de Newton-Raphson */

float SDP,R[5];
int ii,jj;

for (ii=1;ii<=N;ii++)
{ SDP = -BETA[ii];
  for (jj=1;jj<=N;jj++)
  { SDP = SDP + ALPHAVE[ii][jj] * dX[jj];
  }
  R[ii] = SDP;
}

LUBKSB(ALPHA,N,INDX,R);
/* Resolução das equações algébricas lineares com matriz decomposta por LUDCMP */

for (ii=1;ii<=N;ii++)
{ dX[ii] = dX[ii] - R[ii]; /* incrementos de Newton-Raphson */
}

printf("\n\nMPROVE dX = {%f, %f, %f}",dX[1],dX[2],dX[3]);

}

```

**ANEXO 6 - IMPLEMENTAÇÃO DO ALGORITMO
DE TREINAMENTO GDR.**

```
void GDR (pR,px1,pT,v,w,Th,na)
```

```
float pR[21], pT[21], px1[21], v[63][63], w[63][10], Th[10][63];
int *na;
```

```
/* pR = vetor sequência de ações de controle implementadas
   px1 = vetor sequência de valores amostrados da composição de topo da coluna
   pT = vetor sequência de valores amostrados da temperatura de fundo da coluna
   v e w = matrizes de fatores ponderais de conexões entre neurônios artificiais
   Th = matriz dos valores residuais internos dos neurônios artificiais
   *na = ponteiro indicativo do número de amostragem */
```

```
{ /* Treinamento da Rede Neural Artificial com o Algoritmo de Regra de Delta Generalizado */
```

```
int ncam=3, Ln=6, Mn=15, Nn=2; /* Ln tem que ser múltiplo de 3 */
int ii, jj, kk, nn, ite=1;
float Dv[63][63], Dw[63][10], X[10][63], Aout[63], SOMX[10][63];
float ETA=3.0, ALPHA=0.7, Bout[63], Cout[10], DIF[10], SQDIF=1.e-20;
float DERV[10][63], DELTA[10][63], SOMDELTA3[63], DELTAV[63][63], DELTAW[63][10];
float PNPOT[63], Dout[10], aux;
clock_t t1, t2;
float dt;
```

```
FILE *fpe;
if((fpe=fopen("tempo.dat","a+b"))==NULL) {
    printf ("\n Cannot open file\n\n");
    getche(); }
```

```
/* Ln, Mn e Nn = números de neurônios da 1a., 2a. e 3a. camadas (máx. 60,60,10)
   Aout, Bout e Cout = saídas dos neurônios da 1a., 2a. e 3a. camadas
   Dout = vetor de saída conhecido da rede */
```

```
if (*na == 1 )
{ for (ii=0;ii<=40;ii=ii+20)
  { for (jj=0;jj<=54;jj=jj+6)
    {v[1+jj][1+ii] = -1.0; v[1+jj][6+ii] = 0.5; /* valores iniciais aleatórios p/ pesos v */
      v[1+jj][2+ii] = -0.5; v[1+jj][7+ii] = 1.0;
      v[1+jj][3+ii] = 0.5; v[1+jj][8+ii] = 0.0;
      v[1+jj][4+ii] = 0.0; v[1+jj][9+ii] = 1.0;
      v[1+jj][5+ii] = 0.5; v[1+jj][10+ii] = -0.5;
      v[1+jj][11+ii] = 0.0; v[1+jj][16+ii] = 0.5;
      v[1+jj][12+ii] = 1.0; v[1+jj][17+ii] = 0.0;
      v[1+jj][13+ii] = -0.5; v[1+jj][18+ii] = 0.5;
      v[1+jj][14+ii] = -1.0; v[1+jj][19+ii] = 1.0;
      v[1+jj][15+ii] = -0.5; v[1+jj][20+ii] = -0.5;
      v[2+jj][1+ii] = 1.0; v[2+jj][6+ii] = -1.0;
      v[2+jj][2+ii] = 0.5; v[2+jj][7+ii] = -0.5;
      v[2+jj][3+ii] = 0.0; v[2+jj][8+ii] = 0.5;
      v[2+jj][4+ii] = 1.0; v[2+jj][9+ii] = 0.0;
      v[2+jj][5+ii] = -0.5; v[2+jj][10+ii] = 0.5;
      v[2+jj][11+ii] = -0.5; v[2+jj][16+ii] = 1.0;
      v[2+jj][12+ii] = 0.5; v[2+jj][17+ii] = 0.5;
      v[2+jj][13+ii] = 0.0; v[2+jj][18+ii] = -0.5;
      v[2+jj][14+ii] = 0.5; v[2+jj][19+ii] = -1.0;
      v[2+jj][15+ii] = 1.0; v[2+jj][20+ii] = 0.5;
```

```

v[3+jj][1+ii] = -1.0; v[3+jj][6+ii] = 0.0;
v[3+jj][2+ii] = -0.5; v[3+jj][7+ii] = 1.0;
v[3+jj][3+ii] = 0.5; v[3+jj][8+ii] = 0.5;
v[3+jj][4+ii] = 0.0; v[3+jj][9+ii] = -0.5;
v[3+jj][5+ii] = 0.5; v[3+jj][10+ii] = 0.0;
v[3+jj][11+ii] = 0.0; v[3+jj][16+ii] = 0.5;
v[3+jj][12+ii] = 1.0; v[3+jj][17+ii] = 0.0;
v[3+jj][13+ii] = -0.5; v[3+jj][18+ii] = 0.5;
v[3+jj][14+ii] = -1.0; v[3+jj][19+ii] = -0.5;
v[3+jj][15+ii] = -0.5; v[3+jj][20+ii] = 1.0;
v[4+jj][1+ii] = 1.0; v[4+jj][6+ii] = -1.0;
v[4+jj][2+ii] = 0.5; v[4+jj][7+ii] = -0.5;
v[4+jj][3+ii] = 0.0; v[4+jj][8+ii] = 0.5;
v[4+jj][4+ii] = 1.0; v[4+jj][9+ii] = 0.0;
v[4+jj][5+ii] = -0.5; v[4+jj][10+ii] = 1.0;
v[4+jj][11+ii] = -0.5; v[4+jj][16+ii] = 1.0;
v[4+jj][12+ii] = 0.5; v[4+jj][17+ii] = 0.5;
v[4+jj][13+ii] = 0.0; v[4+jj][18+ii] = -0.5;
v[4+jj][14+ii] = 0.5; v[4+jj][19+ii] = -1.0;
v[4+jj][15+ii] = 1.0; v[4+jj][20+ii] = 0.5;
v[5+jj][1+ii] = -1.0; v[5+jj][6+ii] = 1.0;
v[5+jj][2+ii] = -0.5; v[5+jj][7+ii] = 0.5;
v[5+jj][3+ii] = 0.5; v[5+jj][8+ii] = 0.0;
v[5+jj][4+ii] = 0.0; v[5+jj][9+ii] = 1.0;
v[5+jj][5+ii] = 0.5; v[5+jj][10+ii] = -0.5;
v[5+jj][11+ii] = 0.0; v[5+jj][16+ii] = 0.5;
v[5+jj][12+ii] = 1.0; v[5+jj][17+ii] = 0.0;
v[5+jj][13+ii] = -0.5; v[5+jj][18+ii] = 0.5;
v[5+jj][14+ii] = -1.0; v[5+jj][19+ii] = 1.0;
v[5+jj][15+ii] = -0.5; v[5+jj][20+ii] = -1.0;
v[6+jj][1+ii] = 1.0; v[6+jj][6+ii] = -1.0;
v[6+jj][2+ii] = 0.5; v[6+jj][7+ii] = -0.5;
v[6+jj][3+ii] = 0.0; v[6+jj][8+ii] = 0.5;
v[6+jj][4+ii] = 1.0; v[6+jj][9+ii] = 1.0;
v[6+jj][5+ii] = -0.5; v[6+jj][10+ii] = 0.5;
v[6+jj][11+ii] = 0.0; v[6+jj][16+ii] = 0.5;
v[6+jj][12+ii] = 1.0; v[6+jj][17+ii] = 0.0;
v[6+jj][13+ii] = -0.5; v[6+jj][18+ii] = 0.5;
v[6+jj][14+ii] = -1.0; v[6+jj][19+ii] = -0.5;
v[6+jj][15+ii] = -0.5; v[6+jj][20+ii] = 1.0;
}
w[1+ii][1] = 1.0; w[6+ii][1] = 0.0; /* valores iniciais aleatórios p/ pesos w */
w[2+ii][1] = 0.5; w[7+ii][1] = -0.5;
w[3+ii][1] = -0.5; w[8+ii][1] = 1.0;
w[4+ii][1] = 0.0; w[9+ii][1] = -1.0;
w[5+ii][1] = 0.5; w[10+ii][1] = 0.0;
w[11+ii][1] = 0.0; w[16+ii][1] = 0.5;
w[12+ii][1] = 1.0; w[17+ii][1] = 0.0;
w[13+ii][1] = -0.5; w[18+ii][1] = 0.5;
w[14+ii][1] = -1.0; w[19+ii][1] = 1.0;
w[15+ii][1] = -0.5; w[20+ii][1] = -1.0;
w[1+ii][2] = 1.0; w[6+ii][2] = 0.0;
w[2+ii][2] = 0.5; w[7+ii][2] = 1.0;
w[3+ii][2] = -0.5; w[8+ii][2] = -0.5;
w[4+ii][2] = 0.0; w[9+ii][2] = 0.5;
w[5+ii][2] = 0.5; w[10+ii][2] = 0.0;
w[11+ii][2] = -0.5; w[16+ii][2] = 1.0;

```

```

w[12+ii][2] = 0.0; w[17+ii][2] = -1.0;
w[13+ii][2] = 0.5; w[18+ii][2] = 0.0;
w[14+ii][2] = 0.0; w[19+ii][2] = 0.5;
w[15+ii][2] = 1.0; w[20+ii][2] = 1.0;
}
}
t1=clock();

jj = 0;
for (ii=1;ii<=Ln;ii=ii+3) /* janela de entrada da rede */
{
  jj = jj + 1;
  PNPOT[ii] = pR[jj];
  PNPOT[1+ii] = pT[jj+1];
  PNPOT[2+ii] = px1[jj+1];
}

Dout[1] = px1[1]; /* fixo: dois neurônios de saída */
Dout[2] = pT[1];

for (ii=1;ii<=Ln;ii++)
{
  for (jj=1;jj<=Mn;jj++)
  {
    DELTAV[ii][jj] = 1.e-10; /* zerar vetor-correção p/ pesos v */
  }
}

for (jj=1;jj<=Mn;jj++)
{
  for (kk=1;kk<=Nn;kk++)
  {
    DELTAW[jj][kk] = 1.e-10; /* zerar vetor-correção p/ pesos w */
  }
}

for (ii=1;ii<=ncam;ii++)
{
  for (jj=1;jj<=63;jj++)
  {
    if (ii==1) Th[ii][jj] = 1.e-10; /* valores fixos p/ bias Th */
    else Th[ii][jj] = 1.0;
  }
}

do {

  /* saída da 1a. camada */
  for (ii=1;ii<=Ln;ii++)
  {
    X[1][ii] = PNPOT[ii] - Th[1][ii];
    Aout[ii] = 1./(1. + exp(-X[1][ii]));
  }

  /* zerar somatórios */
  for (nn=2;nn<=ncam;nn++)
  {
    for (jj=1;jj<=63;jj++)
    {
      SOMX[nn][jj] = 1.e-10;
    }
  }

  /* saída da camada escondida */
  for (jj=1;jj<=Mn;jj++)
  {
    for (ii=1;ii<=Ln;ii++)
    {
      SOMX[2][jj] = SOMX[2][jj] + v[ii][jj]*Aout[ii];
    }
  }
}

```

```

X[2][jj] = SOMX[2][ij] + Th[2][jj];
Bout[jj] = 1./(1. + exp(-X[2][ij]));
}

```

```
/* saída da última camada */
```

```

for (kk=1;kk<=Nn;kk++)
{ for (jj=1;jj<=Mn;jj++)
  { SOMX[3][kk] = SOMX[3][kk] + w[jj][kk]*Bout[jj];
    SOMDELTA3[jj] = 1.e-5;
  }
  X[3][kk] = SOMX[3][kk] + Th[3][kk];
  Cout[kk] = 1./(1. + exp(-X[3][kk]));
  DIF[kk] = Dout[kk] - Cout[kk];
/* erros entre vetores de saída conhecido e calculado pela ANN */
}

```

```
if (fabs(DIF[1])>1.e-5 || fabs(DIF[2])>1.e-5) /* critério para modificar pesos */
```

```
{ ite = ite + 1; /* número de iterações necessárias p/ convergência */
```

```
/* somatório do erro quadrático */
```

```

SQDIF = 1.e-20;
for (kk=1;kk<=Nn;kk++)
{ SQDIF = SQDIF + pow(DIF[kk],2.0);
}

```

```
/* gradiente descendente p/ última camada */
```

```

for (kk=1;kk<=Nn;kk++)
{ DERV[3][kk] = exp(-X[3][kk])/pow((1. + exp(-X[3][kk])),2.0);
  DELTA[3][kk] = DIF[kk] * DERV[3][kk];
}

```

```
/* gradiente descendente p/ camada escondida */
```

```

for (jj=1;jj<=Mn;jj++)
{ DERV[2][ij] = exp(-X[2][ij])/pow((1. + exp(-X[2][ij])),2.0);
  for (kk=1;kk<=Nn;kk++)
  { SOMDELTA3[jj] = SOMDELTA3[jj] + DELTA[3][kk] * w[jj][kk];
  }
  DELTA[2][ij] = DERV[2][ij] * SOMDELTA3[jj];
}

```

```
/* Cálculos dos vetores-correções de v e w */
```

```

for (ii=1;ii<=Ln;ii++)
{ for (jj=1;jj<=Mn;jj++)
  { DELTAV[ii][jj] = (ETA * DELTA[2][ij] * Aout[ii]) + (ALPHA *
    DELTAV[ii][jj]);
    v[ii][jj] = v[ii][jj] + DELTAV[ii][jj]; /* correção dos pesos v */
  }
}
for (jj=1;jj<=Mn;jj++)
{ for (kk=1;kk<=Nn;kk++)
  { DELTAW[jj][kk] = (ETA * DELTA[3][kk] * Bout[jj]) + (ALPHA *
    DELTAW[jj][kk]);
    w[jj][kk] = w[jj][kk] + DELTAW[jj][kk]; /* correção de w */
  }
}
}

```

```
    } while (fabs(DIF[1])>1.e-5 || fabs(DIF[2])>1.e-5); /* critério p/ conclusão do treinamento */

    t2 = clock();
    dt = (t2 - t1)/(CLK_TCK);

    printf ("\n Treinamento efetuado em %f segundos",dt);
    fprintf (fpe, "%f %f %d %f\n",t,dt,ite,SQDIF);
    fclose (fpe);
}
```

**ANEXO 7 - IMPLEMENTAÇÃO DO MÉTODO
DE OTIMIZAÇÃO DE POWELL.**

```
void POWELL(Hpred,Hcont,sp,INPUT,v,w,Th,ACAOTIM,DIR,xpred,disturb)
```

```
int Hpred,Hcont;
```

```
float sp,INPUT[20],v[20][20],w[20][10],Th[10][20],ACAOTIM[20],DIR[20][20],xpred[10],*disturb;
```

```
/* Hpred = horizonte de predição
```

```
   Hcont = horizonte de controle
```

```
   sp = "set point" de composição de topo
```

```
   INPUT = vetor móvel de entrada da rede neural, usado na predição via RNA
```

```
   v,w,Th = matrizes usadas nas predições via RNA, determinadas na sub-rotina GDR
```

```
   ACAOTIM = vetor de ações de controle otimizadas
```

```
   DIR = direção de busca
```

```
   xpred = vetor de predições da variável controlada
```

```
   disturb = discrepância do modelo */
```

```
{ /* minimização da função objetivo */
```

```
float ftol=1.0e-8,AOT[20],fp,del,fptt,ptt[20],tt,xit[20],*fret;
```

```
int itmax=200,iter=0,ii,ibig,jj;
```

```
fret = (float *) malloc(sizeof(float));
```

```
*fret = NULL;
```

```
for (ii=0;ii<=(Hcont+1);ii++) AOT[ii] = ACAOTIM[ii]; /* salva ponto inicial */
```

```
*fret = FUNCOBJ(Hpred,Hcont,sp,INPUT,v,w,Th,ACAOTIM,xpred,disturb);
```

```
for (iter=1;;++iter)
```

```
{ fp = *fret;
```

```
  ibig = 0;
```

```
  del = 0.0;
```

```
  for (ii=1;ii<=(Hcont+1);ii++) /* muda uma direção de cada vez em linmin */
```

```
    { for (jj=1;jj<=(Hcont+1);jj++)
```

```
      { xit[jj] = DIR[jj][ii]; /* copia a direção */
```

```
      }
```

```
    fptt = *fret;
```

```
    linmin(ACAOTIM,xit,Hcont,fret,Hpred,INPUT,v,w,Th,sp,xpred,disturb);
```

```
    /* linmin minimiza a função objetivo através da localização
```

```
    do ponto de mínimo ACAOTIM na direção xit */
```

```
    if (fabs(fptt - (*fret))>del)
```

```
      { del = fabs(fptt - (*fret));
```

```
        ibig = ii;
```

```
      }
```

```
    }
```

```
    if (2.0*fabs(fp - (*fret)) <= ftol*(fabs(fp)+fabs(*fret)))
```

```
      { FUNCOBJ(Hpred,Hcont,sp,INPUT,v,w,Th,ACAOTIM,xpred,disturb);
```

```
        free(fret);
```

```
        return;
```

```
      }
```

```
    if (iter == itmax) /* critério para acabar */
```

```
      { printf("\n\n POWELL EXCEDEU NÚMERO DE ITERAÇÕES MÁXIMO!! \n\n");
```

```
        getche();
```

```
      }
```

```
    ptt[0] = ACAOTIM[0];
```

```
    for (jj=1;jj<=(Hcont+1);jj++)
```

```
      { /* extrapola o ponto de mínimo e muda a direção média;
```



```

        INP[6] = INP[3];
        INP[1] = ACAOTI[Hcont+1]/30.0; /* acima do Hcont a raz. ref. é fixa */
        INP[2] = OUTPRED[2];
        INP[3] = OUTPRED[1] + *disturb;
        PRED(INP,OUTPRED,v,w,Th);
        xpred[ii] = OUTPRED[1] + *disturb;
    }
}

/* cálculo da função objetivo */

for (ii=1;ii<=Hpred;ii++)
    { soma1 = soma1 + pow((sp - xpred[ii]),2.0);
    }
for (ii=1;ii<=(Hcont+1);ii++)
    { soma2 = soma2 + lamb[ii] * pow((ACAOTI[ii] - ACAOTI[ii-1]),2.0);
    }
FO = fabs(soma1 + soma2);
return FO;
}

void linmin(ACAOTIM,xit,Hcont,fret,INPUT,v,w,Th,sp,xpred,disturb)

float ACAOTIM[20],xit[20],*fret,INPUT[20],v[20][20],w[20][10],Th[10][20],sp,xpred[10],*disturb;
int Hcont,Hpred;

{ /* move ACAOTIM na direção xit para achar o mínimo */

int jj;
float TOL=1.e-4, ax,xx,bx,fa,fx,fb,xmin=1.0;

ax=0.0;
xx=1.0;

mnbrak(&ax,&xx,&bx,&fa,&fx,&fb,ACAOTIM,xit,Hcont,Hpred,INPUT,v,w,Th,sp,xpred,disturb);
/* sub-rotina que busca o mínimo da função */

*fret = brent(ax,xx,bx,ACAOTIM,xit,Hcont,TOL,&xmin,Hpred,INPUT,v,w,Th,sp,xpred,disturb);
/* sub-rotina que isola o mínimo da função */

for (jj=1;jj<=(Hcont+1);jj++) /* vetores-resultado que retornam p/ powell */
    { xit[jj] =xmin * xit[jj];
      ACAOTIM[jj] = ACAOTIM[jj] + xit[jj];
    }
}
}

```

```
void mnbrak(ax,bx,cx,fa,fb,fc,ACAOTIM,xit,Hcont,Hpred,INPUT,v,w,Th,sp,xpred,disturb)
```

```
float *ax, *bx, *cx, *fa, *fb, *fc, ACAOTIM[20], xit[20],xpred[10],*disturb;
float INPUT[20],v[20][20],w[20][10],Th[10][20], sp;
int Hcont, Hpred;
```

```
{ /* dada a função fl dim e os ptos. iniciais ax e bx, esta rotina pesquisa
na direção de inclinação abaixo e retorna novos pontos ax,bx e cx que suportam o
mínimo da função */
```

```
float dum, GOLD=1.618034, r,q,*u,ulim,TINY=1.0e-20,GLIMIT=100.0,fu;
u = (float *) malloc(sizeof(float));
*u = NULL;
```

```
*fa = fl dim(ACAOTIM,xit,Hcont,ax,Hpred,INPUT,v,w,Th,sp,xpred,disturb);
*fb = fl dim(ACAOTIM,xit,Hcont,bx,Hpred,INPUT,v,w,Th,sp,xpred,disturb);
/* sub-rotina que transforma a busca em unidimensional */
```

```
if (*fb > *fa)
{ dum = *ax; /* indo inclinação abaixo de ax para bx; */
*ax = *bx;
*bx = dum;
dum = *fb;
*fb = *fa;
*fa = dum;
}
```

```
*cx = *bx + GOLD*((*bx) - (*ax)); /* estimativa inicial para cx */
```

```
*fc = fl dim(ACAOTIM,xit,Hcont,cx,Hpred,INPUT,v,w,Th,sp,xpred,disturb);
```

```
while ((*fb) >= (*fc)) /* calculo de u por extrapolação parabólica a partir de a,b,c */
{ r = (*bx - (*ax)) * (*fb - (*fc));
q = (*bx - (*cx)) * (*fb - (*fa));
*u = *bx - ((*bx - (*cx))*q - (*bx - (*ax))*r)/(2.0*SIGN(FMAX(fabs(q-r),TINY),q-r));
/* TINY evita divisão por zero */
ulim = (*bx) + GLIMIT * (*cx - (*bx)); /* nao passar deste limite */
```

```
if (((*bx) - (*u))*((*u) - (*cx)) > 0.0)
{ fu = fl dim(ACAOTIM,xit,Hcont,u,Hpred,INPUT,v,w,Th,sp,xpred,disturb);
if (fu < (*fc))
{ *ax = *bx; /* pega o mínimo entre bx e cx */
*bx = *u;
*fa = *fb;
*fb = fu;
free(u);
return;
}
else if (fu > (*fb))
{ *cx = *u; /* pega o mínimo entre ax e u */
*fc = fu;
free(u);
return;
}
*u = (*cx) + GOLD*( *cx - (*bx));
/* ajuste parabólico não estava em uso; usar aumento padrão */
```

```

    fu = fl dim(ACAOTIM,xit,Hcont,u,Hpred,INPUT,v,w,Th,sp,xpred,disturb);
}

else if (((*cx) - (*u)) * ((*u) - ulim) > 0.0)
    { /* ajuste parabólico está entre cx e seu limite permitido */
        fu = fl dim(ACAOTIM,xit,Hcont,u,Hpred,INPUT,v,w,Th,sp,xpred,disturb);

        if (fu < (*fc))
            { *bx = *cx;
              *cx = *u;
              *u = *cx + GOLD * (*cx - (*bx));
              *fb = *fc;
              *fc = fu;
              fu = fl dim(ACAOTIM,xit,Hcont,u,Hpred,INPUT,v,w,Th,sp,xpred,disturb);
            }
        }
    else if (((*u) - ulim) * (ulim - (*cx)) >= 0.0)
        { *u = ulim;
          fu = fl dim(ACAOTIM,xit,Hcont,u,Hpred,INPUT,v,w,Th,sp,xpred,disturb);
        }
    else { *u = (*cx) + GOLD * (*cx - (*bx));
          fu = fl dim(ACAOTIM,xit,Hcont,u,Hpred,INPUT,v,w,Th,sp,xpred,disturb);
        }

    *ax = *bx; /* elimina o pto. mais velho e continua */
    *bx = *cx;
    *cx = *u;
    *fa = *fb;
    *fb = *fc;
    *fc = fu;

} /* fecha o while */

}

```

```
float brent(ax,bx,cx,ACAOTIM,xit,Hcont,TOL,xmin,Hpred,INPUT,v,w,Th,sp,xpred,disturb)
```

```
float ACAOTIM[20], xit[20],TOL,ax,bx,cx,*xmin,sp,xpred[10],*disturb;
float INPUT[20],v[20][20],w[20][10],Th[10][20];
int Hcont,Hpred;
```

```
{ /* Dada a função fl dim e as abscissas ax,bx e cx, esta rotina isola o
    mínimo numa precisão de aproximadamente TOL usando o método de Brent */
```

```
int iter,itmax=200;
float a,b,*xb,wb,vb,*ub,fu,fv,fx,e=0.0,xm,ZEPS=1.e-10,r,q,p;
float tol1,tol2,etemp,d,CGOLD=0.3819660;
xb = (float *) malloc(sizeof(float));
ub = (float *) malloc(sizeof(float));
*xb = NULL;
*ub = NULL;
```

```
a = (ax<cx ? ax:cx);
b = (ax>cx ? ax:cx);
```

```

*xb = wb = vb = bx;

fw = fv = fx = fl dim(ACAOTIM,xit,Hcont,xb,Hpred,INPUT,v,w,Th,sp,xpred,disturb);

for (iter=1;iter<=itmax;iter++)
{
  xm = 0.5 * (a+b);
  toll = TOL * fabs(*xb) + ZEPS;
  tol2 = 2.0 * toll;

  if (fabs(*xb - xm) <= (tol2 - 0.5*(b - a))) /* teste para acabar */
  {
    *xmin = *xb;
    free(xb);
    free(ub);
    return fx;
  }

  if (fabs(e) > toll)
  {
    r = (*xb - wb) * (fx - fv); /* ajuste parabólico */
    q = (*xb - vb) * (fx - fw);
    p = (*xb - vb) * q - (*xb - wb) * r;
    q = 2.0*(q-r);
    if (q>0.0) p = -p;
    q = fabs(q);
    etemp = e;
    e = d;

    if (fabs(p) >= fabs(0.5*q*etemp) || p<=(q*(a-(*xb))) || p>=(q*(b-(*xb))))
    { /* condições de aceitabilidade do ajuste parabólico */
      e = ((*xb)>=xm ? a-(*xb) : b-(*xb));
      d = CGOLD * e;
    }
    else { d = p/q;
          *ub = (*xb) + d;
          if (((*ub)-a)<tol2 || (b-(*ub))<tol2) d=SIGN(tol1,xm-(*xb));
        }
  }
  else { e = ((*xb)>=xm ? a-(*xb) : b-(*xb));
        d = CGOLD * e;
      }
  *ub = (fabs(d)>=toll ? (*xb)+d : (*xb)+SIGN(tol1,d));

  fu = fl dim(ACAOTIM,xit,Hcont,ub,Hpred,INPUT,v,w,Th,sp,xpred,disturb);

  if (fu<=fx)
  {
    if ((*ub)>=(*xb)) a=(*xb);
    else b=(*xb);
    vb = wb;
    fv = fw;
    wb = (*xb);
    fw = fx;
    *xb = *ub;
    fx = fu;
  }
  else { if ((*ub)<(*xb)) a = *ub;
        else b = *ub;
        if (fu<=fw || wb==(*xb))
        { vb = wb;

```

```

        wb = *ub;
        fv = fw;
        fw = fu;
    }
    else if (fu <= fv || vb == (*xb) || vb == wb)
        { vb = *ub;
          fv = fu;
        }
    }
} /* fecha o for */

printf ("\n BRENT excedeu o número máximo de iterações \n");
getche();

*xmin = (*xb);
free(xb);
free(ub);
return fx;

}

float fldim(ACAOTIM,xit,Hcont,xf1,Hpred,INPUT,v,w,Th,sp,xpred,disturb)

float ACAOTIM[20],xit[20],*xf1,INPUT[20],v[20][20],w[20][10],Th[10][20],sp,xpred[10],*disturb;
int Hcont,Hpred;

{ /* função usada para transformar a busca em unidimensional */

int jj;
float f,xt[20];

xt[0] = ACAOTIM[0]; /* ação medida e portanto não pode ser otimizada */
for(jj=1;jj<=(Hcont+1);jj++)
    { xt[jj] = ACAOTIM[jj] + *xf1 * xit[jj];
    }

f = FUNCOBJ(Hpred,Hcont,sp,INPUT,v,w,Th,xt,xpred,disturb);

return f;

}

```

REFERÊNCIAS BIBLIOGRÁFICAS.

Abrams, H. J., Miladi, M. M., Attarwala, F. T., "Preferable Alternatives to Conventional Batch Distillation", *Distillation 87*, Brighton 7-9 September (1987).

Arkun, Y., Hollett, J., Canney, W. M., Morari, M., "Experimental study of internal model control", *Ind. Engng Chem. Process Des. Dev.*, **25**, 102 (1986).

Asher, R. B., Adrisani, D., Dorato, P., "Bibliography on adaptive control systems", *IEEE Proc.*, **64**, 1126 (1976) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).

Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).

Aström, K. J., "Ziegler-Nichols auto-tuners", Report CODEN: LUTFD2/(TFRT-3167)/01-025/(1982) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).

Aström, K. J., Wittenmark, B., "On self tuning regulators", *Automatica*, **9**, 185 (1973) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).

Aström, K. J., Wittenmark, B., "Problems of identification and control", *J. Math. Anal. & Applic.*, **34**, 90 (1971) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).

Bellman, R., *Adaptive Processes - A Guide Tour*, Princeton University Press (1961) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).

Bellman, R., *Dynamic Programming*, Princeton University Press (1957) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).

Bhat, N., McAvoy, T. J., "Use of Neural Nets for Dynamic Modeling and Control of Chemical Process Systems", *Comput. Chem. Eng.*, **14**, 573 (1990).

Bhat, N., Minderman Jr., P.A., McAvoy, T. J., Wang, N. S., "Modeling Chemical Process Systems via Neural Computation", *IEEE Control Systems Magazine*, **24**, Abril (1990).

Borisson, U., Syding, R., *Automatica*, **12**, 1 (1976) *apud* Dahlqvist, S. A., "Control of a distillation column using self-tuning regulators", *The Canadian Journal of Chemical Engineering*, **59**, 118 (1981).

Borisson, U., Wittenmark, B., "An industrial application of a self-tuning regulator", *4th IFAC Conference on Digital Computer Applications to Process Control*, Zurique, Suíça

(1974) *apud* Dahlqvist, S. A., "Control of a distillation column using self-tuning regulators", *The Canadian Journal of Chemical Engineering*, **59**, 118 (1981).

Bozenhardt, H., "Better Sensor/Control Utilization Improves Batch Distillations", *InTech*, **34**(3), 43 (1987).

Caldwell, W. I., "Control system with automatic response adjustment", Patente americana 2,517,081 (1950) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).

Cegrell, T., Hedqvist, T., *Automatica*, **11**, 53 (1975) *apud* Dahlqvist, S. A., "Control of a distillation column using self-tuning regulators", *The Canadian Journal of Chemical Engineering*, **59**, 118 (1981).

Cegrell, T., Hedqvist, T., "A new approach to continuous digester control", *4th IFAC Conference on Digital Computer Applications to Process Control*, Zurique, Suíça (1974) *apud* Dahlqvist, S. A., "Control of a distillation column using self-tuning regulators", *The Canadian Journal of Chemical Engineering*, **59**, 118 (1981).

Chang, T. S., Seborg, D. E., "A linear programming approach to multivariable feedback control with inequality constraints", *Int. J. Control*, **37**, 583 (1983).

Clarke, D. W., Mohtadi, C., Tuffs, P. S., "Generalized predictive control - I. The basic algorithm", *Automatica*, **23**, 137 (1987a).

Clarke, D. W., Mohtadi, C., Tuffs, P. S., "Generalized predictive control - II. Extensions and interpretations", *Automatica*, **23**, 149 (1987b).

Converse, A. O., Gross, G. D., "Optimal Distillate-rate Policy in Batch Distillation", *Ind. Engng Chem., Fundam.*, **2**, 217 (1963).

Coward, I., "The Time Optimal Problem in Binary Batch Distillation", *Chem. Engng Sci.*, **22**, 503 (1967a).

Coward, I., "The Time Optimal Problem in Binary Batch Distillation - A Further Note", *Chem. Engng Sci.*, **22**, 1881 (1967b).

Cuille, P. E., Reklaitis, G. V., "Dynamic Simulation of Multicomponent Batch Rectification with Chemical Reactions", *Computers chem. Engng*, **10**, 389 (1986).

Cutler, C. R., Hawkins, R. B., "Constrained multivariable control of a hydrocracker reactor", *Proc. Am. Control Conf.*, Minneapolis, Minnesota, 1014 (1987).

Cutler, C. R., Ramaker, B. L., "Dynamic matrix control - a computer control algorithm", *AIChE National Mtg.* Houston, Texas (1979).

- Dahlqvist, S. A., "Control of a distillation column using self-tuning regulators", *The Canadian Journal of Chemical Engineering*, **59**, 118 (1981).
- DeKeyser, R. M. C., van Cauwenberghe, A. R., "Extended prediction self-adaptive control", *IFAC Symp. on Identification and System Parameter Estimation*, York, U. K., 1255 (1985).
- DeKeyser, R. M. C., VandeVelde, Ph. G. A., Dumoriter, F. A. G., "A comparative study of self-adaptive long-range predictive control methods", *IFAC Symp. on Identification and System Parameter Estimation*, York, U.K., 1317 (1985).
- Diwekar, U. M., Malik, R. K., Madhavan, K. P., "Optimal Reflux Rate Policy Determination for Multicomponent Batch Distillation Columns", *Comp. Chem. Engng*, **11**, 629 (1987).
- Domenech, S., Guiglion, C., Enjalbert, M., "Modele Mathematique D'une Colonne Rectification Discontinue - II. Exploitation Numerique", *Chem. Engng Sci.*, **29**, 1529 (1974).
- Edgar, T. F., Himmelblau, D. M., "Optimization of Chemical Processes", McGraw-Hill, NY (1989).
- Feldbaum, A. A., "Dual control theory I-IV", *Aut. & Remote Control*, **21**, 874; **21**, 1033; **22**, 1; **22**, 109 (1960-1961) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).
- Feldbaum, A. A., "Optimal Control Systems", Academic Press, NY (1965) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).
- Fileti, A. M. F., "Estratégias de Controle em Destilação Batelada", Tese de Mestrado, FEQ/UNICAMP (1992).
- Fletcher, R., "Practical Methods of Optimization", Vol. 1, John Wiley, NY (1980) *apud* Willis, M. J., Montague, G. A., Di Massimo, C., Tham, M. T., Morris, A. J., "Artificial Neural Networks in Process Estimation and Control", *Automatica*, **28**, 1181 (1992).
- Garcia, C. E., "Quadratic dynamic matrix control of nonlinear processes. An application to a batch reaction process", *AIChE Annual Mtg*, San Francisco, California.
- Garcia, C. E., Prett, D. M., Morari, M., "Model Predictive Control: Theory and Practice - a Survey", *Automatica*, **25**, 335 (1989).
- Gregory, P. C., *Proceedings of the Self Adaptive Flight Control Systems Symposium*, WADC Technical Report 59-49, Wright Air Development Centre, Wright-Patterson Air Force Base, Ohio (1959) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).

- Grosdidier, P., Setpoint, Inc., Houston, Comunicação pessoal *apud* Garcia, C. E., Prett, D. M., Morari, M., "Model Predictive Control: Theory and Practice - a Survey", *Automatica*, **25**, 335 (1989).
- Guilandoust, M. T., Morris, A. J., Tham, M. T., "Adaptive Inferential Control", *Proc. IEE*, **134**, Pt.D., 171 (1987).
- Gutman, P. O., "Controllers for bilinear and constrained linear systems", Tese de Doutorado, Lund Institute of Technology (1982).
- Hansen, T. T., Jorgensen, S. B., "Optimal Control of Binary Distillation in Tray or Packed Columns", *Chem. Engng J.*, **33**, 151 (1987).
- Harris, C. J., Billings, S. A., *Self-tuning and Adaptive Control: Theory and Applications*, Peter Peregrinus, London (1981) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).
- Hoskins, J. C., Himmelblau, D. M., "Artificial Neural Network Models of Knowledge Representation in Chemical Engineering", *Comput. Chem. Eng.*, **12**, 881 (1988).
- Hoskins, J. C., Himmelblau, D. M., "Process Control via Neural Networks and Reinforcement Learning", *AIChE Annual Meeting*, Chicago II, Novembro (1990).
- Hudson, J. L., Kube, M., Adomaitis, R. A., Kevrekidis, I. G., Lapedes, A. S., Farber, R. M., "Nonlinear Signal Processing and System Identification Application to Time Series from Electrochemical Reactions", *Chem. Eng. Sci.*, **45**, 2075 (1990).
- Hunt, K. J., Sbarbaro, D., Zbikowski, R., Gawthrop, P. J., "Neural Networks for Control Systems - A Survey", *Automatica*, **28**, 1083 (1992).
- Kalman, R. E., "Design of self-optimizing control systems", *Trans. ASME*, **80**, 468 (1958) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).
- Keerthi, S. S., Gilbert, E. G., "Moving-horizon approximations for a general class of optimal nonlinear infinite-horizon discrete-time systems", *Proc. 20th Annual Conf. Information Science and Systems*, Princeton University, pp. 301 (1989) *apud* Hunt, K. J., Sbarbaro, D., Zbikowski, R., Gawthrop, P. J., "Neural Networks for Control Systems - A Survey", *Automatica*, **28**, 1083 (1992).
- Kerkhof, L. H. J., Vissers, H. J. M., "On the Profit of Optimum Control in Batch Distillation", *Chem. Engng Sci.*, **33**, 961 (1978).
- Kiparissides, C., Shah, S. L., "Self-tuning and stable adaptive control of a batch polymerization reactor", *Automatica*, **19**, 225 (1983).

Lang, P., Yatim, H., Moszkowicz, P., Otterbein, M., "Batch Extractive Distillation Under Constant Reflux Ratio", *Computers chem. Engng*, **18**, 1057 (1994).

Lee, J. H., Yu, Z. H., "Tuning of model predictive controllers for robust performance", *Comp. chem. Engng*, **18**, 15 (1994).

Leversund, E. S., Macchietto, S., Stuart, G., Skogestad, S., "Optimal Control and On-line Operation of Reactive Batch Distillation", *Comp. chem. Engng*, **18**(SUPPL), S391 (1994).

Levien, K. L., "Studies in the design and control of coupled distillation columns", Tese de Doutorado, University of Wisconsin, Madison (1985).

Levien, K. L., Morari, M., "Internal model control of coupled distillation columns", *AIChE J.*, **33**, 83 (1987).

Logsdon, J. S., Diwekar, U. M., Biegler, L. T., "On the Simultaneous Optimal Design and Operation of Batch Distillation Columns", *Trans IChemE*, **68**, Part A, 434 (1990).

Luecke, R. H., Lewis, J. C., Lin, H. Y., Yoon, W. K., "Dynamic Matrix Control of a Batch Distillation Column", *Proc. Am. Control Conf.*, Boston, 209 (1985).

Lundström, P., Lee, J. H., Morari, M., Skogestad, S., "Limitations of Dynamic Matrix Control", *Comp. chem Engng*, **19**, 409 (1995).

Martin, G. D., Caldwell, J. M., Ayril, T. E., "Predictive control applications for the petroleum refining industry", *Japan Petroleum Institute - Petroleum Refining Conf.*, Tokyo, Japan, 27-28 Outubro (1986).

Martin-Sánchez, J. M., Shah, S. L., "Multivariable Adaptive Predictive Control of a Binary Distillation Column", *Automatica*, **20**, 607 (1984).

Matsko, T. N., "Internal model control for chemical recovery", *Chem. Engng Prog.*, **81**(12), 46 (1985).

Mayne, D. Q., Michalska, H., "Receding horizon control of nonlinear systems", *Trans. IEEE on Aut. Control*, **35**, 814 (1990) *apud* Hunt, K. J., Sbarbaro, D., Zbikowski, R., Gawthrop, P. J., "Neural Networks for Control Systems - A Survey", *Automatica*, **28**, 1083 (1992).

Mayur, D. N., Jackson, R., "The Optimal Problems in Batch Distillation for Multicomponent Mixtures and for Columns with Holdup", *Chem. Engng J.*, **2**, 150 (1971).

Mehra, R. K., Rouhani, R., Eterno, J., Richalet, J., Rault, A., "Model algorithm control: review and recent development", *Engng Foundation Conf. on Chemical Process Control II*, Sea Island, Georgia, pp. 287 (1982).

- Mishkin, E., Braun, L., "Adaptive Control Systems", McGraw-Hill, NY (1961) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).
- Morningred, J. D., Paden, B. E., Seborg, D. E., Mellichamp, D. A., "An adaptive nonlinear predictive controller", *Chem. Engng Sci.*, **47**, 755 (1992).
- Mujtaba, I. M., Macchietto, S., "Optimal Operation of Multicomponent Batch Distillation - Multiperiod Formulation and Solution", *Comp. chem. Engng*, **17**, 1191 (1993).
- Murty, B. S. N., Gangiah, K., Husain, A., "Performance of Various Methods in Computing Optimal Control Policies", *Chem. Engng J.*, **19**, 201 (1980).
- Naidu, S. R., Zafiriou, E., McAvoy, T. J., "Use of Neural Networks for Sensor Failure Detection in a Control System", *IEEE Control Systems Magazine*, **49**, Abril (1990).
- Narendra, K. S., Monopoli, R. V., "Applications of adaptive control", Academic Press, NY (1980) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).
- Parrish, J. R., Brosilow, C. B., "Inferential control applications", *Automatica*, **21**, 527 (1985).
- Peterka, V., "Predictor-based self-tuning control", *Automatica*, **20**, 39 (1984).
- Powell, M. J. D., "An efficient method for finding the minimum of a function of several variables without calculating the derivatives", *Comput. J.*, **7**, 155 (1964) *apud* Willis, M. J., Montague, G. A., Di Massimo, C., Tham, M. T., Morris, A. J., "Artificial Neural Networks in Process Estimation and Control", *Automatica*, **28**, 1181 (1992).
- Press, William H., Flannery, Brian P., Teukolsky, Saul A., Vetterling, William T., "Numerical Recipes - The Art of Scientific Computing", Cambridge University Press, 1986.
- Prett, D. M., Gillette, R. D., "Optimization and constrained multivariable control of a catalytic craking unit", *AIChE National Mtg*, Houston, Texas (1979).
- Propoi, A. I., "Use of LP methods for synthesizing sample-data automatic systems", *Autumn Remote Control*, **24**, 837 (1963).
- Psichogios, D. C., Ungar, L. H., "Direct and Indirect Model-Based Control Using Artificial Neural Networks", *AIChE Annual Meeting*, Chicago IL, Novembro (1990).
- Quantrille, T. E., Liu, Y. A., "Artificial Inteligence in Chemical Engineering", Academic Press, Inc., 1991.

- Quintero-Marmol, E., Luyben, W. L., "Inferential Model-Based Control of Multicomponent Batch Distillation", *Chem. Engng Sci.*, **47**(4), 887 (1992).
- Reuter, E., Wozny, G., Jeromin, L., "Modeling of Multicomponent Batch Distillation Processes with Chemical Reaction and their Control Systems", *Computers chem. Engng*, **13**, 499, (1989).
- Richalet, J. A., Rault, A., Testud, J. L., Papon, J., "Model predictive heuristic control: applications to an industrial process", *Automatica*, **14**, 413 (1978).
- Robinson, E. R., "Microcomputers in the Process Industry", Ellis Horwood Limited, England, pp. 265 (1985).
- Robinson, E. R., "The Optimization of Batch Distillation Operations", *Chem. Engng Sci.*, **24**, 1661 (1969).
- Robinson, E. R., "The Optimal Control of an Industrial Batch Distillation Column", *Chem. Engng Sci.*, **25**, 921 (1970).
- Robinson, E. R., *Chem. Engng J.*, **2**, 135 (1971).
- Saridis, G. N., Mendel, J. M., Nikolic, Z. Z., "Report on definitions of self-organizing control processes and learning systems", *IEEE Control Systems Society Newsletter* (1973) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).
- Seborg, D. E., Edgar, T. F., Mellichamp, D. A., "Process Dynamics and Control", 272, John Wiley & Sons, New York, NY (1989).
- Shen, G. C., Lee, W. K., "A predictive approach for adaptive inferential control", *Comp. chem. Engng*, **13**, 687 (1989).
- Shepherd, G. M., "The Synaptic Organization of the Brain", 2nd ed., Oxford University Press, New York, 1979.
- Shinskey, F. G., "Process Control Systems - Application, Design and Tuning", 3rd. Edition, pp. 485 (1979).
- Sistu, P.B., Gopinath, R. S., Bequette, B. W., "Computational issues in non-linear predictive control", *Comp. chem. Engng*, **17**, 361 (1993).
- Stephanopoulos, G., "Chemical Process Control - An Introduction to Theory and Practice", Prentice-Hall, Inc. (1984).
- Su, H.T., McAvoy, T., Werbos, P., "Long-term predictions of chemical processes using recurrent neural networks: a parallel training approach", *Ind. Eng. Chem. Res.*, **31**, 1338 (1992).

- Taylor, L. W., Adkins, E. J., "Adaptive control and the X-15", *Proc. Princeton University Conference on Aircraft Flying Qualities*, Universidade de Princeton (1965) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).
- Truxal, J. G., "Theory of self-adjusting control", *Proc. 2nd. IFAC World Congress* (1964) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).
- Tsyppkin, Ya. Z., "Adaptation and Learning in Automatic Systems", Academic Press, NY (1971) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).
- Unbehauen, H., "Methods and Applications in Adaptive Control", Springer, Berlin (1980) *apud* Aström, K. J., "Theory and applications of adaptive control - A survey", *Automatica*, **19**, 471 (1983).
- Whitely, J. R., Davis, J. F., "Back-Propagation Neural Networks for Qualitative Interpretation of Process Sensor Data", *AICHE Annual Meeting*, Chicago IL, Novembro (1990).
- Willis, M. J., Montague, G. A., Di Massimo, C., Tham, M. T., Morris, A. J., "Artificial Neural Networks in Process Estimation and Control", *Automatica*, **28**, 1181 (1992).
- Yao, S. C., Zafiriou, E., "Control System Sensor Failure Detection via Networks of Localized Receptive Fields", *American Control Conf.*, San Diego, CA, Junho (1990).
- Ydstie, B. E., "Extended horizon adaptive control", *9th World Congress of the IFAC*, Budapest, Hungary (1984).
- Zadeh, L.A., Whalen, B. H., "On optimal control and linear programming", *IRE Trans. Aut. Control*, **7**(4), 45 (1962).

ABSTRACT

Separation processes are very important in chemical industries where batch distillation appears as one of its oldest processes. Over the last few years batch processing has received renewed attention because of the growing interest in the production of high-technology, high-value added chemical and biochemical products and also in environmental problems of industrial waste minimization.

Based on similar phenomenological aspects, both batch and continuous distillation are characterized as non-linear processes, but due to the absence of feed streams in batch equipments the temperature and composition profiles become instantaneous, with substantially different control systems requirements. In order to control top composition, by handling the reflux ratio, digital control systems taking into account the unsteady process characteristics were developed where design did not follow the conventional feedback controllers fixed tuning. Reboiler heating requirement is adjusted through an independent control loop by measuring bottom composition or column pressure differential, therefore minimizing the energy consumption.

An ordinary way of adjusting the controller parameters was initially studied: the Programmable Adaptive Controller (PAC). The adjustments are previously scheduled and on-line performed through measurements of an auxiliary variable which correlates well with the changes in process dynamics. The PAC design is strongly dependent on the knowledge of the fluid process properties and on the involved process phenomena.

In practice, however, where complex mixtures are usually processed, the independence on the referred knowledge about the mixture is crucial and claims an on-line process identification procedure. Still on Adaptive Control class and with the above purpose, a Self-tuning Regulator was implemented where a Regressive Analysis found the coefficients of the discrete polynomial input/output relationship of the process. Two different control laws were used: PI (position or velocity form) and another one derived from Dahlin's Algorithm.

An Artificial Neural Network Model Based Predictive Control was also implemented to find an phenomenological independent process modeling, with the main feature relying on the non-linear approach obtained from the neural network model. The Artificial Neural Network is trained every sampling interval and provides process behavior predictions over certain time horizon. The more important process state variables define the number of artificial neurons to be used on the input and output network layers. This controller has the important generic design characteristic and can be used not only in Regulatory mode but also in Supervisory Control.

Through closed-loop processes simulations with different ternary mixtures processed, a global analysis was performed based on main results obtained from the proposed controllers implementation. The Feedforward Artificial Neural Network, trained through Backpropagation, proved to be a potential alternative to represent the process and the Model Predictive Control showed the more efficient control action in order to keep top composition, reducing the batch time and saving energy.