

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

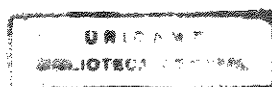
**CONTROLE ADAPTATIVO DE UMA
COLUNA PILOTO DE DESTILAÇÃO EM BATELADA
COM INFERENCIAÇÃO DE COMPOSIÇÃO ATRAVÉS DE
REDES NEURAIS ARTIFICIAIS**

Luciano da Silva Pedrosa

Orientadora: Ana Maria Frattini Fileti

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

Campinas – Agosto/1998



Esta versão corresponde a redação final da Dissertação de Mestrado em Engenharia Química defendida pelo Engenheiro Luciano da Silva Pedrosa e aprovada pela Comissão Julgadora em 28/08/98.

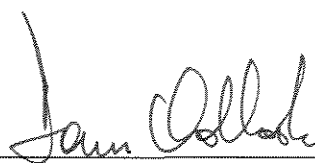


Prof. Dra. Ana Maria Frattini Fileti
Orientadora

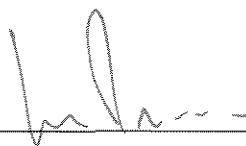
Dissertação de Mestrado defendida e aprovada em 28 de agosto de 1998 pela banca examinadora constituída pelos professores doutores:



Prof. Dra. Ana Maria Frattini Fileti
(Orientador)



Prof. Dr. Darci Odloak



Prof. Dr. João Alexandre Ferreira da Rocha Pereira

À Simone Maria Coimbra Peixoto
uma pessoa muito especial.

AGRADECIMENTOS

A prof^a. Dr^a. Ana Maria Frattini Fileti pela orientação.

Ao prof^o. Dr. João Alexandre F. R. Pereira pela ajuda na montagem experimental.

Aos colegas do DESQ pelas discussões técnicas durante o desenvolvimento deste trabalho.

Aos meus pais pelo incentivo.

A CAPES pelo auxílio financeiro que permitiu a realização deste trabalho.

RESUMO

A destilação em batelada é um processo de separação bastante utilizado na purificação de produtos químicos ou bioquímicos de alta tecnologia e/ou alto valor agregado. A maior vantagem das colunas em batelada é sua versatilidade, pois podem processar diversas misturas sob diferentes condições experimentais. Esta flexibilidade juntamente com a não-linearidade e a natureza não estacionária do processo tornam as técnicas convencionais de controle inadequadas, constituindo um desafio no projeto de sistemas de controle. Além disso, a busca da obtenção de produtos com pureza especificada e a necessidade de minimizar produtos fora de especificação mostram claramente a importância do projeto, do desenvolvimento e da implementação de sistemas de controle para o monitoramento destes processos.

Este trabalho apresenta o desenvolvimento e a implementação experimental de um sistema de controle digital direto para uma coluna piloto de destilação em batelada. Estratégias de controle "feedback" convencionais foram primeiramente implementadas no microcomputador 486 DX 33MHz acoplado a coluna, mostrando um comportamento insatisfatório para o processo descontínuo em questão.

Diante das exigências do processo, posteriormente optou-se pela implementação de algoritmos de controle adaptativo. Um algoritmo de controle auto-ajustável (STR) baseado em identificação recursiva de parâmetros do processo foi desenvolvido. O processo é representado por um modelo discretizado de uma função transferência de primeira ordem com atraso de transporte. O método de Programação Quadrática Sucessiva aplicado a mínimos quadrados, da biblioteca matemática NAG, foi utilizado para a estimação "on-line". O equivalente digital de um controlador PI e o algoritmo de Dahlin foram individualmente testados como leis de controle. Tais controladores foram então comparados com um Controlador Adaptativo Programável (PAC). O projeto e o desempenho do PAC mostraram-se altamente dependentes da qualidade da mistura a ser destilada enquanto o STR atuou de maneira adequada para todas as separações efetuadas, apesar de seu projeto genérico.

Para que os valores das composições de topo e fundo pudessem ser inferenciados a partir de medições de temperatura na coluna, foram utilizados dados de equilíbrio líquido-vapor para a separação do sistema binário (n-hexano/n-heptano) e inferenciação via Redes Neurais Artificiais (RNA) para a separação do sistema ternário (n-hexano/cicloexano/n-heptano).

O sistema de controle desenvolvido tornou o equipamento de destilação em batelada convencional mais eficiente e fácil de operar. Os experimentos realizados sobre uma coluna piloto confirmaram os resultados de simulações de trabalhos anteriores.

ÍNDICE

Capítulo 1 – INTRODUÇÃO	1
1.1 Introdução	2
1.2 Motivação pelo estudo do tema	2
1.3 Organização da dissertação	3
Capítulo 2 – REVISÃO BIBLIOGRÁFICA	5
2.1 Revisão bibliográfica	6
2.2 Proposta da dissertação	12
Capítulo 3 – MONTAGEM EXPERIMENTAL	15
3.1. Introdução	16
3.2. Coluna de destilação	16
3.3. Interface processo/microcomputador	16
3.3.1. Placa analógica-digital-analógica (ADA)	16
3.3.2. Conversor analógico-digital	17
3.3.3. Conversor digital-digital	18
3.3.4. Conversor digital-analógico	19
3.4. Válvula Magnética	20
3.5. Sistema de automação e controle	20
Capítulo 4 – IMPLEMENTAÇÃO DAS ESTRATÉGIAS DE CONTROLE	22
4.1 Introdução	23
4.2 Controle convencional	23
4.3 Controle adaptativo	25
4.3.1 Controlador adaptativo programável (PAC)	25
4.3.1.1 Implementação do PAC	26
4.3.2 Controlador auto-ajustável (STR)	28
4.3.2.1 Implementação do STR	29
4.3.2.2 Identificação do processo	30
4.3.2.3 Projeto do controlador	31
4.4 Resultados	33

4.4.1	Controle convencional	34
4.4.2	Controle adaptativo	37
4.4.2.1	Efeito da dinâmica do processo	38
4.4.2.2	Teste de estabilidade	39
4.4.2.3	Efeito do intervalo de amostragem	40
4.4.2.4	Efeito do parâmetro μ	41
4.4.2.5	Comparação entre os controladores	42
4.4.2.6	Sistema ternário	44
Capítulo 5	– INFERENCIAÇÃO POR REDES NEURAIS ARTIFICIAIS	46
5.1	Introdução	47
5.2	Neurônios artificiais ou nódulos	47
5.2.1	Estado de ativação	48
5.2.2	Função de saída ou função de ativação	49
5.3	Topologia das RNA's	50
5.4	Treinamento das RNA's	51
5.4.1	Algoritmo de retropropagação (ou "Backpropagation")	52
5.4.2	Algoritmo de Levenberg-Marquardt	54
5.5	Inferenciação	55
5.6	Simulador	57
5.7	Implementação	60
5.8	Resultados	61
Capítulo 6	– CONCLUSÕES E SUGESTÕES FINAIS	65
6.1	Conclusões	66
6.2	Sugestões para futuros trabalhos	68
Anexo A	– OBTENÇÃO DO MODELO DISCRETO DO PROCESSO	69
Anexo B	– MÉTODO DE PROGRAMAÇÃO QUADRÁTICA SUCESSIVA	72
Anexo C	– PROGRAMA COM O SISTEMA DE CONTROLE DESENVOLVIDO	76
REFERÊNCIAS BIBLIOGRÁFICAS		104
ABSTRACT		109

NOMENCLATURA

A_p = área do prato;

a_j = atividade do neurônio artificial (cap. 5);

A_j = atividade total do neurônio;

a_1, b_1, c_1, d_1 e d_2 = coeficientes ajustáveis das equações polinomiais discretas do processo e do controlador STR/Dahlin (cap. 4);

b_j = bias do neurônio j (cap. 5);

bias = bias do modelo do processo (cap. 4);

D = fluxo de Destilado;

$D(z)$ = função transferência do controlador (cap. 4);

E = erro entre a variável controlada e o valor de referência;

E_j = eficiência de Murphree no estágio j ;

e = constante da equação de Francis (cap. 5);

e_i = erro entre a variável controlada e o valor de referência no instante atual (cap. 4);

e_q = erro entre o valor real e aquele predito pela RNA (cap. 5);

$f^k(\)$ = forma funcional de cálculo interno do neurônio artificial na iteração k (cap. 5);

$f_1(\), f_2(\), f_3(\), f_4(\), f_5(\)$ = funções da variável auxiliar na implementação do PAC (cap. 4);

$f_i^0 = \phi_i^s P_i^s$ = fugacidade de referência do componente i puro à temperatura e pressão de equilíbrio (cap. 5);

F = função a ser minimizada pelo método SQP (cap. 4);

$\dot{F}^K$ = matriz diagonal das primeiras derivadas (cap. 5);

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

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 ;

$(\Delta H)_0$ = calor latente de vaporização da mistura presente em V_1 ;

$(\Delta H)_{N+1}$ = calor latente de vaporização da mistura presente no refeedor;

IA = número de amostragens passadas usadas na análise regressiva (cap. 4);

K_c = ganho da lei de controle PI;

K = ganho do processo;
 L_w = altura do vertedouro;
 L_j = fluxo de líquido que deixa o estágio j ;
 L = vetor dos limites inferiores das variáveis no método SQP;
 M_j = acúmulo molar de líquido no estágio j ;
 nd = parte inteira da divisão θ/T ;
 N_p = número de pontos disponíveis para o treinamento da RNA;
 P = pressão de operação (cap. 5);
 P_i^s = pressão de saturação calculada pela equação de Antoine;
 $\vec{p}_q$ = vetor dos valores preditos pela RNA;
 Q_c = fluxo de calor retirado no condensador;
 Q_R = fluxo de calor fornecido ao refeedor;
 $\vec{r}_q$ = vetor dos dados reais para o treinamento da RNA;
 S = vetor dos limites superiores das variáveis no método SQP;
 S_k = número de neurônios na camada k ;
 t = tempo de operação;
 T = intervalo de amostragem;
 T_{fundo} = temperatura de fundo da coluna;
 T_{topo} = temperatura de topo da coluna;
 $T_{eb1}, T_{eb2}, T_{eb3}$ = temperaturas de ebulição dos componentes da mistura;
 u_i = variável manipulada no instante atual;
 u_{ss} = variável manipulada no estado estacionário;
 V_j = fluxo de vapor que deixa o estágio j (cap. 5);
 $w_{i,j}$ = peso da conexão do neurônio i sobre o neurônio j ;
 W^{k+1} = matriz dos pesos para a camada $k+1$;
 W^k = matriz hessiana da função de Lagrange (anexo B);
 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. 5);

x_B = fração molar de fundo;

$y_{i,j}$ = fração molar do componente i na fase vapor do estágio j (cap. 5);

$y_{i,j}^*$ = fração molar de equilíbrio do componente i na fase vapor do estágio j (cap. 5);

y = variável controlada (composição de topo);

$\tilde{y}$ = variável controlada amostrada;

$\hat{y}$ = variável controlada predita (cap. 4).

y_{sp} = valor de referência da variável controlada;

y_{ss} = variável controlada no estado estacionário;

Letras Gregas

α = taxa de aprendizado da RNA;

δ = termo de declínio do gradiente para o treinamento da RNA (cap. 5);

δ = delta do subproblema do SQP (anexo B);

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 = coeficiente de “momentum”;

λ = multiplicadores de Lagrange;

μ = constante de tempo no algoritmo de Dahlin (cap. 4);

μ = parâmetro do algoritmo de Levenberg-Marquardt (cap. 5);

θ = tempo morto do processo;

ρ_j = massa específica molar da mistura líquida no prato j ;

τ_I = constante de tempo integral da lei de controle PI;

τ_D = constante de tempo derivativo da lei de controle PID;

τ = constante de tempo do processo.

CAPÍTULO 1

INTRODUÇÃO

1.1 - INTRODUÇÃO

Os processos de separação são amplamente utilizados nas indústrias químicas merecendo um lugar de destaque nestas e na literatura de engenharia química. Um dos mais importantes e utilizados destes processos é a destilação. Basicamente, a destilação pode ser dividida quanto ao modo de operação em contínua ou em batelada. O processo em batelada, embora seja mais antigo e tenha sido considerado ultrapassado com o surgimento das colunas de destilação contínuas, tem se mostrado mais eficiente em algumas situações justificando, assim, a atenção especial que tem recebido nos últimos anos.

As colunas de destilação em batelada são usadas mais frequentemente quando pequenas quantidades de produtos químicos ou bioquímicos de alta tecnologia e alto valor agregado estão sendo separadas. Seu uso também tem sido empregado na redução de volumes residuais e recuperação destes materiais. Além disso, a destilação em batelada pode ser empregada no processamento de misturas contendo sólidos ou substâncias que se tornam altamente viscosas quando concentradas uma vez que estas permanecem no recipiente de carregamento no fundo da coluna e podem ser devidamente retiradas ao término da batelada.

Outra aplicação industrial que tem sido dada a destilação em batelada consiste no acoplamento de uma coluna de retificação sobre um reator tanque para promover uma imediata separação do produto principal mais volátil ou para promover o aumento da temperatura no reator, podendo assim aumentar a velocidade de reação.

O número de produtos obtidos numa coluna de destilação em batelada é outra grande vantagem deste processo. Contudo, o crescente interesse nestas colunas deve-se, principalmente, a sua versatilidade pois pode processar diversas misturas sob diferentes condições experimentais.

1.2 - MOTIVAÇÃO PELO ESTUDO DO TEMA

A modelagem dinâmica fenomenológica e a determinação de razões de refluxo ótimas em colunas de destilação em batelada têm sido extensivamente estudadas. Entretanto, misturas complexas das quais se conhece apenas a presença de um determinado componente, sendo os demais desconhecidos, tornam difícil a otimização da operação deste processo.

Já os estudos relacionados com a operação em linha de tal processo têm sido pouco pesquisados uma vez que a inexistência de uma alimentação contínua na coluna acarreta perfis

instantâneos de composição e temperatura ao longo do equipamento, limitando a utilização de sistemas de controle convencionais nestes processos.

Por outro lado, sendo a destilação em batelada um processo energético bastante intenso, mesmo uma modesta redução no tempo da batelada pode corresponder a uma redução significativa nos custos operacionais. Além disso, produtos de uma dada pureza podem ser obtidos se a razão de refluxo for continuamente ajustada enquanto o produto desejado é retirado da coluna. Sendo assim, fica evidente a importância do desenvolvimento e da implementação de sistemas de controle nestes processo para a obtenção de tais produtos e a minimização de produtos fora de especificação.

Entretanto, para a implementação de sistemas de controle nestes equipamentos, torna-se comumente necessário a medição em linha da composição de topo e/ou fundo. Buscando contornar os altos custos e atrasos associados a estas medidas, muitos sistemas de inferenciação tem sido propostos na literatura. Contudo, a maioria deles consideram os perfis de composição no topo de uma coluna de destilação em batelada como essencialmente binários, independentemente do número de componentes na mistura inicial. Esta hipótese facilita a solução do problema de inferência, mas não pode ser aplicada a qualquer condição operacional da separação multicomponente. Deste modo, inferenciadores mais robustos devem ser propostos visando eliminar todos os fatores acima descritos.

Sendo assim, mostra-se justificada a realização de um estudo detalhado de sistemas de controle avançado e de inferenciação para colunas de destilação em batelada.

1.3 - ORGANIZAÇÃO DA DISSERTAÇÃO

No atual capítulo a importância do processo de destilação em batelada foi exposta juntamente com os principais motivos que levaram ao estudo do tema proposto.

No segundo capítulo, será dada uma visão geral sobre os diversos estudos na área de destilação em batelada buscando-se uma perfeita compreensão de onde está situado o presente estudo. Serão discutidas também as estratégias de operação e controle propostas para as colunas de destilação em batelada. Por último, diante dos trabalhos existentes expõe-se a proposta desta dissertação.

No terceiro capítulo, será apresentada a coluna de destilação em batelada, a placa conversora Analógico-Digital-Analógico e os demais dispositivos utilizados na malha de controle.

O quarto capítulo mostra o desenvolvimento e a implementação em linha do Controlador Adaptativo Auto-ajustável (STR) proposto. Neste controlador são utilizadas duas leis diferentes de controle, o PI digital e a resultante do Algoritmo de Dahlin. Neste capítulo também são expostos os controladores convencionais P, PI e PID digitais e o Controlador Adaptativo Programável (PAC) para posterior comparação com o sistema de controle proposto. Finalizando o capítulo são mostrados os resultados alcançados para os diferentes controladores.

O quinto capítulo trata da implementação do Sistema de Inferenciação baseado nas Redes Neurais Artificiais (RNA's) necessário para a medida em linha da composição de topo da coluna, buscando a extensão do sistema de controle proposto a misturas multicomponentes. Também são expostas as principais características das RNA's e dos procedimentos de treinamento das mesmas.

No sexto e último capítulo, as conclusões finais do estudo são apresentadas juntamente com algumas sugestões para trabalhos futuros.

CAPÍTULO 2
REVISÃO BIBLIOGRÁFICA

2.1 - REVISÃO BIBLIOGRÁFICA

Apesar de ser governada pelos mesmos princípios de equilíbrio líquido-vapor, hidráulica dos pratos e dinâmica das colunas contínuas, a destilação em batelada apresenta um comportamento intrinsecamente transiente uma vez que não existe alimentação contínua da mesma. Devido a este carácter transitório, sua operação apresenta muitas variáveis e, consequentemente, diversas maneiras de se combinar as condições operacionais. Isto proporciona um vasto campo para estudos envolvendo modelagem, simulação, otimização, operação e controle deste processo.

A operação de uma coluna de destilação em batelada pode ser dividida basicamente em duas etapas bem distintas. A primeira (partida ou "start-up") consiste na carga da mistura no refeedor e no aquecimento da mesma em refluxo total até que um estado estacionário seja atingido, no qual os perfis de temperatura e composição estão estabelecidos. O procedimento mais adequado de partida é previamente escolhido de acordo com a estrutura disponível e do sistema a ser destilado.

Já a segunda etapa consiste na destilação propriamente dita. Neste momento, a razão de refluxo é alterada de modo a retirar o produto mais volátil que se encontra num estado de maior pureza no topo da coluna. A composição deste produto coletado tende a acompanhar as variações de composição do material presente na coluna e depende fortemente do refluxo de líquido que retorna a mesma.

O progresso da destilação em batelada pode ser controlado de diversas formas. As principais são: operação a refluxo constante, operação a composição constante de destilado e operação segundo o planejamento ótimo da razão de refluxo.

A operação a refluxo constante apresenta-se como o modo mais simples de conduzir a destilação em batelada. Neste modo operacional produz-se um destilado de composição variável começando num patamar mais elevado que o especificado e terminado num valor abaixo do desejado, contudo, a composição média do produto final acumulado deverá estar dentro da especificação.

Na operação à composição constante de destilado, a razão de refluxo varia de modo a enriquecer o produto à medida que o componente desejado vai se esgotando. Assim, pode-se iniciar a batelada com uma razão de refluxo baixa e terminar com um valor mais elevado.

Já a operação segundo um planejamento ótimo da razão de refluxo segue um critério ótimo de desempenho: tempo mínimo para a produção de determinada quantidade de produto dentro do especificado, produção máxima dentro da especificação em tempo mínimo ou produção visando lucro máximo.

Em colunas cuja instrumentação é precária ou inexistente, pode-se ainda utilizar o modo de operação cíclica alternando períodos com refluxo total e refluxo nulo.

Por outro lado, todos os modos operacionais citados acima, podem alterar, além da razão de refluxo, o calor fornecido a mistura no refeedor buscando-se manter um fluxo fixo de vapor que deixa o mesmo. Isto evita desperdício de energia uma vez que fluxos elevados de vapor tornam necessária uma alta razão de refluxo para purificação do destilado.

Luyben (1988) estudou o efeito do reciclo das misturas intermediárias formadas na operação de uma coluna de destilação em batelada multicomponente usando como critério de análise a metodologia do fator de capacidade proposta por Luyben (1971). Neste trabalho, todas as misturas intermediárias foram adicionadas a uma nova carga a ser destilada. Os modos de operação a refluxo constante e refluxo variável foram testados tendo o primeiro modo obtido um melhor desempenho.

Em Quintero-Marmol e Luyben (1990) novas estratégias para o reciclo destas misturas foram testadas em sistemas ternários de diferentes volatilidades relativas operando a coluna a refluxo constante. Foi observado que para purezas elevadas do destilado (95-99%) a melhor estratégia consistia em armazenar as misturas intermediárias em tanques separados e após reunidas quantidades adequadas das mesmas processá-las individualmente. Foi alcançada uma melhora de 38% sobre a estratégia usada em Luyben(1988).

Mujtaba e Macchietto (1993) e Bonny (1995) também estudaram estratégias de reciclo da misturas intermediárias através de otimizações não-lineares. A razão de refluxo foi discretizada usando valores constantes para cada período de operação da coluna. Eles observaram que a dificuldade de implementação da razão de refluxo variável neste tipo de problema está na resolução das equações algébricas e diferenciais existentes no modelo do processo e no problema de otimização.

Shinskey (1979) mostrou as seguintes desvantagens na operação a refluxo constante: se a razão de refluxo for mantida constante num valor baixo, a separação será pobre e a retirada

do destilado será interrompida mesmo existindo uma grande concentração do componente de interesse no refeedor; se a razão de refluxo tiver um valor elevado a separação será melhor, mas, o consumo de energia e, conseqüentemente, o custo de operação aumentarão bastante.

Offers et al. (1995) buscando um consumo mínimo de energia propuseram um novo método para o cálculo da razão de refluxo mínima tanto para destilação contínua quanto numa destilação em batelada multicomponente com concentração constante do destilado. Neste estudo, além da composição na alimentação (ou carga inicial da batelada) era necessário o conhecimento do equilíbrio de fase. Entretanto isto nem sempre é possível como, por exemplo, na recuperação de resíduos cujos componentes que constituem a mistura não estão completamente determinados.

Converse e Gross (1963) resolveram o problema da máxima quantidade de produto de uma dada pureza dentro de um tempo fixo através de um método de programação dinâmica para sistemas binários com acúmulo de líquido desprezível nos pratos. Eles compararam seus resultados com aqueles obtidos através dos métodos convencionais, composição constante e refluxo constante, tendo encontrado que a trajetória ótima proporciona um aumento de produtividade de cinco por cento em relação aos outros métodos.

Coward (1967) e Price (1967) alcançaram os mesmos resultados que Converse e Gross (1963) utilizando o Princípio do Máximo Contínuo de Pontryagin.

Já Robinson (1969), utilizando o Princípio do Máximo Contínuo de Pontryagin para resolução do problema de tempo mínimo em destilação em batelada multicomponente, comparou a operação com composição constante de destilado e a refluxo constante observando que a primeira está mais próxima da operação ótima. Além disso, constatou uma maior facilidade na implementação da razão de refluxo variável em misturas de volatilidades relativas bastante distintas.

Robinson (1970) observou que os acúmulos nos pratos, mesmo que pequenos, não podem ser desprezados na modelagem da destilação em batelada e mostrou seu efeito numa coluna industrial usada para separação de terpenos. Neste trabalho evidencia-se, também, as dificuldades de extensão do método de otimização a sistemas multicomponentes tendo em vista o aumento do número de equações a serem resolvidas exigindo um maior esforço computacional.

Fahat et al. (1990) buscando resolver o problema de obtenção da máxima quantidade de destilado em misturas binárias e multicomponentes, observaram que a solução ótima pode ser obtida operando a coluna a composição constante de destilado durante a remoção dos componentes no topo e a refluxo constante na retirada das misturas intermediárias. Neste trabalho também foi verificado intenso esforço computacional na resolução das equações algébricas e diferenciais do modelo do processo.

Diwekar et al. (1987) propuseram um método reduzido na resolução do problema de destilado máximo como solução para o aumento de tempo computacional. Contudo, comparações realizadas com os dados de Murty et al. (1980) mostram desvios significativos entre os modelos rigoroso e reduzido do processo.

Diwekar (1991) utilizou um método reduzido para o projeto de uma coluna de destilação em batelada com azeotrópo. Neste caso, os resultados mostraram boa concordância com o modelo rigoroso do processo.

Yu et al. (1989), Britt e Rosen (1989) e Jang (1990) afirmam em seus trabalhos que a combinação dos modos de operação a refluxo constante e a composição constante é perfeitamente capaz de otimizar o processo de destilação em batelada. E sugerem o uso de simulações dinâmicas como meio de obtenção desta combinação.

Kerkhof e Vissers (1978) maximizando uma função lucro que levava em consideração as quantidades produzidas, o tempo de partida, os custos dos produtos e o tempo de batelada, obtiveram um planejamento de refluxo ótimo o qual foi comparado às técnicas convencionais de operação. Eles observaram que em separações difíceis a operação ótima aproximava-se mais da operação com refluxo constante do que com composição constante.

Por muitos anos, os trabalhos em destilação em batelada focalizaram-se apenas em estratégias de otimização do processo. Contudo, como visto acima, em muitos casos a diferença entre estas e os modos de operação convencionais era pequena. Mais recentemente, novas configurações para a coluna e novas estratégias de controle tem sido sugeridas para este processo buscando-se sempre o aumento de produtividade.

As novas configurações diferem basicamente na posição do principal acúmulo de líquido na coluna. Assim, além da coluna convencional (ou regular) surgiram as colunas

invertidas, as colunas com um vaso intermediário e aquelas com vários vasos separando diversos seguimentos do equipamento ("multi vessel distillation").

Sorensen e Skogestad (1996) compararam a destilação em batelada com as configurações regular e invertida em termos da operação ótima e do comportamento dinâmico. Neste estudo observou-se que as colunas invertidas apresentavam um menor tempo de operação para separações de misturas contendo pequenas quantidades do componente mais volátil. Contudo, a principal vantagem encontrada nesta nova configuração consiste na obtenção dos componente leves com elevada pureza no topo da coluna.

Davidyan et al. (1994) estudaram a destilação em batelada em colunas com um vaso intermediário. Neste tipo de equipamento a mistura é carregada no vaso e os produtos são simultaneamente retirados no topo e no fundo da coluna enquanto a mistura remanescente aumenta sua pureza em componentes intermediários. Eles então compararam com a coluna convencional e enumeraram as seguintes vantagens da nova coluna: misturas binárias podem ser separadas em seus componentes puros mais rápido do que no modo convencional; já misturas multicomponente podem ser separadas em frações pesadas, intermediárias e leves simultaneamente, podendo-se remover impurezas leves e pesadas da mistura.

Hasebe et al. (1995) e Skogestad et al. (1997) estudaram o controle de colunas em batelada com vários vasos intermediários. Hasebe et al. propuseram um esquema baseado em controladores de nível. Já Skogestad et al. sugeriram uma estrutura de controle "feedback" baseado em controladores de temperaturas. Contudo, nos dois trabalhos nota-se que o controle deste novo equipamento é bem mais complicado em relação ao da coluna convencional.

Tanto a configuração convencional quanto as novas sugerem que técnicas mais avançadas de controle devem ser aplicadas às colunas de destilação em batelada para operações mais seguras e rentáveis.

Sorensen (1998) descreve uma vasta bibliografia para as novas configurações das colunas de destilação em batelada e compara as estratégias de operação com refluxo ótimo e operação cíclica para colunas convencionais e invertidas. As colunas convencionais apresentam um menor tempo de operação quando misturas com composição inicial de 50 a 90% do componente mais volátil são destiladas segundo um refluxo ótimo e em misturas de 5 a 10% do mesmo componente seguindo uma operação cíclica.

Fileti (1992) estudou estratégias de controle em destilação em batelada através de simulações dinâmicas do processo. Neste estudo, o controlador adaptativo programável foi usado para controlar a composição do destilado no topo da coluna tendo alcançado um desempenho bastante superior às técnicas convencionais de controle. A técnica de controle usada aplica-se a sistemas binários e multicomponentes. Entretanto, a estimação adequada dos parâmetros do processo foi feita "off-line" e exigiu bastante tempo e esforço computacional.

Cunha (1996), num trabalho experimental, utilizou o controlador adaptativo programável para o modo de operação a composição constante do destilado para uma mistura binária n-hexano/n-heptano. As composições de topo e fundo foram inferenciadas através do conhecimento das relações de equilíbrio líquido-vapor da mistura. O sistema de controle mostrou-se bastante eficaz. Contudo, o caráter "off-line" da estimação de parâmetros exige novas simulações todas as vezes que as condições operacionais da coluna forem modificadas.

Fileti (1996), através de simulações, utilizou o controle adaptativo e preditivo por Redes Neurais Artificiais (RNA) numa coluna de destilação em batelada para o modo de operação a composição constante de destilado, sendo a quantidade de calor fornecida no refeedor ajustada para manter a vazão de vapor constante no equipamento. Neste trabalho, foram comparados os controladores adaptativos programável e auto-ajustável bem como o controle preditivo. Além disso, no controlador auto-ajustável foram utilizadas como metodologias de controle o equivalente digital de um controlador PI, tanto na forma posição quanto na forma velocidade, e o algoritmo de Dahlin. Tendo-se notado um melhor desempenho do controlador auto-ajustável quando comparado ao programável.

Em quase todos os trabalhos citados a composição no topo da coluna é a variável a ser controlada. Entretanto, para a implementação em linha de sistemas de controle faz-se necessário a medição desta variável.

O método mais tradicional para a medição da composição de topo na coluna é através de análise cromatográfica. Contudo, o alto custo e o atraso inerente a estas análises tornam inadequada, na maioria dos casos, sua implementação em sistemas de controle em linha.

Bosley e Edgar (1994), na verificação experimental de um modelo dinâmico do processo de destilação em batelada, utilizaram um densímetro do tipo Anton Paar para inferenciação em linha de composição através de medidas de densidade e temperatura. Contudo, seu uso está limitado a alguns tipos de misturas apolares.

Quintero-Marmol e Luyben (1992) desenvolveram dois sistemas de inferenciação de composição em colunas de destilação em batelada. O primeiro para o estado estacionário ("start-up") negligenciando os acúmulos de líquido na coluna e no vaso de refluxo, que utiliza rigorosas equações não-lineares e medidas de temperatura e fluxo de destilado. O segundo inferenciador usando um modelo não-linear quasi-dinâmico considera os perfis de composição na parte superior da coluna como essencialmente binários. Isto possibilita o uso deste tipo de inferenciação em separações multicomponentes.

Morris et al. (1991) e Morris et al. (1994) sugerem as Redes Neurais Artificiais (RNA) como valiosas ferramentas na inferenciação de composição, entre outras variáveis que não podem ser diretamente medidas ou quando existe um atraso de medida elevado em relação a dinâmica do sistema. Eles mostram que o uso das RNA's na inferenciação da biomassa no processo de fermentação da penicilina apresenta excelentes resultados.

Zhang et al. (1997) utilizam as RNA's para a inferenciação do peso molecular médio na polimerização em solução via radical livre do metil metacrilato, com água como solvente e peróxido de benzoíla como iniciador. Diversos experimentos foram realizados sob diferentes condições para o teste da inferenciação proporcionada pela RNA, tendo-se obtido bons resultados.

Ponton e Klemes (1993) comparando diversas alternativas de inferenciação de medidas de composição mostraram que funções do tipo razão de polinômios também apresentam bons resultados. Mas, concluem que modelos fenomenológicos são mais adequados quando se pretende estimar valores fora da faixa comum de operação, enquanto que as RNA's exibem melhores resultados dentro desta faixa. Por outro lado, tendo em vista os poucos parâmetros de ajuste necessários as RNA's são facilmente implementadas.

2.2 - PROPOSTA DA DISSERTAÇÃO

O crescente interesse no processo de destilação em batelada exige a abordagem dos problemas práticos relacionados com o controle e a automação deste processo, além daqueles associados ao seu projeto. Sendo a maior vantagem das colunas em batelada a flexibilidade de manipulação das mais variadas e complexas misturas, não se justifica um projeto otimizado destes equipamentos. A importância do projeto torna-se evidente na escolha da configuração da coluna visto que tanto os fatores econômicos quanto as características da mistura devem

ser considerados. Contudo, uma vez definida a configuração básica do equipamento, a escolha do sistema de controle torna-se o problema chave a ser resolvido na operação deste processo.

Diante do comportamento altamente transiente e não linear da destilação em batelada, os controladores "feedback" convencionais exibem resultados insatisfatórios visto que estão baseados em parâmetros fixos. Diante disso, algoritmos de controle adaptativo surgem como uma alternativa mais adequada no controle deste processo. Com este intuito, Fileti (1992) e Cunha (1996) utilizaram um controlador adaptativo programável ("Programmable Adaptive Control"). Entretanto, a implementação deste controlador mostrou-se fortemente dependente do conhecimento do material a ser processado e exige um bom conhecimento do sistema, uma vez que são necessárias exaustivas simulações do mesmo.

Buscando o desenvolvimento e a implementação de um sistema de controle independente das propriedades do material a ser processado e de um amplo conhecimento do processo, o presente trabalho propõe o desenvolvimento e a implementação de um sistema de controle auto-ajustável ("Self Tuning Regulator") para uma coluna de destilação piloto. Neste tipo de controlador todo o trabalho reside na identificação recursiva do processo realizada em tempo real.

Propõe-se, também, a utilização de outras estratégias de controle (como as "feedback" convencionais e o PAC) para posterior comparação de desempenho. Estas estratégias devem ser implementadas individualmente num microcomputador 486 DX 33MHz acoplado a coluna, consistindo numa aplicação de Sistemas de Controle Digital Direto (DDC).

Como muitos outros processos industriais, destilação em batelada multicomponente é difícil de controlar não somente por seu comportamento não-linear e transiente, como também pela dificuldade na medição rápida e da qualidade do produto. O atraso introduzido pela análise de composição pode reduzir a eficiência do controle.

Como pode ser constatado na revisão da literatura, vários sistemas de inferenciação tem sido propostos (Ponton e Klemes, 1993; Quintero-Marmol e Luyben, 1992; Yu e Luyben, 1987) para superar os altos custos e atrasos associados as medidas de composição "on-line". A maioria deles, contudo, considera os perfis de composição no topo da coluna de destilação em batelada como essencialmente binários, independentemente do número de componentes na mistura inicial. Esta hipótese torna o problema da inferência mais fácil de ser resolvido,

através de dados de equilíbrio líquido-vapor, mas não pode ser usados em quaisquer condições da separação multicomponente.

Os trabalhos de Morris et al.(1991), Morris et al. (1994), Zhang et al. (1997) e demonstraram a facilidade com que as Redes Neurais Artificiais (RNA's) resolvem problemas até então extremamente difíceis de serem solucionados pelos métodos tradicionais, como por exemplo, a inferenciação de variáveis. Diante disso, propõe-se no presente trabalho o uso das RNA's para a inferência das composições de topo da coluna em batelada, a partir de medições de temperatura, tornando possível assim, a implementação do sistema de controle proposto em tempo real para misturas multicomponente.

CAPÍTULO 3

MONTAGEM EXPERIMENTAL

3.1 - INTRODUÇÃO

Quando um computador é responsável pelas ações corretivas num sistema de controle dá-se o nome de sistema de controle digital direto ("Direct Digital Control"). Entretanto, haja visto que o computador trabalha com sinais discretos, são necessários diversos dispositivos para o seu correto funcionamento. Descreve-se, neste capítulo, os diversos dispositivos utilizados no sistema de controle bem como a coluna de destilação em batelada.

3.2 - COLUNA DE DESTILAÇÃO

A coluna de destilação em batelada usada neste trabalho constitui-se de uma estrutura tubular de diâmetro interno de 75 mm e altura de 3 m com 12 pratos distribuídos em seções de dois ou três pratos perfurados, encamisados à vácuo, espelhados e com visor (ver figura. 3.1).

O equipamento possui também os seguintes componentes:

- Refervedor: formado por um reservatório com 10 litros de capacidade, onde a mistura líquida é carregada e aquecida por meio de uma manta elétrica com resistência dupla de 750 W, podendo-se utilizar na forma dupla ou simples;
- Condensador total: de 650 mm de altura e 75 mm de diâmetro interno, com resfriamento na camisa e por serpentina dupla;
- Frascos Coletores: com capacidade de 5 litros, necessários para o armazenamento dos produtos provenientes da coluna, assim como, as misturas intermediárias.

3.3 - INTERFACE PROCESSO/MICROCOMPUTADOR

Uma vez que o microcomputador só pode processar sinais na forma de números binários e o processo utiliza sinais fisico-elétricos na forma analógica e/ou digital (sinais pneumáticos, eletrônicos, etc.), faz-se necessário a conversão entre estes sinais. Portanto, um dispositivo capaz de realizar tal função, chamado placa ADA., deve ser usado.

3.3.1 - PLACA ANALÓGICA-DIGITAL-ANALÓGICA (ADA)

O componente responsável pela interface entre o processo estudado e o microcomputador é a placa ADA. Ela é composta por: um conversor analógico-digital, responsável pela conversão do sinal de tensão oriundo do processo em sinal digital numérico

que pode ser processado pelo microcomputador; um conversor digital-digital, que converte sinais digitais numéricos em sinais digitais de voltagem (0 a 5V) pelos quais aciona-se a válvula magnética e seleciona-se o canal da temperatura a ser lida; e por um conversor digital-analógico cuja finalidade é a reconstrução do sinal.

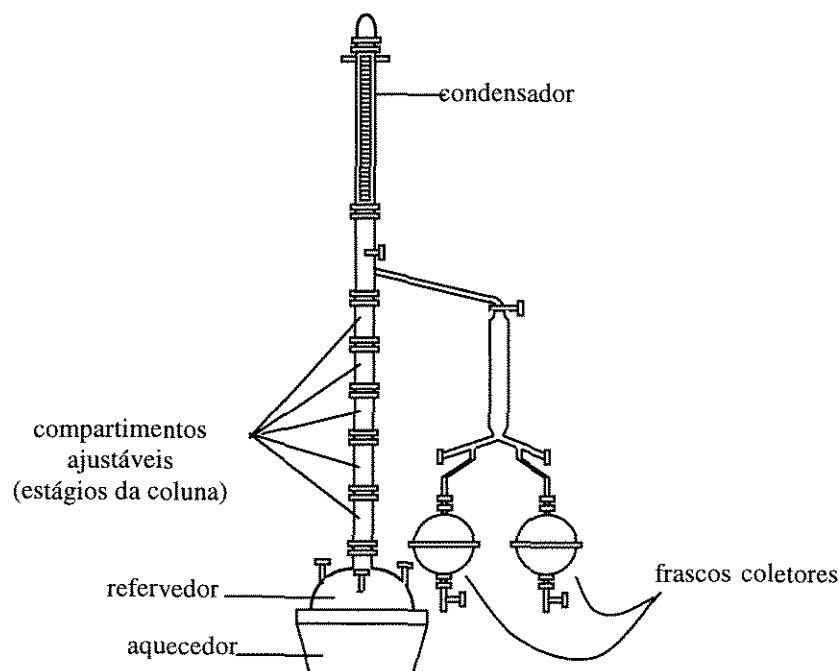


Figura 3.1 - Coluna de destilação em batelada.

3.3.2 - CONVERSOR ANALÓGICO-DIGITAL

O conversor analógico-digital transforma o sinal contínuo de corrente (fig. 3.2a) num sinal digital na forma de um número binário (fig. 3.2b). Fazem parte deste dispositivo um “sample hold” e um multiplexador.

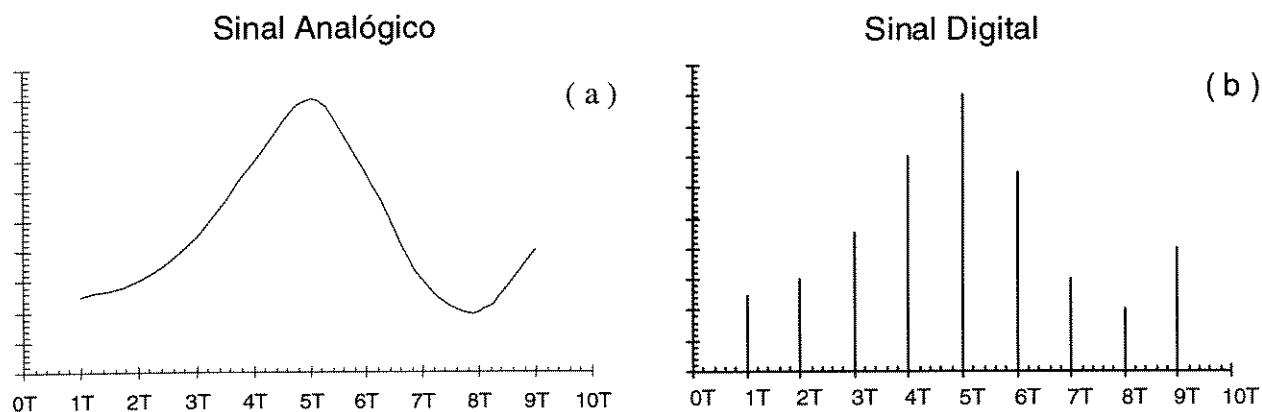


Figura 3.2 – Conversão analógico-digital (DA).

O “sample hold” recebe os sinais contínuos e produz sinais discretos amostrados em intervalos de tempo denominados períodos de amostragem (fig. 3.3).

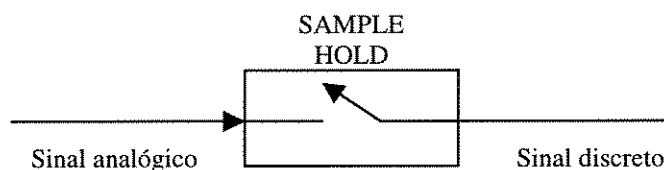


Figura 3.3 – Amostrador.

O multiplexador funciona como um interruptor de várias portas para sinais analógicos. Com este dispositivo torna-se possível converter mais de um sinal analógico com apenas um conversor analógico-digital (fig. 3.4).

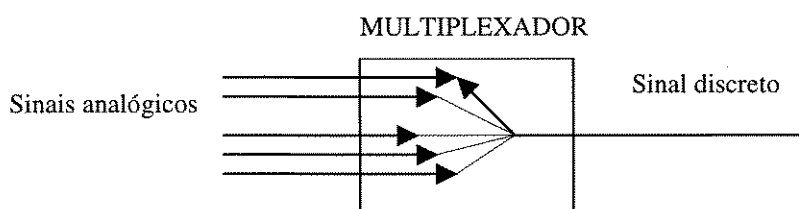


Figura 3.4 – Multiplexador.

3.3.3 - CONVERSOR DIGITAL-DIGITAL.

O sinal digital na forma de número binário é convertido em sinal digital de tensão (0 ou 5 V) pelo conversor digital-digital. Este sinal de tensão é utilizado para o seleccionar o canal

do termopar a ser lido (topo ou fundo) e, também, acionar a válvula magnética para alterar a razão de refluxo.

3.3.4 - CONVERSOR DIGITAL-ANALÓGICO (DATA HOLD).

Este dispositivo converte o sinal digital na forma de número binário (fig. 3.5a) em sinal analógico (fig. 3.5b) efetuando a reconstrução do sinal necessário para a atuação no elemento final de controle.

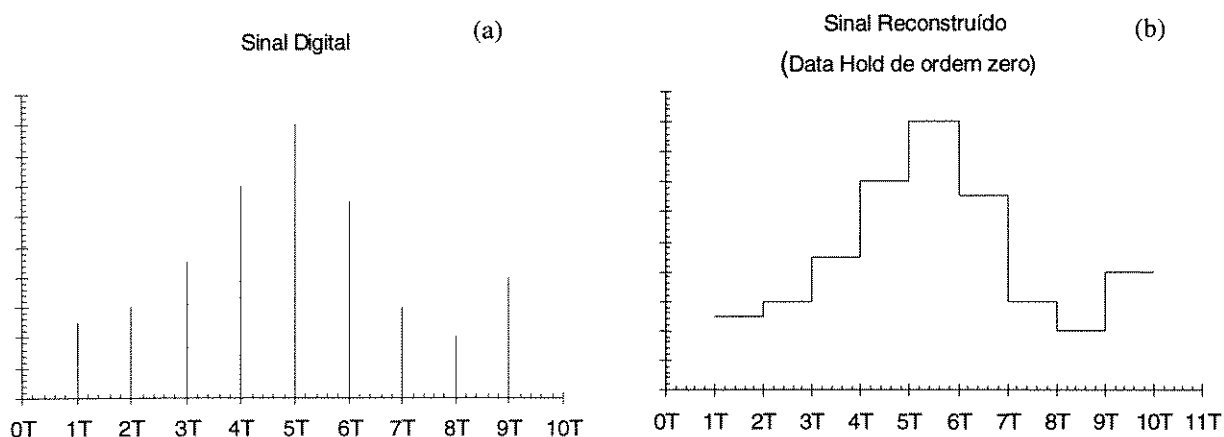


Figura 3.5 – Reconstrução do sinal

A medida em que a ordem de reconstrução aumenta melhor o resultado desta conversão. Entretanto, isto aumenta o tempo computacional dispensado à reconstrução de sinal. Deste modo, a escolha da ordem da reconstrução deve considerar estes dois fatores. A maioria dos sistemas de controle utiliza ordem zero de reconstrução ("zero-order hold") por sua facilidade e simplicidade.

Na falta do "data-hold" os impulsos enviados pelo computador não seriam capazes de manter o elemento final de controle na posição determinada pelo controlador visto que cessado o impulso a válvula retornaria a sua posição padrão (aberta ou fechada) e só atuaria novamente quando recebesse um novo impulso. Apesar disso, neste trabalho não se utilizou o conversor DA dada a natureza "on-off" da válvula magnética e tendo em vista que entre os intervalos de amostragem a válvula magnética não permanecia numa posição fixa e nem seguia uma função transferência definida. Para que a válvula mantivesse a posição estabelecida pelo programa de controle um temporizador ("timer") foi implementado, cuja função era manter a válvula na posição determinada durante o tempo necessário para que a razão de refluxo calculada fosse imposta ao processo.

3.4 - A VÁLVULA MAGNÉTICA.

O elemento final de controle na coluna estudada é a válvula magnética. É através dela que se modifica a razão de refluxo do processo. Ela recebe sinais digitais na forma de tensão de 0 ou 5 V correspondentes as posições refluxo ou destilado, respectivamente.

O sinal proveniente do microcomputador, que corresponde a razão de refluxo calculado pelo algoritmo de controle, é convertido no conversor digital-digital e enviado a válvula magnética a qual assume uma das posições (refluxo ou destilado) e mantém esta posição durante um período de tempo calculado que não pode ser superior ao intervalo de amostragem.

3.5 - SISTEMA DE AUTOMAÇÃO E CONTROLE.

O sistema de automação e controle, esquematizado na figura 3.6, tem como propósito controlar a composição molar no topo da coluna de destilação em batelada através da manipulação da razão de refluxo.

O sistema opera de acordo com as seguintes etapas:

1ª) O programa de controle implementado no microcomputador envia um sinal, através do conversor digital-digital da placa ADA, para selecionar o canal analógico desejado (termopar de topo);

2ª) Selecionado o canal desejado faz-se a leitura do sinal analógico que é convertido em sinal digital pelo conversor analógico-digital e, em seguida, transformado num valor de temperatura segundo uma curva de calibração;

3ª) Após 10.000 leituras os valores de temperatura são filtrados através de um filtro exponencial duplo implementado no programa de controle para a remoção do ruído existente no circuito;

4ª) Depois que as etapas 1, 2 e 3 forem realizadas para o outro canal (termopar de fundo) os dados de temperatura obtidos são utilizados pelo programa para a inferenciação e o controle da composição de topo. É nesta etapa que a razão de refluxo é calculada segundo o algoritmo de controle utilizado;

5ª) Uma vez calculada a razão de refluxo o programa de controle envia um sinal, através do conversor digital-digital da placa ADA, para o acionamento da válvula magnética.

Este procedimento se repete em todos os intervalos de amostragem.

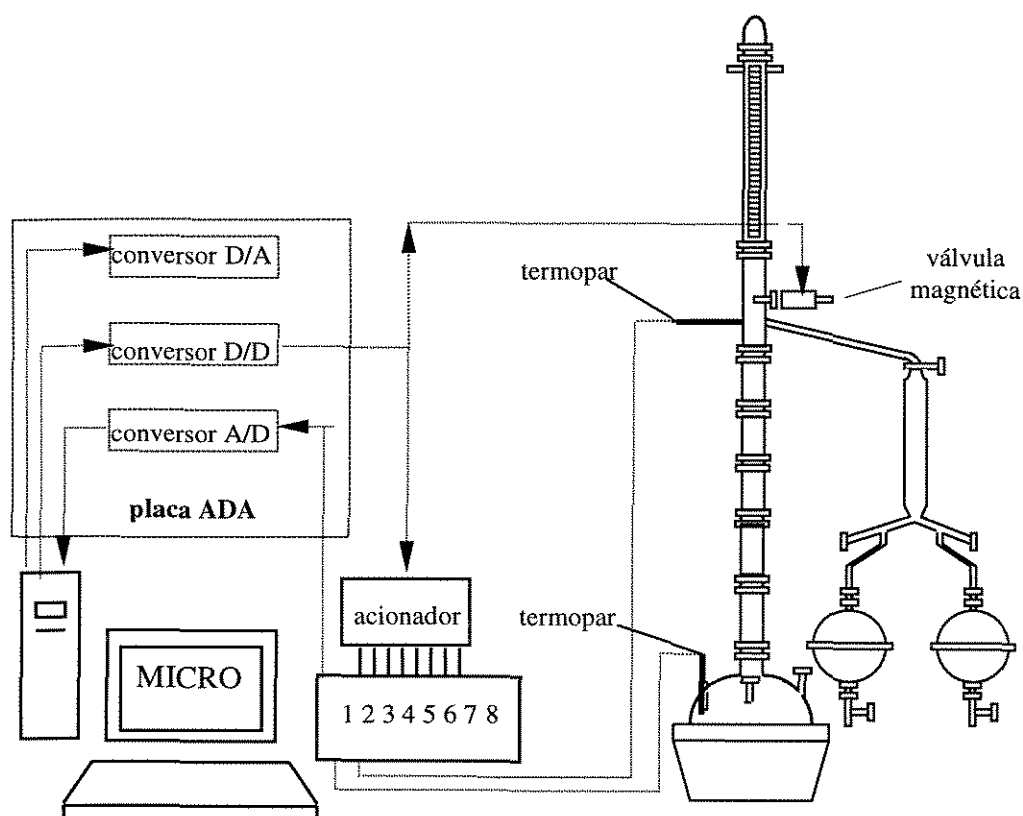


Figura 3.6 – Sistema de automação e controle.

CAPÍTULO 4
IMPLEMENTAÇÃO
DAS ESTRATÉGIAS
DE CONTROLE

4.1 - INTRODUÇÃO

Na implementação de sistemas de controle, o regime transiente e a versatilidade das colunas de destilação em batelada exigem um tratamento bem distinto daquele normalmente seguido em colunas contínuas. Estas propriedades sugerem que os controladores devem modificar seus parâmetros visando seu ajuste às novas características do processo.

Neste capítulo é apresentado o desenvolvimento e implementação experimental de um sistema de controle auto-ajustável ("Self Tuning Regulator") numa coluna de destilação em batelada. O equivalente digital de um controlador PI e o algoritmo de Dahlin são individualmente testados como estratégias de controle.

O desempenho do STR é comparado aos equivalentes digitais dos controladores convencionais (P,PI e PID) e do Controlador Adaptativo Programável (PAC), todos estes implementados num microcomputador 486 DX 33MHz acoplado a coluna.

4.2 - CONTROLE CONVENCIONAL

Sabendo que na engenharia química a maioria dos processos pode ser aproximada por um modelo de primeira ordem com atraso de transporte e procurando usar um sistema de controle o mais simples possível, utilizou-se tal modelo para aproximar o comportamento do sistema, conforme a equação (3.1):

$$G_p(s) = \frac{Y(s)}{U(s)} = \frac{K \cdot e^{-\theta s}}{\tau \cdot s + 1} \quad (4.1)$$

sendo: $G_p(s)$, função transferência do processo;

Y , composição de topo da coluna;

U , razão de refluxo;

K , ganho do processo;

τ , constante de tempo;

θ , atraso por transporte.

Com o auxílio de um método de identificação, como por exemplo a Curva de Reação do processo, encontram-se os parâmetros do modelo (eq. 4.1) K , τ e θ . De posse destes parâmetros pode-se utilizar, por exemplo, o ajuste de Cohen-Coon para a determinação dos parâmetros dos controladores convencionais (P, PI e PID), conforme a tabela 4.1.

Tabela 4.1 – Parâmetros dos controladores convencionais.

Tipo do Controlador	Ajuste de Cohen-Coon		
	Ganho do controlador (K_c)	Constante de tempo integral (τ_I)	Constante de tempo derivativo (τ_D)
P	$K_c = \frac{1}{K} \cdot \frac{\tau}{\theta} \cdot \left(1 + \frac{\theta}{3\tau}\right)$	–	–
PI	$K_c = \frac{1}{K} \cdot \frac{\tau}{\theta} \cdot \left(0,9 + \frac{\theta}{12\tau}\right)$	$\tau_I = \theta \cdot \frac{30 + 3\theta/\tau}{9 + 20\theta/\tau}$	–
PID	$K_c = \frac{1}{K} \cdot \frac{\tau}{\theta} \cdot \left(\frac{4}{3} + \frac{\theta}{4\tau}\right)$	$\tau_I = \theta \cdot \frac{32 + 6\theta/\tau}{13 + 8\theta/\tau}$	$\tau_D = \theta \cdot \frac{4}{11 + 2\theta/\tau}$

* Fonte: Stephanopoulos (1984), pp. 312.

Com os avanços na área computacional, a implementação dos controladores convencionais na forma digital para a sua utilização em microprocessadores tem crescido bastante. Para tanto, as funções transferência dos controladores analógicos convencionais devem ser modificadas para a sua adequação aos sistemas digitais, que realizam seus cálculos em intervalos regulares de tempo T (o intervalo de amostragem). Sendo assim, com a ajuda da tabela 4.2, que mostra as funções transferência no domínio de Laplace e da transformada Z, obtêm-se as equações dos equivalentes digitais dos controladores P, PI e PID (eqs. 4.2).

Tabela 4.2 – Funções transferência.

Tipo do Controlador	Transformada de Laplace	Transformada Z
P	$D(s) = \frac{u(s)}{e(s)} = K_c$	$D(z) = \frac{u(z)}{e(z)} = K_c \cdot \left(\frac{1}{1 - z^{-1}}\right)$
PI	$D(s) = \frac{u(s)}{e(s)} = K_c \cdot \left(1 + \frac{1}{\tau_I s}\right)$	$D(z) = \frac{u(z)}{e(z)} = K_c \cdot \left(1 + \frac{T}{\tau_I} \cdot \frac{1}{1 - z^{-1}}\right)$
PID	$D(s) = \frac{u(s)}{e(s)} = K_c \cdot \left(1 + \frac{1}{\tau_I s} + \tau_D s\right)$	$D(z) = \frac{u(z)}{e(z)} = K_c \cdot \left(1 + \frac{T}{\tau_I} \cdot \frac{1}{1 - z^{-1}} + \frac{\tau_D}{T} \cdot \frac{1 - 2z^{-1} + z^{-2}}{1 - z^{-1}}\right)$

* Fonte: Stephanopoulos (1984), pp. 611.

$$u_i = u_{i-1} + K_c \cdot (e_i - e_{i-1}) \quad (4.2a)$$

$$u_i = u_{i-1} + K_c \cdot \left(1 + \frac{T}{\tau_i}\right) \cdot e_i - K_c \cdot e_{i-1} \quad (4.2b)$$

$$u_i = u_{i-1} + K_c \cdot \left(1 + \frac{T}{\tau_i} + \frac{\tau_D}{T}\right) \cdot e_i - K_c \cdot \left(1 + 2 \cdot \frac{\tau_D}{T}\right) \cdot e_{i-1} + K_c \cdot \frac{\tau_D}{T} \cdot e_{i-2} \quad (4.2c)$$

sendo: u_i a variável manipulada no instante atual;

e_i o erro atual na variável controlada.

Estas equações são então implementadas no programa de controle para o ajuste da razão de refluxo. Contudo, os parâmetros destes controladores permanecem constantes durante toda a operação e, como as características da destilação em batelada mudam constantemente, com o progresso da batelada o ajuste inicial tende a perder sua eficiência.

Uma possível extensão destes controladores seria obtida através da repetição periódica do ajuste dos parâmetros do controlador. Isto constitui a essência do Controle Adaptativo.

4.3 – CONTROLE ADAPTATIVO

Os diversos tipos de controladores adaptativo são constituídos por uma malha simples de controle por retroalimentação contendo um processo e um regulador com parâmetros ajustáveis. A diferença básica entre eles consiste no modo como os parâmetros do controlador são ajustados em resposta as variações no processo.

4.3.1 – CONTROLADOR ADAPTATIVO PROGRAMÁVEL (PAC)

Programar as adaptações do controlador é o método mais simples de se efetuarem os ajustes em seus parâmetros. Isto é possível desde que existam variáveis do processo que estejam intimamente relacionadas com a dinâmica do mesmo. Satisfeita esta condição, os efeitos das variações dos parâmetros do processo podem ser amenizados alterando-se os parâmetros do controlador como funções destas variáveis auxiliares (fig. 4.1).

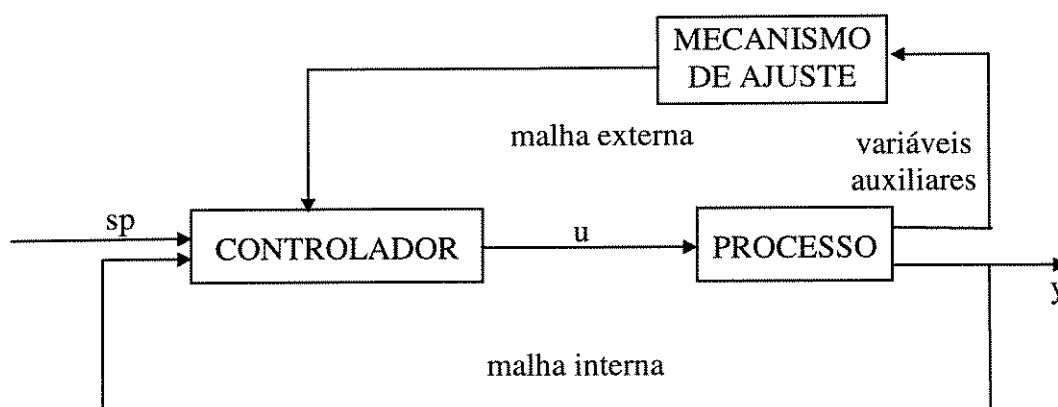


DIAGRAMA DE BLOCOS DO CONTROLADOR ADAPTATIVO PROGRAMÁVEL (PAC)

Figura 4.1 – Controlador Adaptativo Programável (PAC).

A determinação das variáveis auxiliares normalmente baseia-se no conhecimento físico do sistema. E, uma vez escolhidas, os parâmetros do controlador poderão ser encontrados em qualquer condição de operação mediante um método de sintonização adequado. Já a estabilidade e o desempenho do sistema são normalmente avaliados através de simulações.

A principal vantagem do PAC está na relação direta, sob a forma de funções matemáticas, existente entre as variáveis auxiliares e os parâmetros do controlador. Sendo os fatores limitantes dependentes da rapidez com que as variáveis auxiliares respondem às mudanças no processo.

Contudo, o PAC não é capaz de compensar um planejamento inadequado visto que não existe retroalimentação dos parâmetros do controlador. Sendo assim, o PAC pode ser visto como um sistema de controle por retroalimentação onde os parâmetros do controlador são ajustados por compensação “feedforward”.

Por outro lado, o tempo consumido para o projeto do sistema, cujos parâmetros devem ser determinados para diversas condições de operação do processo, e a necessidade de testar o seu desempenho através de exaustivas simulações tornam a implementação do PAC honerosa.

4.3.1.1 - IMPLEMENTAÇÃO DO PAC

Para manter constante a composição de topo numa coluna de destilação em batelada utilizando um controlador adaptativo do tipo PAC busca-se inicialmente uma variável auxiliar que esteja intimamente relacionada com a dinâmica deste processo. Com este intuito, utiliza-

se a composição de fundo da coluna pois ela representa um forte indicativo das composições futuras de topo no equipamento.

Escolhida a variável auxiliar, divide-se o processo numa sequência de estados pseudo-estacionários visando estudar o sistema como um processo contínuo. Em seguida, através de simulações “off-line” de colunas contínuas em diversos estados estacionários e após uma regressão não-linear dos dados obtidos, encontra-se a relação entre a composição de fundo e a razão de refluxo. Deste modo, obtêm-se o termo de estado estacionário (u_{ss}) para a equação digital do controlador, conforme a equação 4.3a.

$$u_{ss} = f_1(x_B) \quad (4.3a)$$

Segundo Cunha (1996), para uma mistura de n-hexano e n-heptano com fração molar inicial de 33% em n-hexano, a equação 4.3a para um controlador PI e dada por

$$u_{ss} = 0,435683 \cdot x_B^{-1,90183} \quad (4.3b)$$

O mecanismo de ajuste dos parâmetros do controlador pode ser implementado através da caracterização prévia do processo. Fileti(1996) e Cunha (1996) utilizaram esta ferramenta aproximando o processo por uma função transferência de primeira ordem com atraso de transporte (eq. 3.1) e aplicando o Método da Curva de Reação do Processo “off-line” para a obtenção dos parâmetros (K , τ e θ) em cada intervalo pseudo-estacionário.

De posse dos parâmetros do processo, pode-se programar a adaptação dos parâmetros de um controlador PI conforme as equações semi-empíricas propostas por Cohen-Coon (tabela 4.1).

Finalmente, por meio de um ajuste de curvas obtêm-se os parâmetros do controlador como função da variável auxiliar (composição de fundo). Tendo Cunha (1996) obtido para a mistura 30% molar de n-hexano e 70% molar de n-heptano as seguintes relações:

$$K_C = f_2(x_B) = 22,4905 \cdot x_B^{-1,90183} \quad (4.5a)$$

$$\tau_I = f_3(x_B) = 0,0998669 - 0,0166394 \cdot x_B \quad (4.5b)$$

Entretanto, vale ressaltar que a medida de composição normalmente exige a utilização de instrumentos sofisticados ou de custo elevado. Para contornar este obstáculo, pode-se

utilizar no caso de misturas binárias, dados de equilíbrio líquido-vapor a uma dada pressão para a obter a relação entre a temperatura e composição, visto que a primeira é facilmente medida. Para mistura n-hexano e n-heptano utilizando-se uma regressão não-linear obtêm-se através de dados da literatura as seguintes expressões para a pressão de 1 atm:

$$x_{fundo} = f_4(T_{fundo}) = 5,19986 - 8,12379 \cdot 10^2 \cdot T_{fundo} + 2,89109 \cdot 10^4 \cdot T_{fundo}^2 \quad (4.6a)$$

$$y_{topo} = f_5(T_{topo}) = -1,10148 + 7,484458 \cdot 10^2 \cdot T_{topo} - 6,45643 \cdot 10^4 \cdot T_{topo}^2 \quad (4.6b)$$

Quando se está trabalhando com misturas multicomponentes a relação entre a temperatura e a composição torna-se mais complexa. Isto implica no uso de técnicas mais avançadas para a inferenciação da composição, dentre as quais pode-se citar as redes neurais, que serão abordadas no capítulo seguinte.

4.3.2 - CONTROLADOR AUTO-AJUSTÁVEL (STR)

Outro método de ajuste dos parâmetros do controlador é o auto-ajustável. Ele foi originalmente proposto por Kalman (1958) e posteriormente clareado por Aström e Wittenmark (1973). Ele pode ser visto como constituído por duas malhas (fig. 4.3). A malha interna é constituída pelo processo e por um regulador ordinário por retroalimentação. Já a malha mais externa é composta por um estimador de parâmetros recursivo e um mecanismo de ajuste dos parâmetros do controlador.

Segundo Aström (1983), a grande atenção dada ao STR se deve a sua flexibilidade, fácil entendimento e fácil implementação com microprocessadores. Em vista disso, diferentes extensões surgiram quanto ao método de ajuste dos parâmetros e ao tipo de estimador de parâmetros recursivo utilizado.

Neste trabalho, onde o processo mostra-se transiente e fortemente não-linear, o STR surge como uma atraente ferramenta para o controle da composição no topo da coluna.

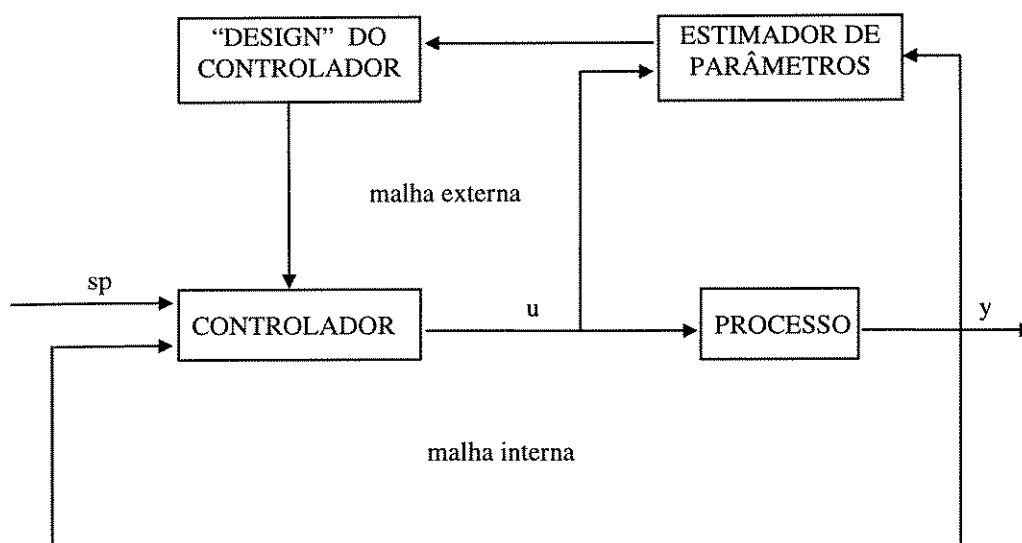


DIAGRAMA DE BLOCOS DO CONTROLADOR ALTO-AJUSTÁVEL (STR)

Figura 4.3 – Controlador Auto-ajustável (STR).

4.3.2.1 - IMPLEMENTAÇÃO DO STR

Na implementação do STR não é necessário o conhecimento das propriedades dos componentes presentes no sistema. Basta apenas, representar o processo por um modelo discreto obtido de uma função transferência de ordem n com ou sem atraso de transporte ou por um modelo discreto do tipo polinomial.

Estando o controlador situado no programa implementado no microcomputador, os sinais discretos normalmente são convertidos em sinais analógicos haja visto que a maioria dos elementos finais de controle operam com tais sinais. Isto implica que além da função transferência do processo deve-se levar em consideração a função transferência do “data-hold” utilizado na reconstrução do sinal.

Contudo, a válvula magnética utilizada neste estudo para manipular a razão de refluxo da coluna possui as características de uma válvula “on-off”, ora direcionando todo o fluxo de líquido, proveniente do condensador, para refluxo(“on”) ora para o destilado(“off”). Isto associado ao fato de que o conversor digital-digital da placa ADA pode ser utilizado simultaneamente tanto para a seleção do canal do termopar a ser lido quanto para o acionamento da válvula, por meio de um sinal digital de tensão, tornou desnecessário utilizar o conversor digital-analógico (“data-hold”). Sendo assim, a função transferência de pulso

entre as variáveis controlada (y) e manipulada (u) pode ser obtida com o ajuda da transformada Z:

$$\frac{\hat{y}(z)}{\hat{u}(z)} = G_p(z) = Z\{G_p(s)\} = Z\left\{\frac{K \cdot e^{-\theta s}}{\tau \cdot s + 1}\right\} \quad (4.7)$$

Da equação 4.7 obtêm-se o modelo discreto para o processo (ver ANEXO A):

$$y_i = a_1 \cdot y_{i-1} + b_1 \cdot u_{i-1-nd} + bias \quad (4.8a)$$

sendo:

$$a_1 = \exp\left(\frac{-T}{\tau}\right) \quad (4.8b)$$

$$b_1 = \frac{K}{\tau} \cdot \exp\left(\frac{-(1+nd) \cdot T - \theta}{\tau}\right) \quad (4.8c)$$

$$bias = y_{ss} \cdot (1 - a_1) - b_1 \cdot u_{ss} \quad (4.8d)$$

$$nd = \text{int}\left(\frac{\theta}{T}\right) \quad (4.8e)$$

4.3.2.2 - IDENTIFICAÇÃO DO PROCESSO

O processo de identificação constitui uma etapa de grande importância na implementação do STR. Existem na literatura diversos procedimentos de estimação recursiva que podem ser utilizados na realização desta tarefa. Entretanto, após testes experimentais preliminares de vários métodos (Newton, Broyden, Powell, etc.) optou-se pelo Método de Programação Quadrática Sucessiva aplicado a mínimos quadrados existente na biblioteca de subrotinas matemáticas NAG (Anexo B). Tal escolha está baseada na observação de que os parâmetros do modelo necessitam de restrições para que as adaptações do processo sejam suavizadas, constituindo uma sensível melhora no controle do equipamento em estudo.

Os parâmetros do modelo do processo (K, τ e θ) são obtidos através da resolução do seguinte problema:

$$\text{Minimize } F(X) = \frac{1}{2} \cdot f(X)^T f(X) = \frac{1}{2I} \sum_{i=1}^I f_i(X)^2$$

sujeito a: $L \leq X \leq S$

sendo:

$$X, \text{ o vetor dos parâmetros a determinar} - X = [K, \tau, \theta]^T \quad (4.9a)$$

$$L, \text{ limite inferior dos parâmetros} - L = [L_K, L_\tau, L_\theta]^T \quad (4.9b)$$

$$S, \text{ limite superior dos parâmetros} - S = [S_K, S_\tau, S_\theta]^T \quad (4.9c)$$

$$f_i(X), \text{ subfunções do problema} - f_i(X) = \tilde{y}_i - y_i = \tilde{y}_i - a_1 \cdot y_{i-1} - b_1 \cdot u_{i-1-nd} \quad (4.9d)$$

I, número de instantes anteriores utilizados.

Portanto, os valores otimizados dos parâmetros do processo serão aqueles que resultarem num erro quadrático médio mínimo entre a variável controlada medida (y) e o calculado pelo modelo discreto de acordo com as equações 4.8.

Comumente, faz-se necessário perturbar o processo real por uma variação pulso ou degrau na variável manipulada, vazão de refluxo (u), efetuando-se a amostragem da variável controlada (y), composição de topo. Mais uma vez, as características da válvula magnética utilizada neste estudo, além da natureza não-estacionária do processo, tornam desnecessárias tais perturbações.

4.3.2.3 - PROJETO DO CONTROLADOR

Determinados os parâmetros do modelo do processo pode-se, então, efetuar o ajuste do controlador. Novamente, o STR mostra-se bastante flexível quanto a escolha da metodologia de projeto do controlador. No presente trabalho, o equivalente digital de um controlador PI, ajustado segundo critério proposto por Cohen-Coon e uma lei de controle obtida do algoritmo de Dahlin foram implementados individualmente.

A equação discretizada de um controlador PI foi utilizada na forma velocidade como sugerido na análise do controle convencional (eq. 4.2b). Contudo, independentemente da forma escolhida, o controlador PI tem seus parâmetros ajustados a cada instante de amostragem conforme tabela 4.1.

Já o algoritmo proposto por Dahlin implementado, impõe a condição de que a resposta em malha fechada de um Controlador Digital Direto (DDC) se comporte como um sistema de primeira ordem com atraso de transporte, tendo sido perturbado o valor de referência em um degrau unitário. Deste modo, no domínio de Laplace:

$$\bar{y}(s) = \frac{e^{-\theta s}}{\mu \cdot s + 1} \cdot \frac{1}{s} \quad (4.10)$$

sendo: θ , o atraso por transporte, $\theta = k \cdot T$, com k inteiro;

μ , constante de tempo da resposta desejada (parâmetro ajustável).

Discretizando a equação 4.10 com a ajuda da transformada Z obtém-se:

$$\hat{y}(z) = \frac{z^{-k} \cdot (1 - e^{-T/\mu}) \cdot z^{-1}}{(1 - z^{-1}) \cdot (1 - e^{-T/\mu} \cdot z^{-1})} \quad (4.11)$$

Impondo a condição de degrau unitário no valor de referência encontra-se:

$$\frac{\hat{y}(z)}{\hat{y}_{sp}(z)} = \frac{z^{-k} \cdot (1 - e^{-T/\mu}) \cdot z^{-1}}{1 - e^{-T/\mu} \cdot z^{-1}} \quad (4.12)$$

Uma vez que a resposta em malha fechada de um DDC típico diante de variações no “set-point” pode ser representado por:

$$\hat{y}(z) = \frac{G_p(z) \cdot D(z)}{1 + G_p(z) \cdot D(z)} \cdot \hat{y}_{sp}(z) \quad (4.13)$$

obtêm-se a função transferência do controlador $D(z)$:

$$D(z) = \frac{1}{G_p(z)} \cdot \frac{\hat{y}(z) / \hat{y}_{sp}(z)}{1 - \hat{y}(z) / \hat{y}_{sp}(z)} \quad (4.14)$$

Portanto, conhecendo-se a função transferência de pulso do processo

$$\frac{u(z)}{E(z)} = D(z) = \frac{1 - a_1 \cdot z^{-1}}{b_1 \cdot z^{-1}} \cdot \frac{(1 - e^{-T/\mu}) \cdot z^{-1-k}}{1 - e^{-T/\mu} \cdot z^{-1} - (1 - e^{-T/\mu}) \cdot z^{-1-k}} \quad (4.15)$$

Assim, a lei de controle pode ser obtida

$$u_i = u_{ss} + \frac{d_1}{b_1} \cdot e_i + \frac{d_2 - a_1 \cdot d_1}{b_1} \cdot e_{i-1} - \frac{a_1 \cdot d_2}{b_1} \cdot e_{i-2} + c_1 \cdot u_{i-1} + d_1 \cdot u_{i-1-nd} + d_2 \cdot u_{i-2-nd} \quad (4.16a)$$

sendo:

$$\begin{aligned} a_1 &= \exp\left(-\frac{T}{\tau}\right); \\ b_1 &= \frac{K}{\tau} \cdot \exp\left[-\frac{(1+nd) \cdot T - \theta}{\tau}\right]; \\ c_1 &= \exp\left(-\frac{T}{\mu}\right); \\ d_1 &= 1 - \exp\left[-\frac{(1+nd) \cdot T - \theta}{\mu}\right]; \\ d_2 &= 1 - d_1 - c_1. \end{aligned} \quad (4.16b)$$

4.4 - RESULTADOS

Diversas bateladas foram realizadas com o intuito da verificação experimental do desempenho dos modelos de controladores expostos anteriormente e, também, para uma detalhada comparação entre eles.

Inicialmente utilizou-se uma mistura binária de n-hexano e n-heptano, com diferentes valores de concentração inicial, intervalo de amostragem, número de instantes usados na identificação, entre outras variáveis para a obtenção de um ajuste adequado dos controladores. As composições de topo e fundo foram determinadas através de dados de equilíbrio líquido-vapor disponíveis na literatura e medições de temperatura. Isto não tira o caráter genérico do modelo visto que sistemas de inferenciação poderiam ser empregados.

O modelo já devidamente adequado ao sistema em estudo foi usado no controle da composição de topo com uma mistura ternária n-hexano/cicloexano/n-heptano. Neste momento, uma Rede Neural Artificial foi utilizada na inferenciação da composição, cuja explicação detalhada será dada no capítulo seguinte.

4.4.1 – CONTROLE CONVENCIONAL

Após a determinação de diversos valores de K , τ e θ correspondentes a diferentes faixas de operação, estes parâmetros foram usados na obtenção dos parâmetros dos controladores convencionais P, PI e PID com o auxílio do ajuste de Cohen-Coon. Os equivalente digitais na forma velocidade de tais controladores (eqs 4.2a(P), 4.2b(PI) e 4.2c(PID)) foram, então, individualmente implementados e testados no sistema em estudo.

Conforme pode ser visto na figura 4.4a, o controlador proporcional para baixos valores de K_c não conseguiu ajustar devidamente o sistema. Já para valores intermediários e elevados de K_c (figuras 4.4b e 4.4c) tal controlador alcançou um bom ajuste do ponto de vista da variável controlada (y_i). Por outro lado, dadas as ações bruscas observadas na variável manipulada (R), o controlador proporcional mostrou-se inadequado uma vez que estas ações acarretariam um desgaste excessivo no elemento final de controle.

Utilizando controlador PI (figura 4.5) observa-se um ajuste mais apurado do sistema com relação tanto a variável controlada quanto a variável manipulada. Isto sugere que tal controlador seria o indicado para o sistema. Entretanto, uma vez que a destilação em batelada pode ser utilizada para processar diversas misturas sob diferentes condições operacionais, seriam necessários diversos ajustes ao longo da batelada para a remoção de todos os componentes de interesse. Para esta tarefa o operador da coluna seria obrigado a ajustar manualmente o controlador.

O controlador PID tanto na forma posição quanto na forma velocidade também foi testado na coluna conforme figuras 4.6a e 4.6b. Este apresentou um comportamento similar ao controlador proporcional com ganho elevado. Para melhorar o desempenho deste controlador seriam necessários novos ajustes, tarefa que se tornaria mais complicada a medida em que aumentasse o número de componentes na mistura. Pode-se observar, ainda, o efeito da saturação na forma posição indicando que a forma velocidade seria a mais indicada para o controlador.

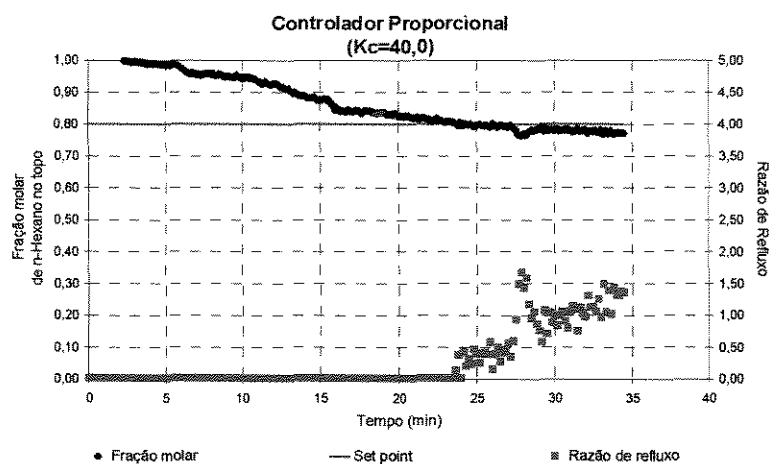


Figura 4.4a – Controlador Proporcional ($K_c = 40,0$).

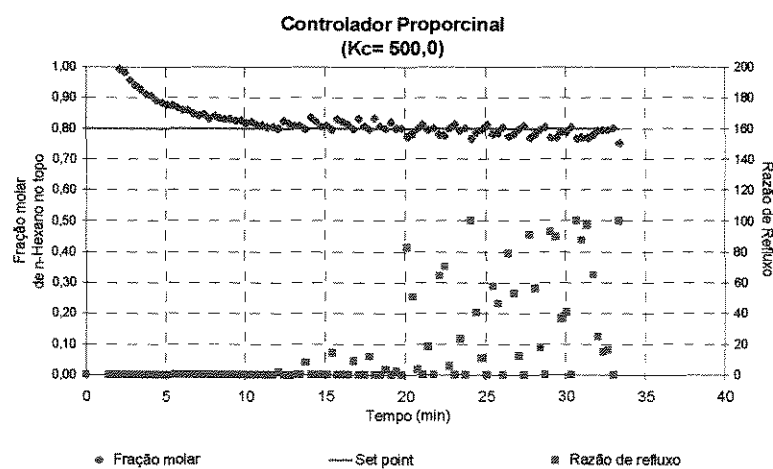


Figura 4.4b – Controlador proporcional ($K_c = 500,0$).

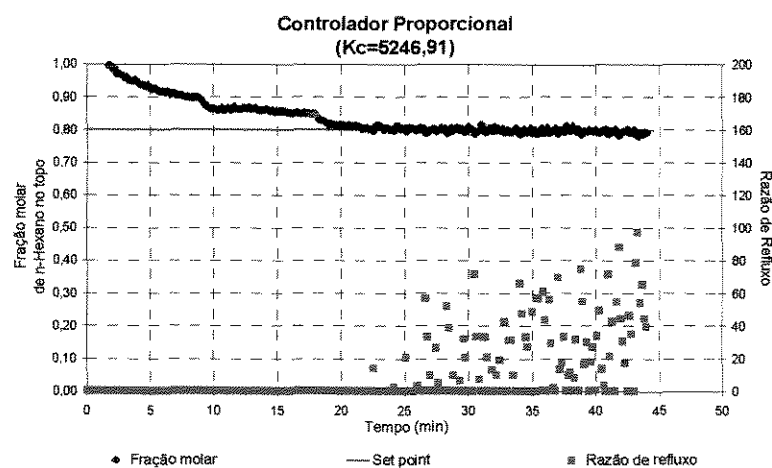


Figura 4.4c - Controlador proporcional ($K_c = 5246,91$).

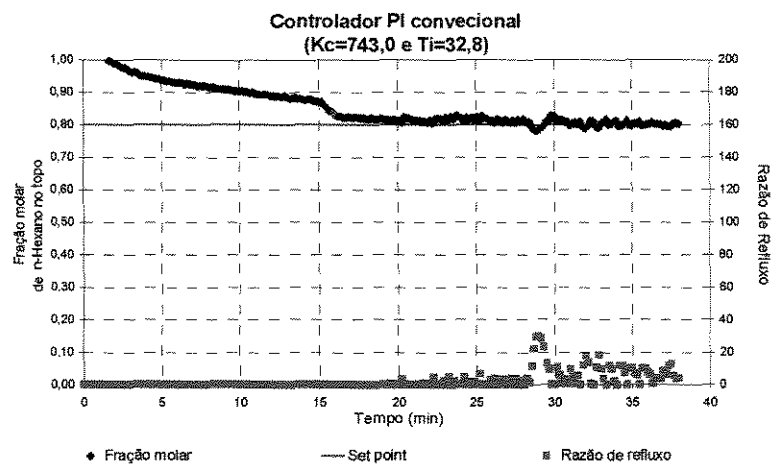


Figura 4.5 – Controlador Proporcional Integral.

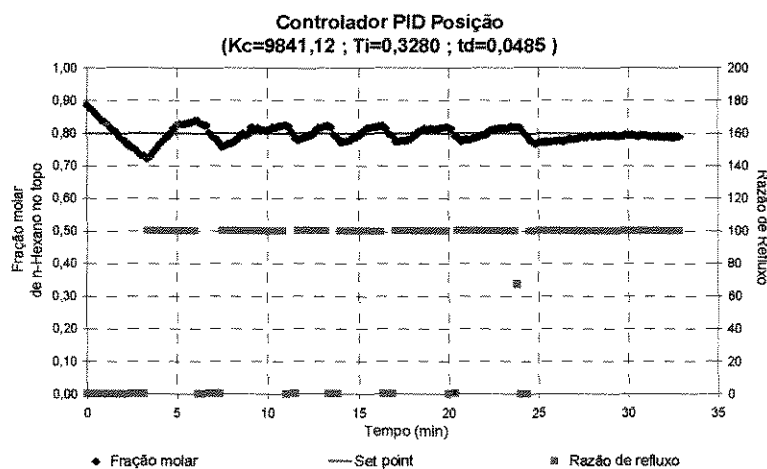


Figura 4.6a - Controlador PID forma posição.

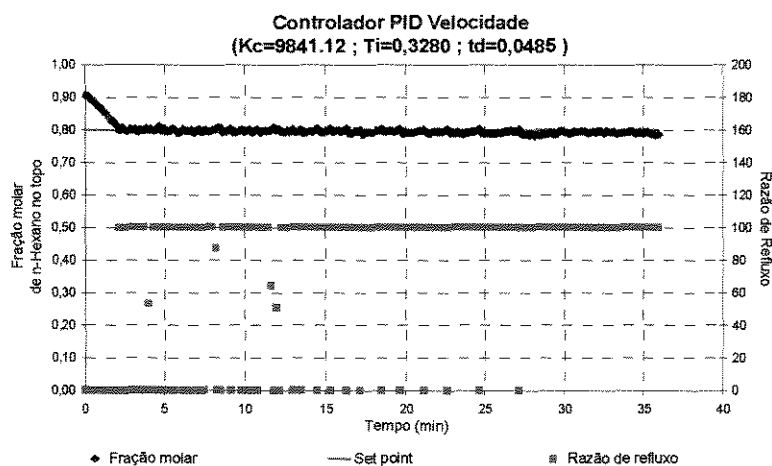


Figura 4.6b - Controlador PID forma velocidade.

Tendo em vista o desempenho dos controladores acima, propõe-se o controlador adaptativo como melhor alternativa para o sistema em estudo, visto que, este tipo de controlador possui a capacidade de ajustar seus parâmetros ao longo da batelada bem como alterar tais parâmetros para a remoção dos demais componentes da mistura.

4.4.2 – CONTROLE ADAPTATIVO

Como foi dito anteriormente, a etapa de identificação do processo é de extrema importância na implementação dos controladores adaptativos. No caso do STR, grandes variações nas ações de controle podem acarretar em problemas de convergência no método de otimização utilizado nesta etapa.

Os resultados experimentais de três métodos de otimização sem restrições são mostrados na figura 4.7, onde se percebe as ações bruscas de controle.

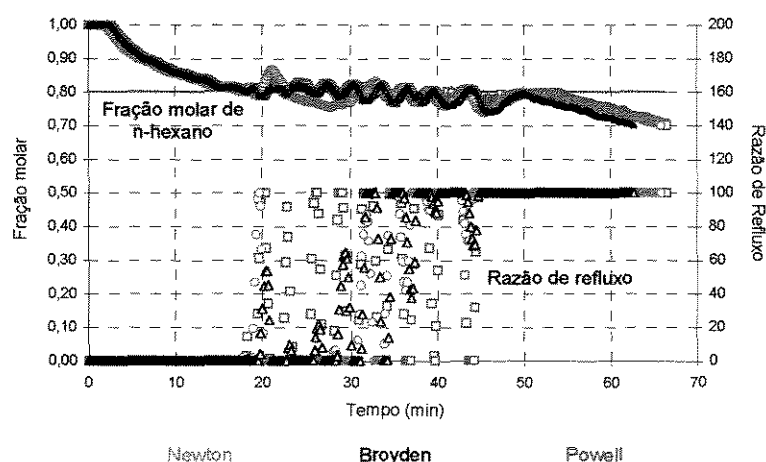


Figura 4.7 – Métodos de Otimização sem Restrições.

Deste modo, um limite de flutuação de $\pm 10\%$ nos parâmetros do processo, em relação ao seu valor imediatamente anterior, foi imposto para a suavização das ações de controle no método SQP da NAG. Em seguida, várias bateladas foram realizadas no sentido de determinar os melhores valores dos demais parâmetros necessários na implementação do STR, tendo-se obtido os resultados expostos na tabela 4.3 para a separação de diversas misturas de n-hexano e n-heptano com "set-point" de 80% molar de n-hexano no topo.

De posse destes dados, os efeitos dos parâmetros sobre o desempenho do controlador STR foram analisados, como mostram os itens seguintes. Além disso, buscando uma melhor comparação entre os diversos ajustes e controladores, o tempo da batelada em cada caso foi escalonado dividindo-se o mesmo pelo tempo final de cada batelada.

Tabela 4.3 – Dados iniciais e finais das batelada.

BAT	Fração molar inicial de n-Hexano	Estratégia de controle	IA	T	Tempo final da batelada (min)	Fração molar média de n-Hexano	Desvio padrão
1	0,194	STR/PI	15	10	63,58	0,815	0,047
2	0,194	STR/PI	10	10	86,62	0,811	0,044
3	0,118	STR/Dahlin ($\mu=1$)	5	10	81,96	0,850	0,054
4	0,253	STR/PI	5	10	68,33	0,809	0,062
5	0,253	STR/Dahlin ($\mu=0,2$)	5	10	65,67	0,811	0,062
6	0,253	STR/PI	5	10	66,33	0,805	0,065
7	0,227	STR/PI	5	7	63,82	0,817	0,069
8	0,227	STR/Dahlin ($\mu=0,2$)	5	7	81,92	0,815	0,070
9	0,227	PAC/PI	5	7	50,52	0,824	0,069

4.4.2.1 – EFEITO DA DINÂMICA DO PROCESSO

O número de instantes usados na identificação (IA) serve para a adequação do modelo à dinâmica do processo. Em princípio, quanto menor for o valor deste parâmetro menor será sua dependência dos instantes anteriores (histórico do sistema). Com base nesta informação, três valores de IA (5, 10 e 15) foram testados na coluna de destilação em batelada estudada. E, conforme pode ser visto na tabela 4.3 (BAT's 1, 2 e 4), o desvio padrão na variável controlada aumentou com a diminuição de IA, indicando uma piora do controlador. Por outro lado, os gráficos 4.8a e 4.8b mostram que o comportamento da variável manipulada melhorou à medida em que IA diminuiu. Além disso, razões de refluxo menores foram necessárias durante a maior parte da batelada, indicando uma redução nos custos operacionais do sistema.

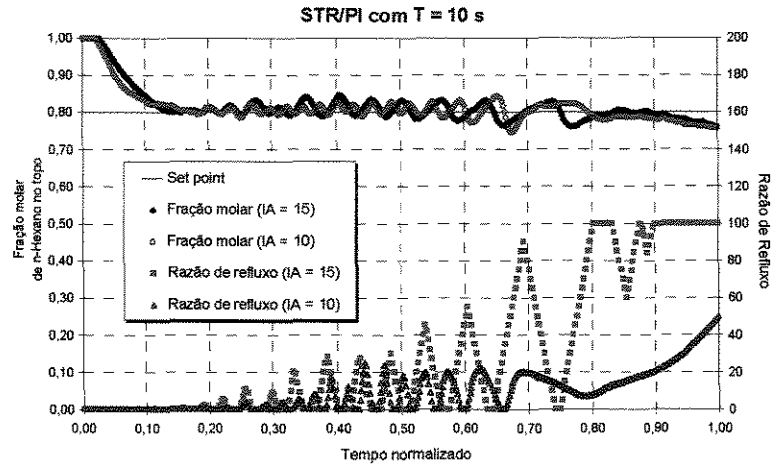


Figura 4.8a - Controlador STR/PI.

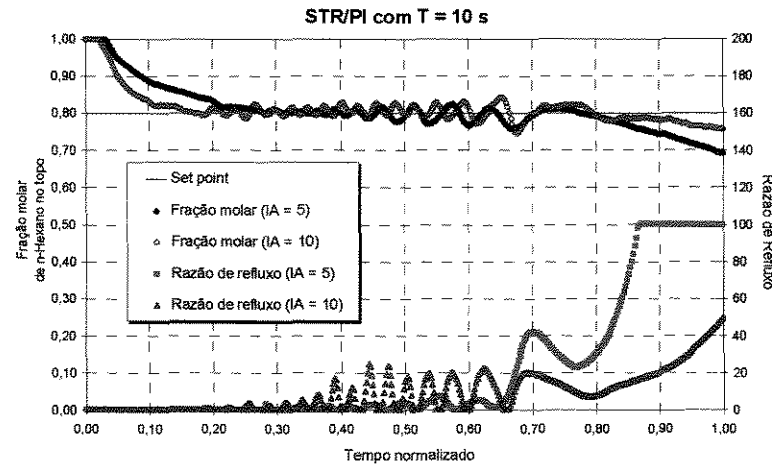


Figura 4.8b - Controlador STR/PI.

4.4.2.2 – TESTE DE ESTABILIDADE

Os comportamentos oscilatórios nos gráficos 4.8a e 4.8b poderiam ser um indicio de instabilidade do sistema caracterizado pela existência de alguma raiz da equação característica discretizada (eq. 4.17) fora do círculo de raio unitário. Sendo assim, realizou-se um teste de estabilidade para cada conjunto de parâmetros do processo obtidos na etapa de identificação do sistema juntamente com os parâmetros do controlador (fig. 4.9).

$$1 + G_p(z)D(z) = 0 \quad (4.17)$$

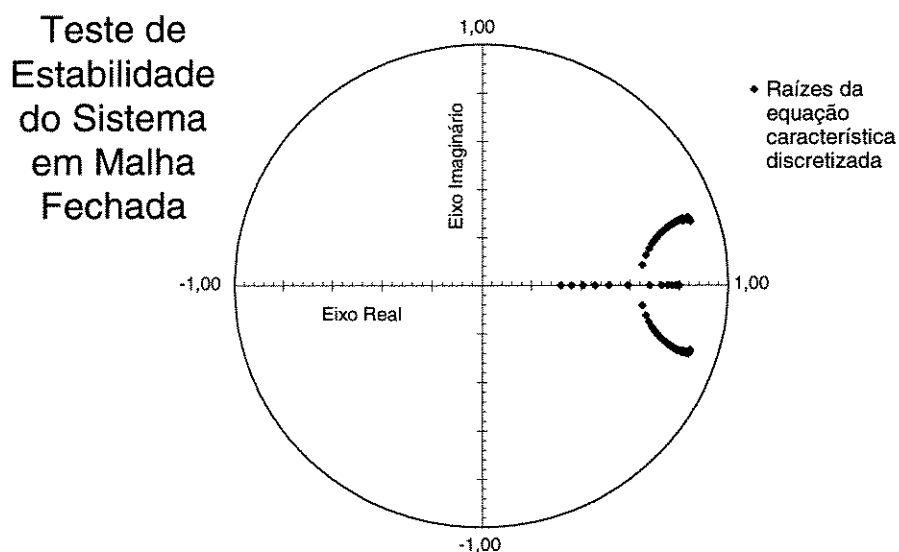


Figura 4.9 – Teste de Estabilidade.

Descartada a hipótese de instabilidade justifica-se o comportamento oscilatório pela ação conjunta de dois fatores existentes no sistema experimental. São eles: a dinâmica não definida da válvula magnética (on-off), fato que poderia ser minimizado se existisse um vaso acumulador no topo da coluna; e, a existência de uma vazão mínima de destilado no sistema, a qual impossibilitava o corte total da vazão de destilado. Apesar disso, as composições médias no balão coletor (ver tabela 4.3) mostram o bom funcionamento do controlador visto que as mesmas situam-se bem próximas aos valores especificados ("set-point").

4.4.2.3 – EFEITO DO INTERVALO DE AMOSTRAGEM

Os sistemas de controle digitais trabalham com valores discretos das variáveis manipulada e controlada. Portanto, um dos mais importantes parâmetros que deve ser escolhido na implementação destes sistemas é o intervalo de amostragem (T).

Uma idéia errada sobre este parâmetro é a de que menores valores deste melhoram o desempenho do controlador. Sabe-se, porém, que pequenos intervalos de amostragem causam interferência na medição das variáveis. Além disso, em sistemas de controle digital direto (DDC) este parâmetro possui um limite inferior que corresponde ao tempo necessário para as medições, os cálculos das ações de controle e atuação sobre o processo.

Dois valores de T foram testados no sistema estudo tanto no STR/PI (BAT's 4 e 7) quanto no STR/Dahlin (BAT's 5 e 8), de acordo com os gráficos 4.10a e 4.10b.

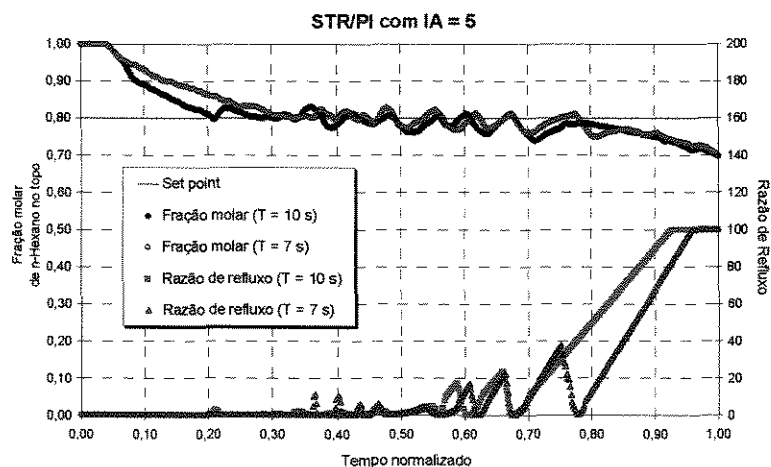


Figura 4.10a - Controlador STR/PI.

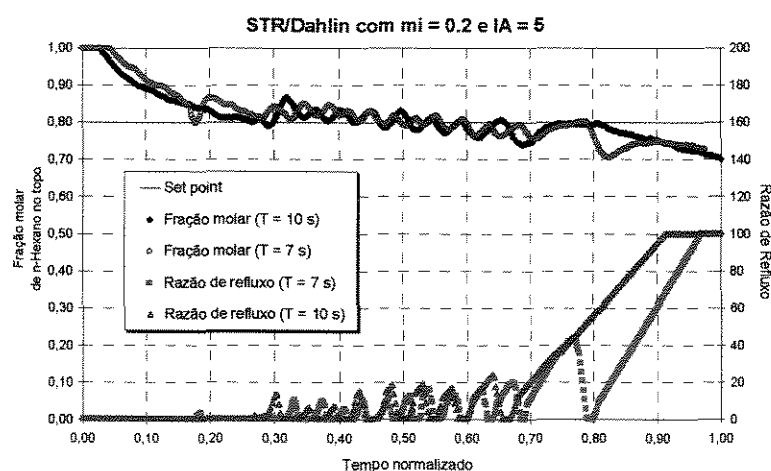


Figura 4.10b - Controlador STR/Dahlin.

Para ambas estratégias de controle houve um aumento no desvio da variável controlada quando utilizado o menor intervalo de amostragem. Contudo, houve uma melhora no comportamento da variável manipulada, indicando como no item anterior uma redução nos custos de operação do processo.

4.4.2.4 – EFEITO DO PARÂMETRO μ

No caso do algoritmo de Dahlin (STR/Dahlin), o parâmetro μ , que corresponde a constante de tempo da resposta desejada do sistema em malha fechada, também precisa ser

determinado. Sabe-se, contudo, que baixos valores de μ acarretam numa resposta mais rápida enquanto altos valores resultam em respostas lentas do sistema em malha fechada.

Para a verificação destes efeitos no sistema estudado dois valores desta constante de tempo (0,2 e 1,0) foram individualmente testadas (BAT's 3 e 5). O gráfico 4.11 ilustra os comportamentos obtidos.

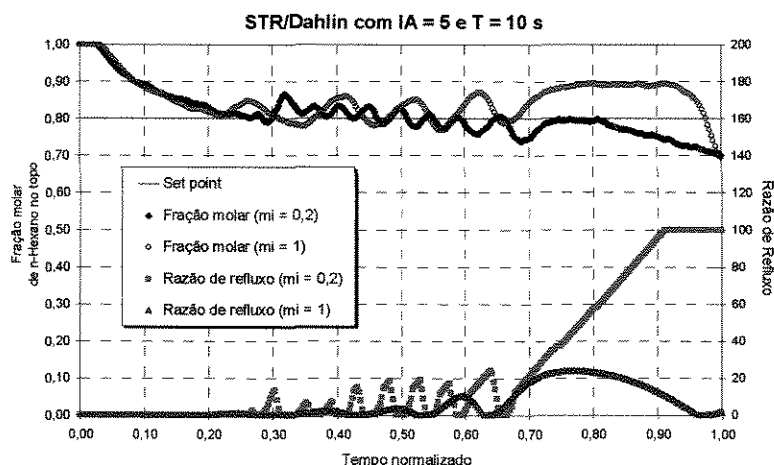


Figura 4.11 - Controlador STR/Dahlin.

Notamos que apesar do melhor comportamento da variável manipulada o sistema com maior μ apresenta uma resposta excessivamente lenta, tornando o sistema de controle menos eficiente. Por outro lado, o comportamento do sistema com um menor μ se mostra mais adequado, mesmo apresentando um maior desvio na variável controlada.

4.4.2.5 – COMPARAÇÃO ENTRE OS CONTROLADORES

Já que os resultados obtidos se mostraram sensivelmente melhores que aqueles alcançados com os controladores convencionais, restou apenas encontrar qual dos controladores adaptativos detalhados anteriormente apresenta melhor desempenho para a coluna de destilação em batelada.

Em primeiro lugar fez-se uma comparação entre os controladores STR e PAC com a mesma estratégia de controle (PI). Os dados da tabela 4.3 (BAT's 7 e 9) juntamente com o gráfico 4.12a mostram uma maior eficiência do controlador STR/PI tendo em vista um comportamento mais suave da variável manipulada e mesmo desvio padrão na variável sob controle (composição no topo). Isto associado ao desnecessário conhecimento das

propriedades e concentração inicial da mistura sendo processada, ao contrário do PAC, tornam o STR bem mais atrativo.

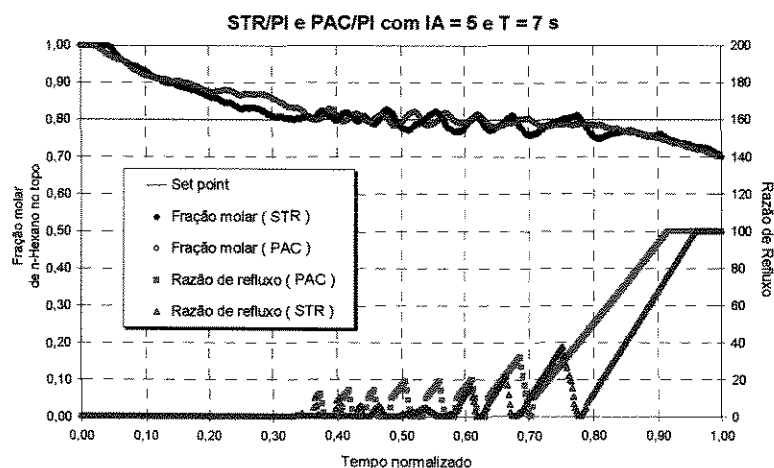


Figura 4.12a - Controlador STR/PI e PAC/PI.

Constatada a maior eficiência do STR em relação ao PAC buscou-se a comparação entre as duas estratégias de controle PI e Dahlin, como pode ser observado nos gráficos 4.12b e 4.12c para diferentes valores do intervalo de amostragem.

Com a ajuda dos gráficos abaixo e dos dados da tabela 4.3 (BAT's 4, 5, 7 e 8) observamos que a estratégia de controle PI é mais eficiente dado a sua atuação mais suave na variável manipulada com aproximadamente o mesmo desvio na variável controlada. Além disso, para mesmas concentrações iniciais o tempo total da batelada com o STR/PI foram menores os obtidos com o STR/Dahlin.

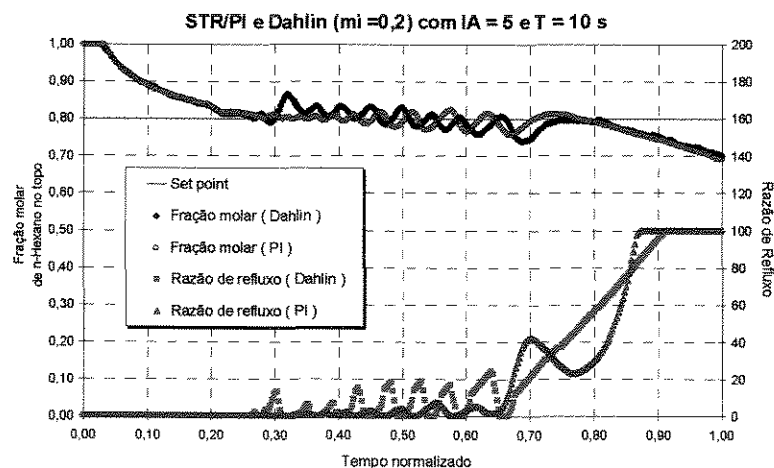


Figura 4.12b - Controlador STR/PI e Dahlin.

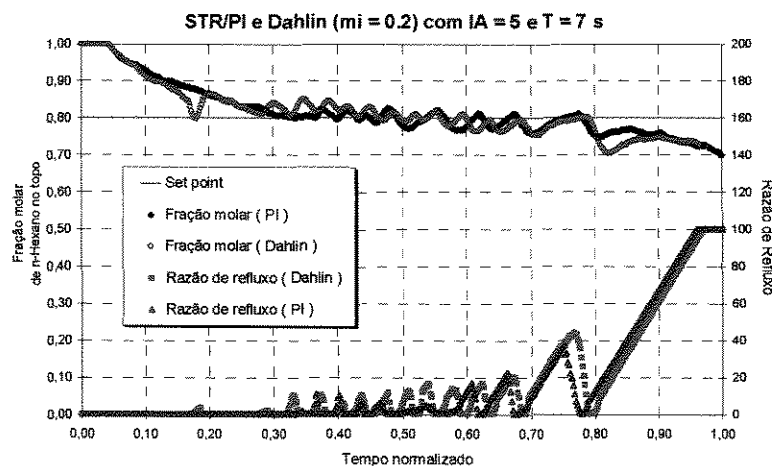


Figura 4.12c - Controlador STR/PI e Dahlin.

Vale observar que em todos os casos estudados acima estudados, independentemente da concentração inicial da mistura inicial carregada na coluna, o valor médio da concentração do produto de topo final sempre esteve situado acima da concentração especificada ("set-point"). Isto pode ser explicado pelo fato da fração molar inicial no topo está bem acima do valor desejado, mostrando que o número de estágios usado (12 pratos) poderia ser reduzido para aproximar esta fração molar inicial do valor desejado.

4.4.2.6 – SISTEMA TERNÁRIO

Com a adequação de parâmetros realizada para a mistura binária, o STR foi usado no controle da composição para uma mistura equimolar de n-hexano/ciclohexano/n-heptano obtendo-se bons resultados. O intervalo de amostragem utilizado foi 10 s com o número instantes de amostragem para identificação do processo igual a 5, como mostra o gráfico 4.13.

Para a inferenciação da composição usou-se uma Rede Neural Artificial (RNA) treinada com dados obtido do simulador da coluna, Fileti (1996), para a referida mistura ternária. A implementação deste sistema bem com o algoritmo de treinamento proposto por Marquardt-Levenberg, Hagan e Menhaj (1994), utilizado serão detalhados no capítulo seguinte.

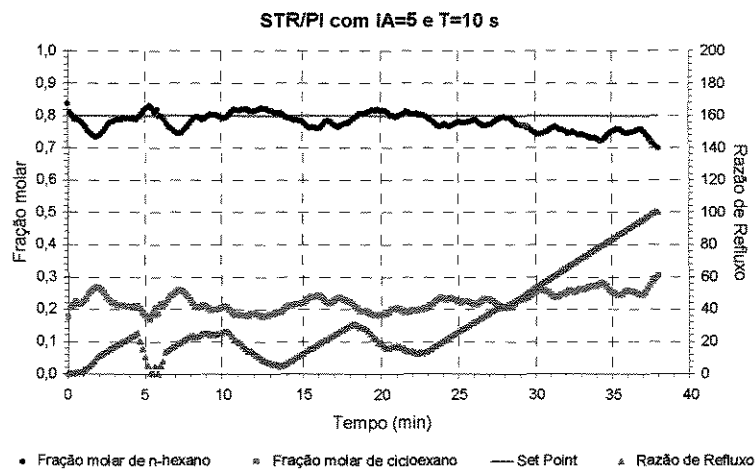


Figura 4.13 - Controlador STR/PI para a mistura ternária.

Observa-se um bom desempenho do sistema de controle em vista dos valores moderados de razão de refluxo visto que as composições médias molares obtidas (80% de n-hexano, 18% de ciclohexano e 2% de n-heptano) encontram-se bem próximas dos valores especificados (80% em n-hexano e 20% em ciclohexano).

Mostra-se, no capítulo seguinte, a implementação e o desempenho do sistema de inferência utilizado.

CAPÍTULO 5
INFERENCIAÇÃO
POR REDES NEURAIS ARTIFICIAIS

5.1 - INTRODUÇÃO

Por várias décadas, cientistas tem buscado entender e modelar a estrutura do cérebro acreditando que o processo de aprendizagem humana pode ser reproduzido por um algoritmo equivalente. Assim, surgiram diversas propostas, baseadas no conhecimento biológico, neste ramo da ciência de Inteligência Artificial (IA). Dentre elas, as Redes Neurais Artificiais (RNA) tem merecido grande interesse nos últimos anos com um aumento significativo no número de artigos, jornais e conferências sobre o assunto.

Tanto a estratégia de processamento paralelo quanto sua capacidade de "aprender" tornam as RNA's bastante úteis e rápidas em aplicações como sistemas de controle e inferenciação, processamento de imagens, entre outras.

De acordo com estes fatos e com os trabalhos de Morris et al.(1991), Morris et al. (1994) e Zhang et al. (1997) este capítulo apresenta os detalhes da implementação do sistema de inferenciação baseado em RNA's.

Inicialmente serão apresentados os elementos constituintes das RNA's (neurônios) e os diversos tipos de conexões entre eles (topologia), citados na literatura. Em seguida, serão mostrados dois procedimentos utilizados para o treinamento das RNA's ("Back-propagation" e "Marquardt-Levenberg"), buscando uma exata compreensão do seu funcionamento, necessária para a sua correta implementação.

Será dado um maior ênfase ao uso das RNA's para sistemas de inferenciação, uma vez que estes se mostram bastante úteis na implementação de sistemas de automação e controle em sistemas dinâmicos quando a medição das variáveis controladas possui um custo elevado ou um atraso associado que inviabilize o ajuste em linha das variáveis manipuladas.

Por último, será mostrado o desenvolvimento e implementação experimental de um sistema de inferenciação de composição de topo, para uma mistura ternária n-hexano/cicloexano/n-heptano, numa coluna de destilação em batelada, utilizando Redes Neurais Artificiais.

5.2 - NEURÔNIOS ARTIFICIAIS OU NÓDULOS

A Rede Neural Artificial possui componentes básicos de processamento chamados de neurônios artificiais ou nódulos. Eles podem ser vistos também como simples funções que

permitem o mapeamento de um espaço n -dimensional, sendo n o número de entradas recebidas das vizinhanças, num espaço unidimensional, ou seja, um valor de saída transmitido às vizinhanças que é função dos valores de entrada recebidos. Os neurônios podem realizar seus cálculos simultaneamente conferindo a Rede Neural Artificial um caracter inerentemente paralelo.

A figura 5.1 mostra duas características típicas de um neurônio artificial. São elas: a atividade " a " (ou estado do nódulo) resultante das entradas; e a saída, que é função da atividade, " $f(a)$ ".

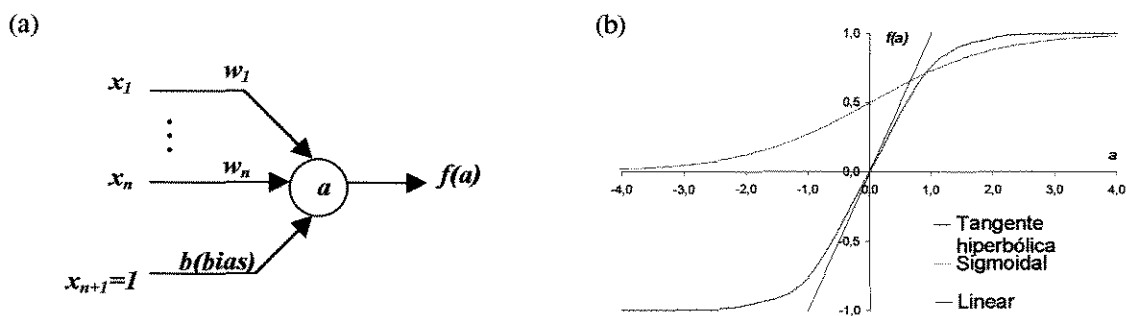


Figura 5.1 – Características do Neurônios Artificiais.

5.2.1. ESTADO DE ATIVAÇÃO

Cada neurônio deve ter uma regra para combinar suas entradas para atingir um novo estado de ativação. Esta atividade é um reflexo da intensidade com que as entradas influenciam o nódulo. Neste trabalho, a atividade do neurônio é dada pela soma ponderada das entradas visto que esta é a mais usada na literatura. Sendo assim, para a atividade no neurônio j tem-se que:

$$a_j = \bar{u}_j^T \cdot \bar{v} = \sum_{i=1}^n w_{ij} \cdot x_i \quad (5.1a)$$

sendo:

$$\bar{v} = (x_1, x_2, \dots, x_n), \text{ o vetor de entrada que reflete o estado do processo;} \quad (5.1b)$$

$$\bar{u}_j^T = (w_{1j}, w_{2j}, \dots, w_{nj}), \text{ o vetor dos pesos ou vetor ganho.} \quad (5.1c)$$

Percebe-se, portanto, que além do vetor de entrada os pesos também afetam o estado e excitação do neurônio. Deste modo, se w_{ij} for ajustado tal que o produto $w_{ij}x_i$ seja positivo e

de alto valor, a tendência é de excitação do neurônio j . Por outro lado, se $w_{ij}x_i$ for negativo, esta entrada ponderada causará a inibição do mesmo neurônio.

Uma vez que a soma ponderada pode assumir valores muito próximos a zero, acrescenta-se, normalmente, mais um termo a atividade do neurônio, chamado de atividade residual (ou bias). Este termo altera o espaço de não-linearidade entre as entradas e saídas dos neurônios e controla a ativação total dos mesmos.

Nilsson (1965) sugere que a inclusão do termo residual (ou bias) seja feita acrescentando-se uma entrada constante (normalmente de valor unitário) ao vetor de entrada do neurônio e mais um fator ponderal ao vetor ganho. Desta forma, a atividade total do neurônio é

$$A_j = \overline{U}_j^T \cdot \overline{V}_{j-1} = \sum_{i=1}^n w_{ij} \cdot x_i + b_j \quad (5.2a)$$

sendo:

$$\overline{V} = (x_1, x_2, \dots, x_n, 1); \quad (5.2b)$$

$$\overline{U}_j^T = (w_{1j}, w_{2j}, \dots, w_{nj}, b_j). \quad (5.2c)$$

5.2.2. FUNÇÃO DE SAÍDA OU FUNÇÃO DE ATIVAÇÃO

A saída de cada neurônio é obtida passando-se a sua atividade para uma função (ou operador) não-linear responsável pela projeção do corrente estado de ativação num sinal de saída. Tipicamente, as funções residuais são utilizadas já que são monotonamente crescentes com valores limitantes. Estas características resultam normalmente em RNA's bem comportadas, além de propiciarem um treinamento mais eficiente e rápido. Contudo, a princípio não existe um critério bem definido para a escolha de tais funções, sendo as mais empregadas mostradas na tabela abaixo:

Tabela 5.1 – Funções de ativação

Nome	Fórmula
Sigmoidal	$f(a) = 1/(1 + \exp(-a))$
Tangente hiperbólica	$f(a) = \tanh(a)$
Gausiana	$f(a) = \exp(-a^2)$
Linear	$f(x) = a$

5.3 - TOPOLOGIA DAS RNA's

A topologia de uma RNA é estabelecida pelo modo como os neurônios artificiais estão interconectados e organizados em camadas. Basicamente, existem três possibilidades para as conexões dos neurônios: intracamada, intercamada e recursiva (figura 5.2).

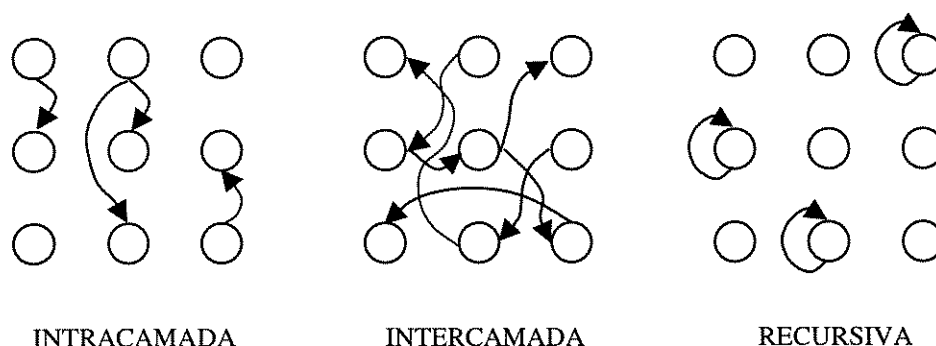


Figura 5.2 – Tipos de Conexões nas RNA's.

Nas conexões intracamadas, a saída de um neurônio numa camada serve de entrada para os outros neurônios desta mesma camada. Já nas conexões intercamada a saída de um neurônio numa camada alimenta os neurônios de outras camadas. De modo distinto dos anteriores, nas conexões recursivas a saída de um neurônio alimenta ele próprio.

Embora existam um sem número de topologias propostas na literatura, Quantrille e Liu (1991) sugerem em seu trabalho que as conexões intercamadas são particularmente importantes em aplicações de engenharia. Este tipo de conexão pode ser dividido em conexões por retroalimentação e por alimentação direta (figura 5.3). Entretanto, os autores acima indicam a conexão por alimentação direta como a mais apropriadas para mapear uma respostas baseadas em informações de entrada.

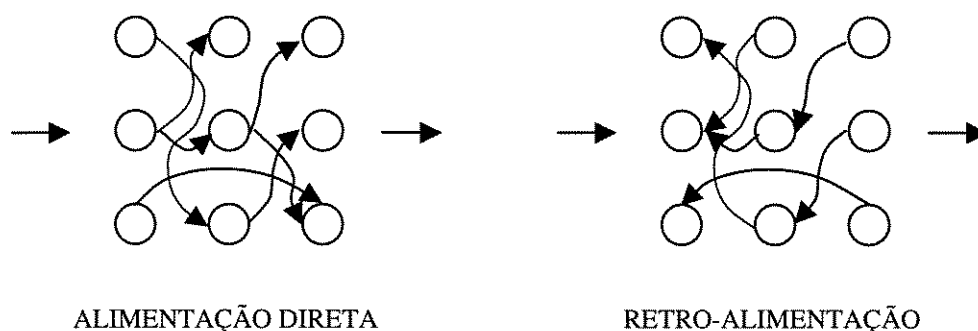


Figura 5.3 – Tipos de Conexões Intercamada.

Já Cybenko (1989) vai mais adiante e sugere que no máximo duas camadas intermediárias seriam necessárias para a maioria dos problemas de engenharia tendo em vista o menor esforço computacional exigido. Ele também afirma que este tipo de rede pode aproximar de uma forma compacta qualquer tipo de função contínua.

Sendo assim, no restante deste trabalho serão discutidas somente as RNA's com conexões intercamada por alimentação direta contendo uma única camada intermediária (RNA's do tipo "Feedforward") tendo em vista sua maior simplicidade e maior número de teorias matemáticas relacionadas.

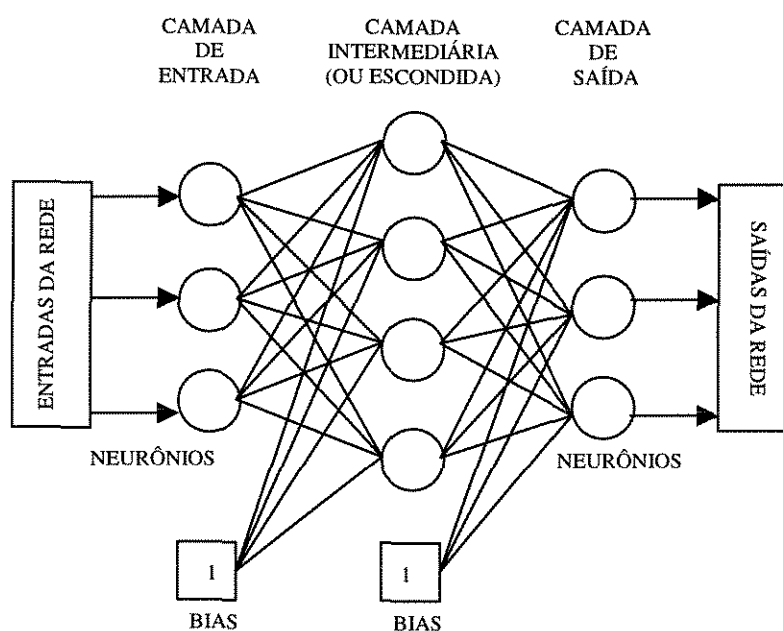


Figura 5.4 – Rede Neural Artificial do Tipo "Feedforward".

5.4 - TREINAMENTO DAS RNA's

A topologia da RNA juntamente com as funções de ativação são os fatores que determinam a precisão e o grau de representação do comportamento do sistema estudado. Contudo, para a utilização das RNA's os fatores ponderais e a ativação residual (ou bias) devem ser determinados por algum mecanismo de treinamento, ajustando-se estes parâmetros até que a resposta calculada, baseada nos dados de entrada, alcance a relação causa-efeito desejada. Esta tarefa consiste basicamente numa otimização não-linear.

5.4.1. ALGORITMO DE RETROPROPAGAÇÃO (OU "BACKPROPAGATION")

A mais simples técnica de otimização utilizada para o treinamento de uma RNA, com conexões intercamadas por alimentação direta, usa o Jacobiano de uma função objetivo para determinar a direção de busca. A soma dos quadrados dos erros entre os valores reais e aqueles preditos pela RNA é normalmente usado como função objetivo (eq. 5.3).

$$Z = \frac{1}{2} \cdot \sum_{q=1}^{Np} e_q^T \cdot e_q = \frac{1}{2} \cdot \sum_{q=1}^{Np} (\bar{r}_q - \bar{p}_q^M)^T \cdot (\bar{r}_q - \bar{p}_q^M) \quad (5.3)$$

sendo:

e_q , vetor erro = $\bar{r}_q - \bar{p}_q$;

$\bar{r}_q$, vetor de saída real do processo;

$\bar{p}_q$, vetor de saída predito pela RNA;

Np , número de vetores de entrada/saída.

Um termo chamado taxa de aprendizagem (α) que influencia a taxa de ajuste dos fatores ponderais (ou pesos) é usado como a base para retropropagar o erro ao longo da RNA conforme a técnica do gradiente descendente, buscando conduzir a função objetivo a um mínimo. Esta técnica foi então chamada de Retropropagação (ou "Backpropagation"). As etapas abaixo dão uma perfeita compreensão do funcionamento deste mecanismo de ajuste dos pesos:

- (1) Inicializam-se os todos os pesos da RNA com valores randômicos;
- (2) Um vetor de entrada é propagado através da rede para fornecer a predição do vetor de saída. Nesta etapa são determinados todas as saídas dos neurônios da rede conforme a equação 5.4 e 5.5:

$$\bar{x}^0 = \bar{v} \quad (5.4)$$

$$\bar{x}^{k+1} = f^{k+1} \left(W^{k+1} \bar{x}^k + \bar{b}^{k+1} \right) \quad (5.5)$$

sendo:

$k = 0, 1, 2, \dots, M-1$, índice para as M camadas de neurônios da RNA;

f^{k+1} , função de ativação dos neurônios da camada $k+1$;

$\bar{x}^k = (x_1, x_2, \dots, x_{Sk})^T$, vetor de saída da camada k ;

$$W^{k+1} = \begin{bmatrix} w_{1,1}^{k+1} & w_{1,2}^{k+1} & \dots & w_{1,Sk}^{k+1} \\ w_{2,Sk}^{k+1} & w_{2,2}^{k+1} & \dots & w_{2,Sk}^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{Sk+1,1}^{k+1} & w_{Sk+1,2}^{k+1} & \dots & w_{Sk+1,Sk}^{k+1} \end{bmatrix}, \text{ matrix dos pesos para camada } k+1;$$

$\bar{b}^{k+1} = (b_1, b_2, \dots, b_{Sk+1})$, vetor de bias para a camada $k+1$;

Sk , número de neurônios da camada k

$\bar{v}$, vetor de entrada da RNA.

(3) Com o vetor de saída predito pela RNA calcula-se a soma do erro quadrático total como descrito na equação 5.3;

(4) Os termos de declínio (δ) das diversas camadas são calculados de acordo com as equações 5.6 e 5.7:

$$\bar{\delta}^M = \dot{F}(\bar{x}^M) \cdot (\bar{r}_q - \bar{p}_q) \quad (5.6)$$

$$\bar{\delta}^k = \dot{F}^k(\bar{x}^k) \cdot W^{k+1T} \cdot \bar{\delta}^{k+1} \quad (5.7)$$

sendo:

$$\dot{F}^k(\bar{x}^k) = \begin{bmatrix} \dot{f}^k(x_1^k) & 0 & \dots & 0 \\ 0 & \dot{f}^k(x_2^k) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dot{f}^k(x_{Sk}^k) \end{bmatrix}, \text{ matriz diagonal das primeiras derivadas};$$

$$\dot{f}^k(x_i^k) = \frac{df^k(x_i^k)}{dx_i^k}, \text{ primeira derivada da função de ativação do neurônio } i.$$

- (5) Conhecendo-se os termos de declínio e as saídas de cada neurônio da RNA, os pesos são ajustados de segundo as equações 5.8 e 5.9:

$$w_{i,j}^k|_{novo} = w_{i,j}^k|_{atual} + \Delta w_{i,j}^k = w_{i,j}^k|_{atual} + \alpha \cdot \delta_i^k \cdot x_j^{k-1} \quad (5.8)$$

$$b_i^k|_{novo} = b_i^k|_{atual} + \Delta b_i^k = b_i^k|_{atual} + \alpha \cdot \delta_i^k \quad (5.9)$$

- (6) Repete-se então as etapas 2 a 5 para todos os Np vetores de entrada/saída utilizados no treinamento, até que o erro quadrático total alcance o critério de convergência desejado.

A ampla utilização deste algoritmo de treinamento pode ser atribuída a sua fácil implementação. Porém, quando a função objetivo aproxima-se de um mínimo local o algoritmo torna-se ineficiente. Na maioria das aplicações de redes neurais um termo de "momentum" (η) é adicionado a equação que altera os pesos de forma que a mudança atual dos mesmos seja também função das alterações passadas (equações 5.10 e 5.11).

$$\Delta w_{i,j}^k|_{novo} = \alpha \cdot \delta_i^k \cdot x_j^{k-1} + \eta \cdot \Delta w_{i,j}^k|_{atual} \quad (5.10)$$

$$\Delta b_i^k|_{novo} = \alpha \cdot \delta_i^k \cdot x_j^{k-1} + \eta \cdot \Delta b_i^k|_{atual} \quad (5.11)$$

Muito embora esta modificação melhore o desempenho do método, este continua exigindo um elevado tempo de convergência e problemas com um grande número de dados de entrada/saída.

5.4.2. ALGORITMO DE LEVENBERG-MARQUARDT

Os métodos Quasi-Newton mostram-se em princípio mais rápidos e mais robustos do que aqueles baseados no gradiente descendente. Com base nisto, Hagan e Menhaj (1994) utilizaram o algoritmo de Levenberg-Marquardt (Marquardt, 1963) para o treinamento de RNA's do tipo "Feedforward" e verificaram seu melhor desempenho para os casos onde existiam apenas algumas centenas de fatores ponderais. Este método consiste numa aproximação do método de Newton e propõe que a atualização dos pesos seja dada pela seguinte equação:

$$\Delta \bar{y} = [J^T(\bar{y}) \cdot J(\bar{y}) + \mu \cdot I]^{-1} \cdot J^T(\bar{y}) \cdot e(\bar{y}) \quad (5.12)$$

sendo:

$$\bar{y} = [w_{1,1}^1, w_{1,2}^1, \dots, w_{S1,R}^1, b_1^1, \dots, b_{S1}^1, w_{1,1}^2, \dots, w_{i,j}^k, \dots, b_{Sk}^k, \dots, b_{SM}^M]^T, \text{ vetor peso;}$$

$$J(\bar{y}) = \begin{bmatrix} \frac{\partial e_1(\bar{y})}{\partial y_1} & \frac{\partial e_1(\bar{y})}{\partial y_2} & \dots & \frac{\partial e_1(\bar{y})}{\partial y_n} \\ \frac{\partial e_2(\bar{y})}{\partial y_1} & \frac{\partial e_2(\bar{y})}{\partial y_2} & \dots & \frac{\partial e_2(\bar{y})}{\partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{Np}(\bar{y})}{\partial y_1} & \frac{\partial e_{Np}(\bar{y})}{\partial y_2} & \dots & \frac{\partial e_{Np}(\bar{y})}{\partial y_n} \end{bmatrix}, \text{ matriz jacobiana dos erros;}$$

$$\frac{\partial e_q(\bar{y})}{\partial w_{i,j}^k} = \delta_i^k \cdot a_j^{k-1} \quad \text{e} \quad \frac{\partial e_q(\bar{y})}{\partial b_i^k} = \delta_i^k;$$

$$q = 1, 2, \dots, Np; \quad k = 0, 1, 2, \dots, M-1; \quad i = 1, 2, \dots, Sk+1; \quad j = 1, 2, \dots, Sk.$$

As etapas realizadas no treinamento são as mesmas já descritas. Além disso, o cálculo dos termos de declínio (δ) usam as mesmas equações do método de Retropropagação (eq. 5.6 e 5.7) apenas modificando a equação 5.6 para

$$\bar{\delta}^M = \dot{F}(\bar{x}^M) \quad (5.13)$$

Como pode-se notar o cálculo da matriz jacobiana do erros entre os valores reais e os preditos pela RNA é a etapa que mais exige esforço computacional. Para problemas com grandes quantidades de dados a memória requerida para tal processamento torna-se um fator limitante na utilização deste algoritmo.

5.5 - INFERENCIAÇÃO

Muitos problemas existentes nas indústrias químicas e bioquímicas estão relacionados com a determinação em linha de parâmetros e variáveis que mostrem o comportamento do processo. Numa boa parte dos casos, o problema principal deve-se a difícil medição a uma taxa adequada de tais variáveis de modo a permitir seu eficaz controle. Isto pode ocorrer devido ao atraso necessário para análises em laboratório ou por causa do limitado e longo ciclo de amostragem do analisador em linha. Além disso, analisadores rápidos e eficientes são bastante caros.

Uma opção bastante interessante para resolver estes problemas é utilizar um modelo juntamente com medidas secundárias do processo para estimar as variáveis principais que são difíceis ou mesmo impossíveis de se medir em linha. Uma vez que a relação entre as variáveis desejadas e as variáveis secundárias medidas em linha tenha sido encontrada, o modelo resultante pode ser utilizado dentro de um esquema de controle para melhorar o ajuste do processo. Este princípio é conhecido como Estimação Inferencial (ou "Soft-sensing").

Historicamente, com diversos graus de sucesso, modelos lineares, modelos adaptativos e modelos mecânicos específicos tem sido usados na inferenciação de variáveis do processo. Entretanto nos últimos anos o uso das RNA's para tal atividade tem se mostrado muito bem sucedido (Tham et al. 1991 e Willis et al. 1991).

Segundo Morris et al. (1994), um problema em potencial com o uso de inferenciadores baseados em Redes Neurais Artificiais está no banco de dados necessário para o treinamento da rede. Na maior parte dos casos, os dados foram coletados do processo em operação com algum controle já em malha fechada. Desta forma, o modelo resultante do treinamento da rede será identificado com dados correlacionados e não representa o comportamento básico do processo. Quando tal modelo é usado dentro de um sistema de controle "feedback", ele estará sujeito a novos dados correspondentes a novas condições do processo, que também estão correlacionados, causando uma degradação nas predições do mesmo. Neste caso é importante identificar uma nova rede usando os novos dados para melhorar as predições da rede.

Utilizando RNA's do tipo "feedforward", Willis et al. (1992) desenvolveram um sistema de inferenciação em linha da massa de microorganismos na fermentação da penicilina e da composição de metanol no topo e no fundo numa coluna de destilação contínua. Neste trabalho, o ajuste dos sistemas foi realizado por um controlador PI convencional usando os dados inferidos do processo alcançando excelente resultados.

No caso do processo de destilação em batelada, muitos sistemas de inferenciação tem sido propostos na literatura (Quintero-Marmol e Luyben, 1992; Yu e Luyben, 1987), vista que seu comportamento transiente e não-linear dificulta medidas rápidas de composição no interior da coluna. Entretanto, a maioria deles consideram os perfis de composição no topo de uma coluna de destilação em batelada como essencialmente binário, independentemente do número de componentes na mistura inicial. Esta hipótese facilita a solução do problema de inferência, mas não pode ser aplicado a qualquer condição operacional da separação multicomponente ou mesmo no caso de misturas com volatilidades relativas muito próximas.

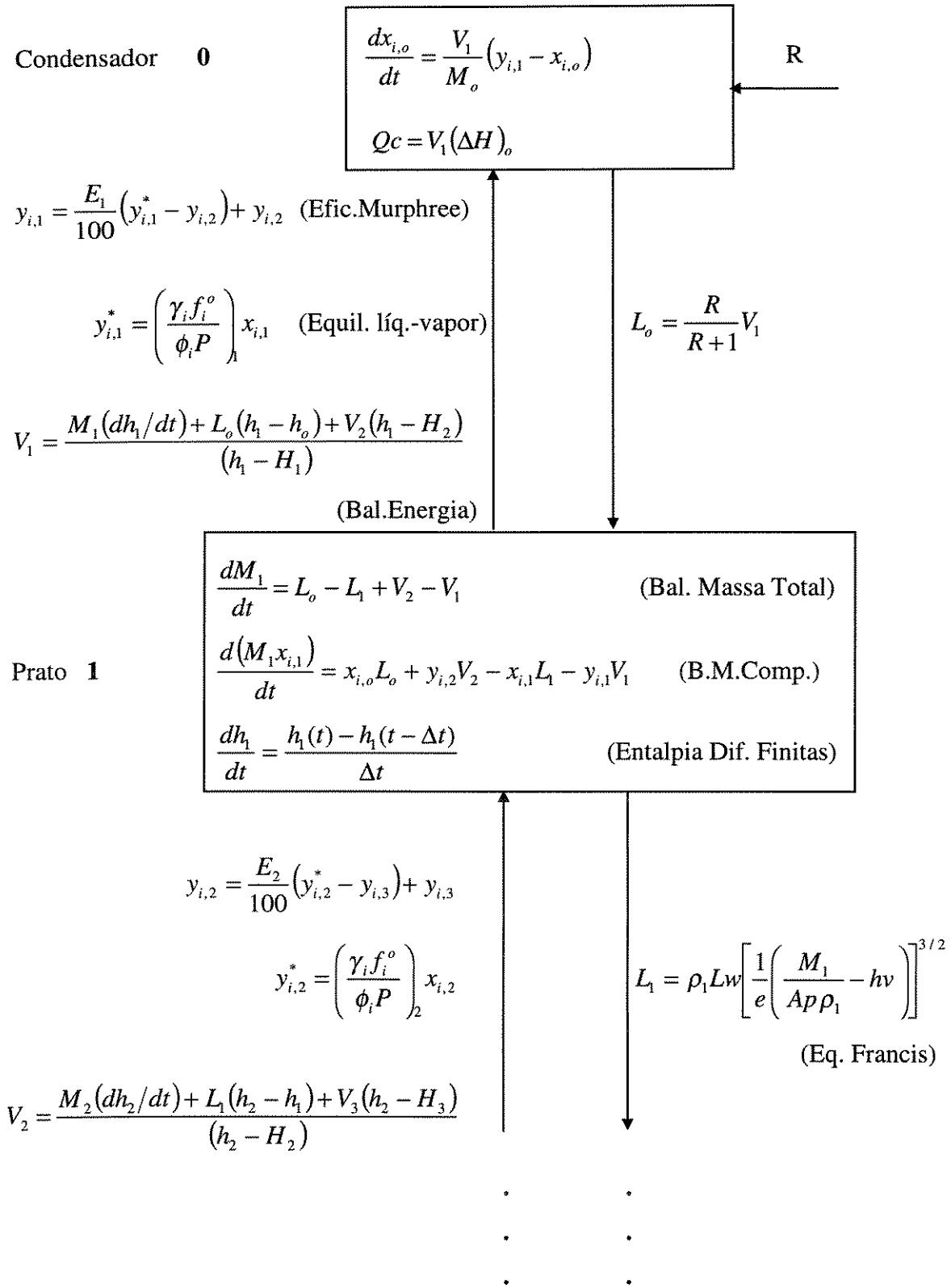
Diante disso, um sistema de inferenciação de composição de topo numa coluna de destilação em batelada utilizando Redes Neurais Artificiais foi desenvolvido e implementado experimentalmente utilizando como vetor de entrada as temperatura de topo e fundo medidas e a razão de refluxo interna calculada pelo algoritmo de controle. O algoritmo de treinamento de Marquardt-Levenberg foi usado "off-line", com dados obtidos por diversas simulações do processo, para a obtenção dos parâmetros da rede.

5.6 – SIMULADOR

Tendo por objetivo a geração de dados de operação para posterior treinamento das Redes Neurais Artificiais, utilizou-se o programa computacional de simulação COLBAT (Fileti, 1992). Apresentam-se aqui apenas as principais suposições feitas e o equacionamento básico (fig. 5.5) do simulador da coluna de pratos utilizada para o processo de destilação em batelada.

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

O método numérico utilizado na integração das equações diferenciais ordinárias foi Runge-Kutta de 4ª ordem.



(continua)

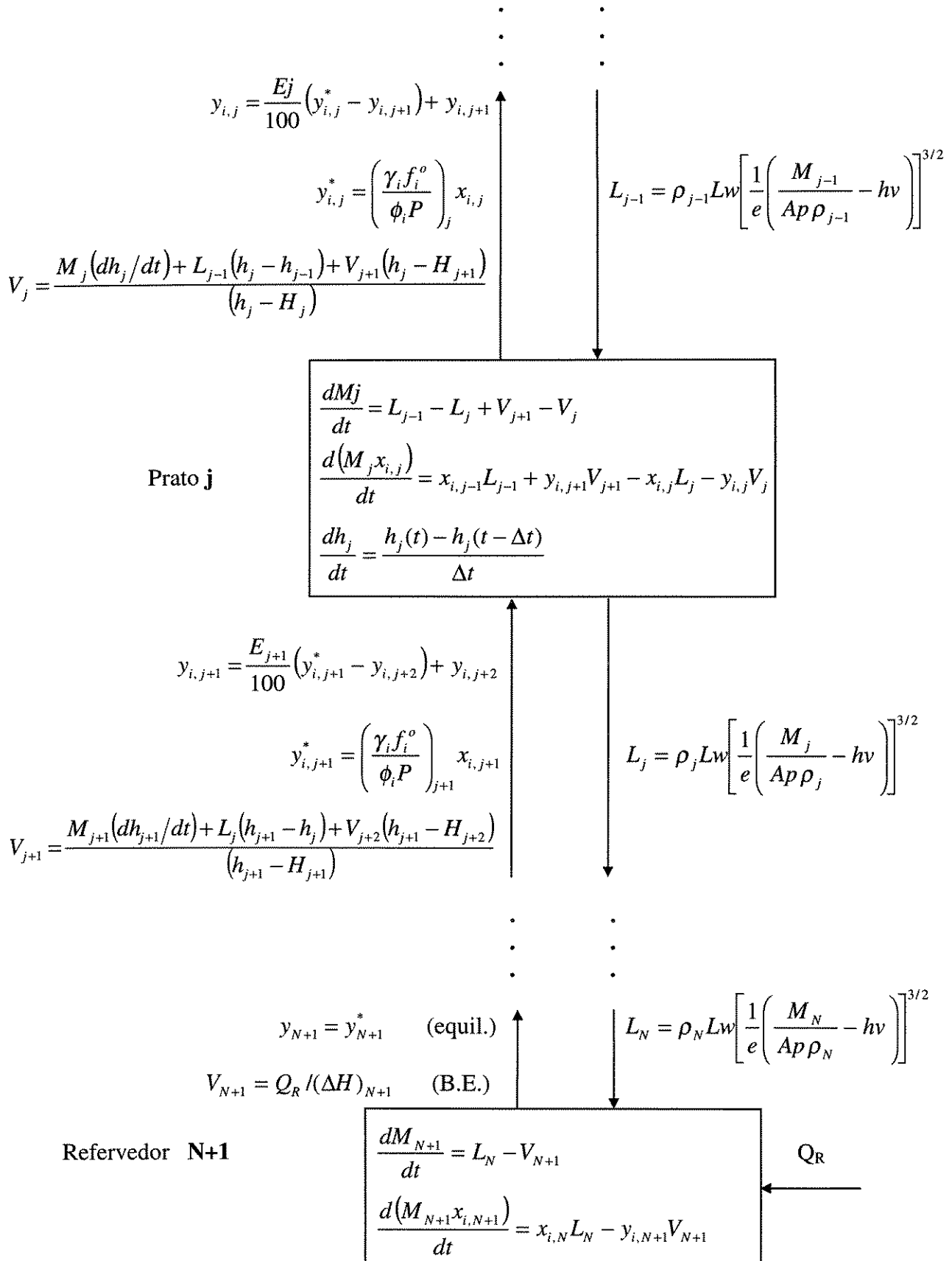


Figura 5.5 - Modelo fenomenológico de destilação multicomponente em batelada.

5.7 - IMPLEMENTAÇÃO

Tendo em vista a falta de dados experimentais disponíveis sobre o sistema em estudo, diversas simulações com diferentes estratégias de refluxo foram realizadas para a construção de um banco de dados contendo as temperaturas de topo e fundo, as composições dos componentes no topo e a razão de refluxo. A escolha da razão de refluxo e das temperaturas está baseada em sua estreita relação com as composições no topo da coluna.

Devido a diferença na ordem de grandeza entre as variáveis, as temperaturas e a razão de refluxo foram normalizadas visando evitar problemas de convergência no treinamento. Sendo assim, as temperaturas normais de ebulição dos componentes mais e menos volátil, n-hexano e n-heptano, respectivamente, foram utilizadas conforme as expressões abaixo:

$$\overline{T}_{\text{topo}} = \frac{T_{\text{topo}} - T_{\text{eb}_1}}{T_{\text{eb}_3} - T_{\text{eb}_1}} \quad (5.14a)$$

$$\overline{T}_{\text{fundo}} = \frac{T_{\text{fundo}} - T_{\text{eb}_1}}{T_{\text{eb}_3} - T_{\text{eb}_1}} \quad (5.14b)$$

Já a razão de refluxo (R) foi normalizada através do refluxo interno visto que os valores desta última estão dentro do intervalo [0,1]. Deste modo,

$$R_i = \frac{L}{V} = \frac{L}{L+D} = \frac{R}{1+R} \quad (5.15)$$

Utilizando-se o pacote computacional MATLAB 4.2b, a arquitetura da rede foi determinada através das seguintes etapas:

- 1ª) A razão de refluxo interno e as temperaturas de topo e fundo foram usadas como dados de entrada da rede. Isto definiu o número de neurônios da camada de entrada como sendo três.
- 2ª) O número dos neurônios de saída está relacionado com o número de componentes da mistura destilada. Cada neurônio de saída representa a fração molar de um componente específico no topo (ou fundo). Neste trabalho foram usados dois neurônios de saída para a mistura n-hexano(1)/ciclohexano(2)/n-heptano(3). A composição do terceiro componente foi determinada por diferença. Deste modo,

$$\sum_{i=1}^3 x_i = 1 \quad \therefore \quad x_3 = 1 - \sum_{i=1}^2 x_i \quad (5.16)$$

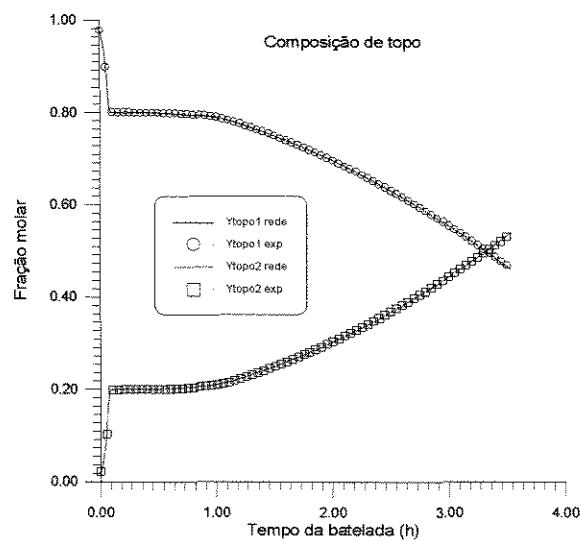
- 3ª) Dados da literatura tem mostrado que uma camada intermediária é suficiente para a maioria dos casos. Sendo assim, restava encontrar o número de neurônios da mesma. Várias tentativas foram realizadas com 20, 30 e 50 neurônios tendo-se obtido 20 neurônios como o número mais adequado para o presente trabalho.
- 4ª) Como função de ativação foram testadas as funções: sigmoideal, tangente hiperbólica e linear (ver tabela 5.1). Tanto o tempo necessário para convergência quanto o número de iterações apresentaram significativa redução quando se utilizou a função tangente hiperbólica nos neurônios da camada intermediária e linear nos neurônios da camada de saída.

5.8 - RESULTADOS

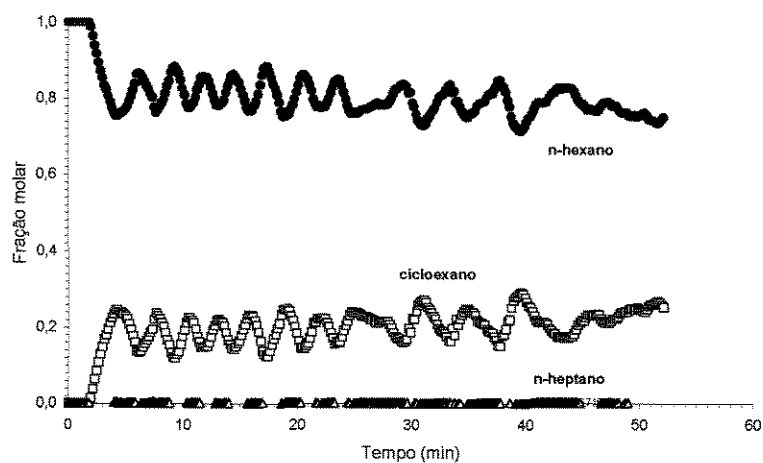
Para a definição da arquitetura da rede bem como as funções de ativação realizou-se o treinamento "off-line" da mesma, com os dois algoritmos citados anteriormente tendo-se usado um total 2500 vetores de entrada/saída. Foi estabelecido como critério de convergência que o somatório dos erros quadráticos entre os dados de saída e aqueles fornecidos pela rede não fosse superior a 10^{-7} .

O algoritmo de Marquardt-Levenberg atingiu mais rapidamente a convergência e portanto foi usado para a obtenção dos parâmetros da rede. Estabeleceu-se a função tangente hiperbólica como a função de ativação da camada intermediária e a função linear para a camada de saída. Já a topologia mais adequada para a rede foi determinada, após várias tentativas, como sendo 3-20-2.

Os pesos e bias obtidos no treinamento foram usados numa nova simulação com dados não "vistos" no primeiro treinamento para verificar o desempenho da inferenciação da composição, como pode ser observado na figura 5.6. Nota-se uma boa concordância dos dados obtidos do simulador com os fornecidos pela rede neural.

**Figura 5.6** – Resultados de Simulações.

O sistema experimental foi então testado para a verificar a eficácia da inferenciação "on-line" das composições de topo para que as ações de controle fossem aplicadas em tempo real. O "set-point" escolhido foi 80% de n-hexano na retirada do primeiro componente. Como pode ser visto nas figuras 5.7 e 5.8, o sistema de controle atuou de maneira satisfatória nos dois casos.

**Figura 5.7** – Resultados Experimentais.

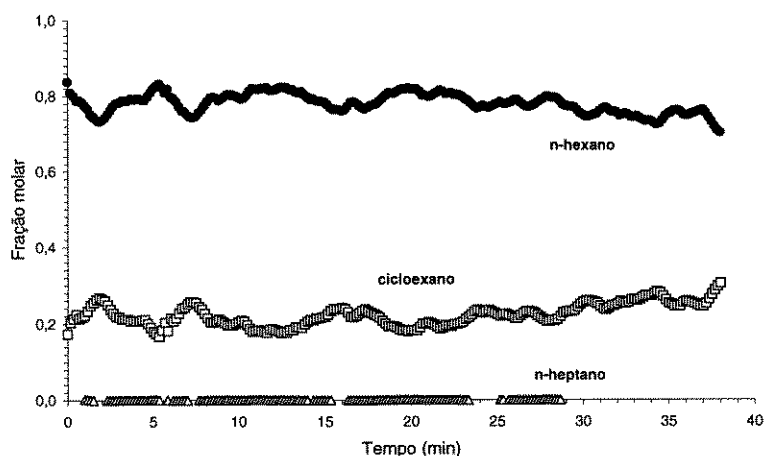


Figura 5.8 – Resultados Experimentais.

Durante a operação da coluna foram coletadas amostras no topo da coluna para análise cromatográfica posterior. Além disso, ao término da batelada a composição final do destilado foi calculada com a ajuda de dados de vazão média (figuras 5.9 e 5.10) e uma amostra deste também foi analisado. O resultados das análises confirmam as estimativas obtidas pela rede neural, conforme pode ser visto na tabela 5.2.

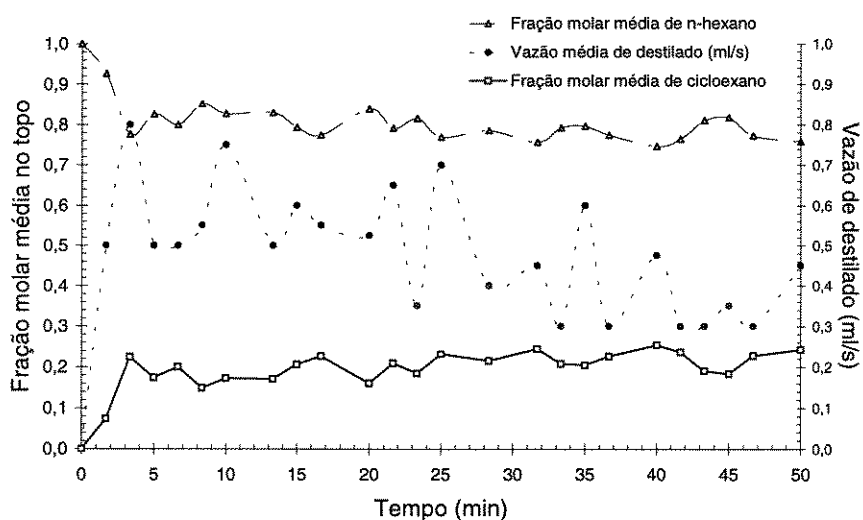


Figura 5.9 – Fração molar média e Vazão média de destilado (BAT1).

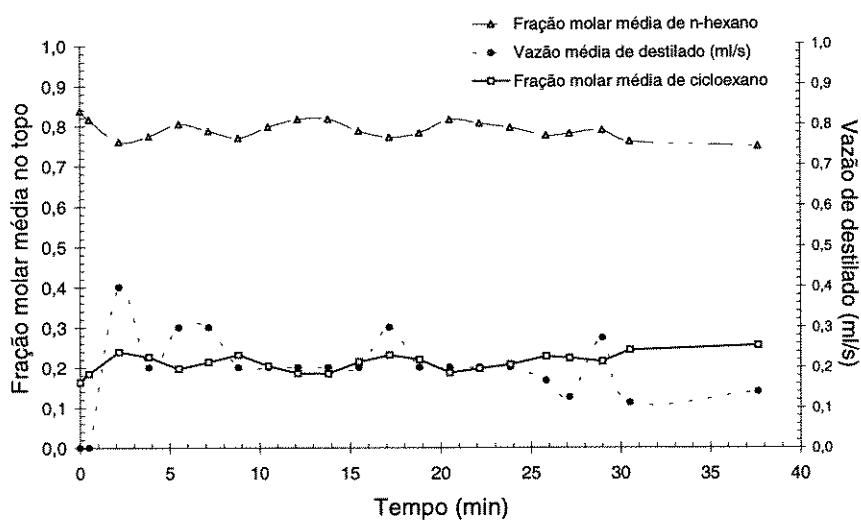


Figura 5.10 – Fração molar média e Vazão média de destilado (BAT2).

Tabela 5.2 – Resultados das análises cromatográficas

BAT	Tempo (min)	Fração molar de n-hexano		Fração molar de ciclohexano		Fração molar de n-heptano	
		Infe-rência	Croma-tógrafo	Infe-rência	Croma-tógrafo	Infe-rência	Croma-tógrafo
1	29,33	0,82	0,79	0,18	0,17	0,00	0,02
	44,00	0,79	0,77	0,21	0,19	0,00	0,02
	média	0,82	0,80	0,18	0,17	0,00	0,03
2	5,33	0,78	0,75	0,22	0,22	0,00	0,02
	13,17	0,79	0,76	0,21	0,21	0,00	0,02
	média	0,75	0,74	0,25	0,23	0,00	0,03

Os resultados obtidos expõem claramente a eficácia das RNA's na inferenciação da composição de topo. Esta ferramenta pode ainda ser utilizada para o controle deste processo conforme Fileti (1996). Contudo, este tema não foi abordado no presente trabalho.

CAPÍTULO 6

CONCLUSÕES E

SUGESTÕES FINAIS

6.1 – CONCLUSÕES

Apesar dos grandes avanços nas técnicas de controle, os controladores por retroalimentação convencionais (P, PI e PID) ainda são amplamente utilizados na indústria química. Portanto, mesmo em aplicações de sistemas de controle digital direto (DDC), eles ainda são encarados como a primeira alternativa no ajuste do processo. Contudo, no caso de processos altamente transientes e não-lineares, como a destilação em batelada, seu desempenho mostra-se claramente insatisfatório uma vez que a sintonia destes controladores é realizada através de valores médios dos parâmetros do modelo do processo, obtidos em diferentes instantes da batelada. Como no decorrer do tempo estes parâmetros atingem valores bem distintos do médio, o ajuste torna-se inadequado.

Frente às exigências da destilação em batelada, os algoritmos de controle adaptativo surgem como uma alternativa mais apropriada. Novamente, imbuídos na busca de métodos de fácil compreensão e implementação o Controlador Adaptativo Programável (PAC) aparece como opção inicial. Entretanto, o tempo consumido no projeto do sistema de controle, através de exaustivas simulações, e a necessidade do conhecimento prévio das propriedades dos componentes da mistura a ser processada, tornam o PAC aceitável apenas nos casos onde a coluna é sempre usada na destilação de misturas com a mesma composição inicial e com propriedades bem conhecidas. Por outro lado, satisfeitas as condições anteriores, seu desempenho mostra-se satisfatório visto que os cálculos realizados em linha são extremamente rápidos.

Outra forma de ajustar os parâmetros do modelo do processo é através da identificação recursiva em linha dos mesmos num algoritmo de Controle Auto-ajustável (STR). Para tanto, a relação entre as entradas e saídas do processo precisam ser estabelecidas por um modelo do tipo polinomial discreto. Em virtude disso, utilizou-se neste trabalho o fato da coluna em batelada poder ser aproximada por uma função transferência de primeira ordem com tempo morto para a obtenção do modelo discretizado. Nota-se, porém, que em nenhum momento foi exigido o conhecimento das propriedades da mistura ou o uso de simulações do processo. E, além disso, a estrutura do STR permitiu que várias leis de controle fossem implementadas. Deste modo, um controlador PI na forma velocidade e o algoritmo de Dahlin foram individualmente testados, tendo o primeiro obtido um melhor desempenho.

Por sua independência de um amplo conhecimento do processo e fácil implementação tanto de leis de controle quanto do modelo do processo conclui-se que o Controlador Auto-ajustável (STR) desponta como a melhor alternativa no controle do processo estudado. Contudo, deve-se utilizar métodos robustos na etapa de identificação do processo já que é através dela que os parâmetros do modelo proposto para o processo são adequadamente determinados, possibilitando assim a utilização das mais diversas estratégias de controle devidamente atualizadas em tempo real. Além disso, o STR pode ser implementado tanto em Sistemas Digitais de Controle Distribuído (SDCD) quanto em sistemas de Controle Digital Direto (DDC).

No caso de misturas multicomponentes, para um controle eficiente da coluna em batelada, além do sistema de controle, um mecanismo apropriado para a inferência de composição torna-se imprescindível tanto na redução de custos quanto na redução dos atrasos associados a análises de composição. Em vista disso, o sistema de inferenciação baseado nas Redes Neurais Artificiais aqui desenvolvido mostrou-se bastante eficaz na determinação "on-line" da composição de topo, além da sua fácil implementação junto ao STR. Contudo, a utilização destas ferramentas requer a existência de um banco de dados suficientemente amplo, abrangendo, se possível, toda a faixa das condições de operação.

A etapa de treinamento da RNA pode ser significativamente melhorada com a utilização de algoritmos mais sofisticados. Mas, a medida que o número de dados aumenta estes métodos exigem um maior esforço computacional. Portanto, uma análise do processo para determinar as principais variáveis envolvidas é de suma importância na implementação prática das RNA's.

Os sistemas de controle e inferenciação desenvolvidos tornaram o equipamento de destilação em batelada convencional mais eficiente e fácil de operar. Isto está de acordo com os resultados obtidos através de simulações em trabalhos anteriores. Além disso, visto que tanto o STR quanto as RNA's não necessitam do conhecimento das propriedades físicas do sistema, a utilização destas ferramentas em sistemas de controle de processos químicos torna-se bastante ampla.

6.2 – SUGESTÕES PARA FUTUROS TRABALHOS

Demonstrada a eficiência dos sistemas de controle e inferenciação propostos na destilação em batelada de misturas aproximadamente ideais (n-hexano/cicloexano/n-heptano), a verificação do desempenho destes sistemas na destilação de misturas mais complexas, surge como sugestão inicial para trabalhos futuros.

Já a ampla capacidade de ajuste do STR na destilação em batelada sugere a sua utilização no controle da potência de aquecimento juntamente com a razão de refluxo, consistindo num sistema tipo MIMO ("Multiple Input – Multiple Output"). Para tanto, técnicas de desacoplamento de malhas devem ser levadas em consideração buscando-se ações de controle adequadas.

Por outro lado, fica claro a importância do teste do sistema de controle proposto em ações supervisórias e não só regulatórias, além de seu uso em outros processos transientes e não-lineares para a obtenção sistemas de controle digitais diretos mais eficientes.

Muito embora tenhamos usado um simulador para a obtenção dos dados necessário para a implementação das RNA's, podemos utilizar os dados experimentais obtidos ao longo da batelada para enriquecer o banco de dados de treinamento das redes e melhorar a eficiência do sistema de inferenciação desenvolvido.

O emprego das RNA's em sistemas de controle baseados em modelo pode ser amplamente agilizado através de treinamentos "off-line" com algoritmos de treinamento mais sofisticados que o da retropropagação. É possível assim, a obtenção de ações mais rápidas e eficazes de controle dada a rapidez dos cálculos em tempo real realizados pela rede.

Anexo A

OBTENÇÃO DO

MODELO DISCRETO

DO PROCESSO

A função transferência de pulso entre as variáveis controlada (y) e manipulada (u) pode ser determinada através da transformada Z, conforme a equação 4.7. Entretanto, para a sua obtenção algumas adaptações são necessárias visto que a ordem de grandeza do tempo morto (θ) é desconhecida. Tais alterações consistem na utilização do conceito da transformada Z modificada.

$$Z[y(t - \theta)] = \sum_{n=0}^{\infty} y(nT - \theta) \cdot z^{-n} \quad (\text{A.1})$$

Considerando $\theta = (nd + \Delta)T$, com $0 < \Delta < 1$ e nd inteiro, substituímos em A.1 obtendo

$$Z[y(t - \theta)] = z^{-nd} \cdot \left\{ z^{-1} \cdot \sum_{l=0}^{\infty} y(lT + mT) \cdot z^{-l} \right\} = z^{-nd} \cdot Z_m[y(t)] \quad (\text{A.2})$$

sendo: $l = n - nd - 1$;

$$m = 1 - \Delta;$$

$$Z_m[y(t)] = z^{-1} \cdot \sum_{l=0}^{\infty} y(lT + mT) \cdot z^{-l} = \hat{y}(z, m), \text{ transformada Z modificada.}$$

Portanto, aplicando a equação A.2 na equação 4.7 temos

$$Z\{G_p(s)\} = \frac{K}{\tau} \cdot Z\left\{ \frac{e^{-\theta s}}{s + \frac{1}{\tau}} \right\} = \frac{K}{\tau} \cdot z^{-nd} \cdot Z_m\left\{ \frac{1}{s + \frac{1}{\tau}} \right\} = \frac{K}{\tau} \cdot z^{-nd} \cdot \frac{e^{-mT/\tau} \cdot z^{-1}}{1 - e^{-T/\tau} \cdot z^{-1}} \quad (\text{A.3})$$

Por outro lado, podemos escrever

$$m = 1 - \Delta = 1 - \left(\frac{\theta}{T} - nd \right) = \frac{(1 + nd) \cdot T - \theta}{T} \quad (\text{A.4})$$

Substituindo A.4 em A.3 e rearrajando-se a expressão resultante obtém-se

$$\frac{\hat{y}(z, m)}{u(z, m)} = z^{-nd} \cdot \frac{b_1 \cdot z^{-1}}{1 - a_1 \cdot z^{-1}} \quad (\text{A.5})$$

sendo:

$$a_1 = \exp\left(\frac{-T}{\tau}\right) \quad (\text{A.6})$$

$$b_1 = \frac{K}{\tau} \cdot \exp\left(\frac{-(1+nd) \cdot T - \theta}{\tau}\right) \quad (\text{A.7})$$

$$nd = \text{parte inteira da divisão} \left(\frac{\theta}{T} \right) \quad (\text{A.8})$$

Deste modo, o modelo discreto para o sistema será

$$\hat{y}(z, m) \cdot (1 - a_1 \cdot z^{-1}) = b_1 \cdot z^{-1-nd} \cdot u(z, m)$$

ou seja,

$$y_i = a_1 \cdot y_{i-1} + b_1 \cdot u_{i-1-nd} + bias \quad (\text{A.9})$$

sendo: y_{ss} valor da variável controlada no estado estacionário;

u_{ss} valor da variável manipulada no estado estacionário;

$$bias = y_{ss} \cdot (1 - a_1) - b_1 \cdot u_{ss} \quad (\text{A.10})$$

Não havendo estado estacionário definido, podemos supor que a média aritmética dos últimos valores das variáveis serve como uma boa aproximação deste ponto.

Anexo B

MÉTODO DE

PROGRAMAÇÃO

QUADRÁTICA

SUCESSIVA

O desenvolvimento de algoritmos para a resolução de problemas em programação não-linear, cuja função objetivo e as restrições são não-lineares, é uma tarefa difícil. Tanto que neste campo de pesquisa novos trabalhos surgem continuamente. Porém, podemos considerar que existem basicamente três categorias de métodos:

- 1ª) Métodos de penalização e barreira;
- 2ª) Programação quadrática sucessiva;
- 3ª) Gradiente reduzido generalizado.

Neste trabalho, foi utilizado um método de programação quadrática sucessiva aplicado a mínimos quadrados, disponível na biblioteca matemática NAG, na etapa de identificação do processo necessária a implementação do controle auto-ajustável (STR). A seguir serão apresentadas as idéias básicas que caracterizam este método.

Inicialmente tomemos o seguinte problema com restrições não-lineares:

$$\begin{aligned} & \underset{x}{\text{Minimizar}} \ f(x) \\ & \text{sujeito a } c_{\pi}(x) \geq 0 \end{aligned} \tag{B.1}$$

sendo: $f : \mathbb{R}^n \rightarrow \mathbb{R}$;

$c_{\pi} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ com $m < n$;

$\pi = \{i_1(x), \dots, i_{s(x)}(x)\}$ é o conjunto de índices correspondentes às restrições de desigualdade que estão ativas no ponto x ($c_i(x) = 0$).

Assim, dado o problema com restrições (B.1), associa-se a ele uma função objetivo sem restrições chamada de função Lagrangeana definida como:

$$L(x, \lambda) = f(x) - \sum_i \lambda_i \cdot c_i(x) \tag{B.2}$$

onde: λ_i são denominados multiplicadores de Lagrange.

Para determinar um mínimo local da função (B.2), o seu gradiente deverá ser igualado a zero.

$$\bar{\nabla}L(x^{(k)} + \delta x, \lambda^{(k)} + \delta \lambda) = 0 \quad \therefore \quad \bar{\nabla} = \begin{pmatrix} \nabla_x \\ \nabla_\lambda \end{pmatrix} \quad (\text{B.3})$$

Por outro lado, uma expansão em série de Taylor de (B.2) em torno do ponto $(x^{(k)}, \lambda^{(k)})$ fornece:

$$\bar{\nabla}L(x^{(k)} + \delta x, \lambda^{(k)} + \delta \lambda) = \bar{\nabla}L(x^{(k)}, \lambda^{(k)}) + \bar{\nabla}^2 L(x^{(k)}, \lambda^{(k)}) \cdot \begin{pmatrix} \delta x \\ \delta \lambda \end{pmatrix} + \dots \quad (\text{B.4})$$

Desprezando-se os termos de ordem superior e sabendo que o termo a esquerda da igualdade deve ser zero (B.3) obtemos:

$$\bar{\nabla}^2 L(x^{(k)}, \lambda^{(k)}) \cdot \begin{pmatrix} \delta x \\ \delta \lambda \end{pmatrix} = -\bar{\nabla}L(x^{(k)}, \lambda^{(k)}) \quad (\text{B.5})$$

ou seja,

$$\begin{bmatrix} W^{(k)} & -A^{(k)} \\ -A^{(k)\top} & 0 \end{bmatrix} \cdot \begin{pmatrix} \delta x \\ \delta \lambda \end{pmatrix} = -\begin{pmatrix} g^{(k)} - A^{(k)} \cdot \lambda^{(k)} \\ -c^{(k)} \end{pmatrix} \quad (\text{B.6})$$

sendo: $A^{(k)}$, a matriz Jacobiana das restrições $c(x)$ avaliada em $x^{(k)}$;

$W^{(k)}$, a matriz Hessiana de B.2, dada por $W^{(k)} = \nabla^2 f(x^{(k)}) - \sum_i \lambda_i^{(k)} \cdot \nabla^2 c_i(x^{(k)})$;

Uma forma mais conveniente de expressar B.6 é

$$\begin{bmatrix} W^{(k)} & -A^{(k)} \\ -A^{(k)\top} & 0 \end{bmatrix} \cdot \begin{pmatrix} \delta^{(k)} \\ \lambda^{(k+1)} \end{pmatrix} = \begin{pmatrix} -g^{(k)} \\ c^{(k)} \end{pmatrix} \quad (\text{B.7})$$

sendo: $\lambda^{(k+1)} = \lambda^{(k)} + \delta \lambda$ e $\delta^{(k)} = \delta x$.

Resolvendo a equação B.7 determinam-se $\delta^{(k)}$ e $\lambda^{(k+1)}$. E, deste modo, $x^{(k+1)}$ é dado por

$$x^{(k+1)} = x^{(k)} + \delta^{(k)} \quad (\text{B.8})$$

Fornecendo estimativas iniciais $(x^{(I)}, \lambda^{(I)})$ e utilizando as equações B.7 e B.8 o método pode determinar a solução do problema B.1. Contudo, o esforço excessivo na resolução do

sistema B.7 sugere o uso de técnicas alternativas.

Uma das formas possíveis de facilitar a busca da solução do problema proposto (B.1) é associá-lo a um subproblema que envolva apenas a minimização de uma função quadrática. Assim sendo, vamos considerar o seguinte subproblema

$$\begin{aligned} & \underset{\delta}{\text{Minimizar}} \quad q^{(k)}(\delta) \\ & \text{sujeito a} \quad h_{\pi}^{(k)}(\delta) \geq 0 \end{aligned} \tag{B.9}$$

sendo: $q^{(k)}(\delta) = f^{(k)} + g^{(k)\top} \cdot \delta + \frac{1}{2} \delta^T \cdot W^{(k)} \cdot \delta$;

$$h_{\pi}^{(k)}(\delta) = A^{(k)\top} \cdot \delta + c^{(k)};$$

π é o conjunto dos índices das restrições ativas em $\delta(h_i(\delta) = 0)$.

Novamente, fornecidas as estimativas iniciais $(x^{(1)}, \lambda^{(1)})$, podemos resolver o problema. Contudo, o novo método iterativo sugerido é agora composto pelos seguintes passos:

- 1ª) Dados $x^{(k)}$ e $\lambda^{(k)}$, estimativas do mínimo x^* e λ^* , resolver o subproblema B.9, determinando $\delta^{(k)}$ e $\lambda^{(k+1)}$;
- 2ª) Definir $x^{(k+1)} = x^{(k)} + \alpha^{(k)} \delta^{(k)}$, sendo $\alpha^{(k)}$ escolhido de maneira a fazer decrescer uma função de mérito adequada e verificar o critério de convergência; Se a convergência for atingida $x^{(k+1)}$ é a melhor aproximação para x^* , senão vá para o passo 3;
- 3ª) Calcular uma aproximação de $W^{(k+1)}$ e retornar a primeiro passo.

Pode-se demonstrar (Fletcher, 1987) que existindo um único mínimo para B.9 satisfazendo as condições de primeira e segunda ordem (Kuhn-Tucker) para todo k , então, a sequência de interação assim definida é idêntica àquela fornecida pelo problema original B.1.

Desta forma, recomenda-se a resolução do subproblema quadrático não só pela maior simplicidade nos cálculos como também pela garantia da existência de um mínimo visto que podem existir soluções de B.7 que satisfaçam as condições de Kuhn-Tucker mas que não são o mínimo.

Anexo C

PROGRAMA COM

O SISTEMA DE CONTROLE

DESENVOLVIDO

Listagem do Programa Principal responsável pela coordenação das atividades de aquisição e armazenamento de dados, filtragem digital, inferenciação, identificação e controle do processo, além da interface gráfica.

```

program BATCHDIS
  use msflib
  use portlib
  implicit none
  include 'batchdis.fi'
  character(50)      fparam(5)
  character          resp
  logical(4)         corte,test
  integer            i,na,nat,cna,ced,nc,nt,nv,ia,np,iparam(9),med
  parameter          (nc=8,nt=8,nv=8,ia=20)
  real(8)            t,dt,tD,tL,Ta,rparam(9),R,yss,uss,desv,yant,Te(nt), &
                    &      cp(nc,nt),sp(nc),y(ia),u(ia),erro(ia),par(nc,nv)

  call mkwindows()
  call inputdata(iparam,rparam,fparam,nc,nv,par,sp,test)
  np=iparam(4)
  Ta=rparam(1)
  desv=rparam(2)
  na=1
  nat=0
  cna=1
  ced=1
  dt=0.d0
  tD=0.d0
  tL=0.d0
  R=0.d0
  y=0.d0
  u=0.d0
  uss=0.d0
  yss=0.d0
  erro=0.d0
  call initgraph(1)
  call selectcanal(128)
  write(10,'(18(A5,2X))' 'na','Tt','Tf','yt','yf','t','tD','dt','tL','R', &
        &      'Kp','Tp','td','erro','y','u','yss','uss'

  i=0
  do while(.true.)
    call tread(i,nt,iparam(2),Te,test)
    call graphic(1,nat,ced,nc,nt,nv,np,iparam,par,cp,sp,Te,R)
    if(i.le.10) then
      call selectcanal(0)
    end if
    if(.not.test) then
      call sleepqq(int(Ta*500.))
    end if
    call selectcanal(128)
  end do

```

```

        if(.not.test) then
            call sleepqq(int(Ta*500.))
        end if
        write(10,'(I4,2X,2(F12.4,2X))' i,Te(1),Te(2)
        write(*,'(I4,2X,2(F12.4,2X))' i,Te(1),Te(2)
        if(Te(1).gt.64.d0) then
            med=mod(i,2)
            if(med.ne.0) then
                write(*,*) 'Deseja iniciar o controle (S/N) ?'
                read(*,*) resp
                if(resp.eq.'s'.or.resp.eq.'S') exit
            end if
        end if
        i=i+1
    enddo
    call clearscreen($GCLEARSCREEN)
    call initgraph(2)
    t=timef()
    do while(.not.corte)
        call selectcanal(128)
        do while(.true.)
            if((t.ge.(nat*Ta)).or.test) then
                call tread(nat,nt,iparam(2),Te,test)
                call yinfer(ced,nc,nt,iparam(2),Te,cp)
                call control(na,cna,ced,nc,nt,nv,ia,t,yant,R,cp,sp,par,erro,yss, &
                    &      uss,y,u,iparam,rparam,fparam)
                exit
            else
                t=timef()
            end if
        enddo
        call graphic(2,nat,ced,nc,nt,nv,np,iparam,par,cp,sp,Te,R)
        nat=nat+1
        call valve(t,Ta,tD,tL,dt,R,test)
        write(10,'(I4,2X,17(F14.4,2X))' nat,Te(1),Te(2),cp(ced,1),cp(ced,2),t,tD, &
            & dt,tL,R,par(ced,1),par(ced,2), &
            & par(ced,3),erro(cna),y(cna),u(cna),yss,uss
        if((dabs(sp(ced)-cp(ced,1)).gt.0.1d0).and.(sp(ced).gt.cp(ced,1))) corte=.true.
        t=timef()
    enddo
    write(10,'(27H Tempo total da batelada = , F12.4,3H s.))' t
    write(10,'(19H Arquivo de dados: ,A80)' fparam(1)
    close(10)
    if(test) close(15)
    stop 'Fim do Programa'
end program

```

 Protótipos (ou interfaces) para as funções implementadas em outros arquivos.

```

interface
  subroutine
    control(na,cna,ced,nc,nt,nv,ia,t,yant,R,cp,sp,par,erro,yss,uss,y,u,iparam,rparam,fparam)
    character(50) fparam(5)
    integer na,cna,ced,nc,nt,nv,ia,iparam(9)
    real(8) t,yant,R,cp(nc,nt),sp(nc),par(nc,nv),erro(ia),yss,uss,y(ia),u(ia),rparam(9)
  end subroutine
end interface
!
interface
  subroutine graphic(tipo,na,ced,nc,nt,nv,np,iparam,par,cp,sp,Te,R)
  integer tipo,na,ced,nc,nt,nv,np,iparam(9)
  real(8) par(nc,nv),cp(nc,nt),sp(nc),Te(nt),R
  end subroutine
end interface
!
interface
  subroutine initgraph(tipo)
  integer tipo
  end subroutine
end interface
!
interface
  subroutine inputdata(iparam,rparam,fparam,nc,nv,par,sp,test)
  logical(4) test
  character(50) fparam(5)
  integer nc,nv,iparam(9)
  real(8) par(nc,nv),sp(nc),rparam(9)
  end subroutine
end interface
!
interface
  subroutine mkwindows()
  end subroutine
end interface
!
interface
  subroutine selectcanal(p)
  integer p
  end subroutine
end interface
!
interface
  subroutine solve(par,nc,nv,ia,ced,na,cna,yss,uss,y,u,iparam,rparam,fparam)
  character(50) fparam(5)
  integer nc,nv,ia,ced,na,cna,iparam(9)
  real(8) par(nc,nv),uss,yss,y(ia),u(ia),rparam(9)
  end subroutine

```

```

end interface
!
interface
    subroutine valve(t,Ta,tD,tL,dt,R,test)
    real(8) t,Ta,tD,tL,dt,R
    logical(4) test
    end subroutine
end interface
!
interface
    subroutine yinfer(ced,nc,nt,cnt,Te,cp)
    integer ced,nc,nt,cnt
    real(8) Te(nt),cp(nc,nt)
    end subroutine
end interface
!
interface
    subroutine tread(na,nt,cnt,Te,test)
    integer na,nt,cnt
    real(8) Te(nt)
    logical(4) test
    end subroutine
end interface

```

Função em fortran 90 que seleciona o canal analógico a ser lido.

```

subroutine selectcanal(p)
    implicit none
    include 'cfiles.fi'
    integer(1) p
    call write_dig(p)
    return
end subroutine

```

Função em fortran 90 para a leitura e filtragem das temperaturas de topo e fundo.

```

subroutine tread(na,nt,cnt,Te,test)
    implicit none
    ! ms$attributes C,EXTERN :: offset
    character(1) offset
    include 'cfiles.fi'
    integer na,nt,cnt
    real(8) Te(nt)
    logical(4) test
    integer p,i,j,nl1,nl2
    real(8) a,leit,soma,leitdig,vfa1(2),vfa2(2)
    parameter (a=0.33d0,nl1=2.d+4,nl2=1.d+4)
    save vfa1,vfa2
    if(test) then

```

```

        read(15,*) Te(1),Te(2)
    else
        do i=1,cnt
            if(i.eq.1) then
                p=128+2
            else
                p=128+1
            end if
            call selectcanal(p)
            if(.not.test) then
                call sleepqq(250)
            end if
            p=p-128
            select case(na)
                case(0:1)
                    soma=0.d0
                    do j=1,nl1
                        soma=soma+read_anl(char(0),offset)
                    enddo
                    leitdig=soma/nl1
                    vfa2(p)=vfa1(p)
                    vfa1(p)=leitdig
                case default
                    soma=0.d0
                    do j=1,nl2
                        soma=soma+read_anl(char(0),offset)
                    enddo
                    leit=soma/nl2
                    leitdig=leit*(a**2.d0)+(1.0d0-a)*2.d0* &
                    &      vfa1(p)-((1.d0-a)**2.d0)*vfa2(p)
                    vfa2(p)=vfa1(p)
                    vfa1(p)=leitdig
            end select
            Te(i)=1.5d2*(leitdig-1050.)/(4220.-1050.)
        enddo
    end if
    return
end subroutine

```

Funções responsáveis pela inferenciação baseada nas Redes Neurais Artificiais.

```

subroutine yinfer(ced,nc,nt,Te,cp,R)
    implicit none
    integer      ced,nc,nt
    real(8)      Te(nt),cp(nc,nt),R
    integer      i
    select case(ced)
        case(1)
            call neuralnet(ced,nc,nt,Te,cp,R)
        case(2)

```

```

        call neuralnet(ced,nc,nt,Te,cp,R)
    end select
    do i=1,nc
        if(cp(i,1).gt.1.d0) cp(i,1)=1.d0
        if(cp(i,1).lt.0.d0) cp(i,1)=0.d0
    enddo
    return
end subroutine
!
subroutine neuralnet(ced,nc,nt,Te,cp,R)
    implicit none
    integer      ced,nc,nt
    real(8)      Te(nt),cp(nc,nt)
    integer      nin,nhid,nout
    parameter    (nin=3,nhid=20,nout=2)
    real(8)      input(nin),outnet(nout),w1(nhid,nin),w2(nout,nhid), &
        &        b1(nhid),b2(nout),a1(nin),a2(nhid),a3(nout)
    integer      i,j
!
!   Neuronio      | Variavel
!   nin=1         |   Rint=R/(1+R)
!   nin=2         |   Tfunco(K) norm.
!   nin=3         |   Ttopo(K) norm.
!   nout=1        |   Yhexano (topo)
!   nout=2        |   Ycicloexano (topo)
!   Dados: Teb1 = 68.78oC;
!               Teb2 = 80.75oC;
!               Teb3 = 98.41oC.
!
    input(1)=(R/(1.d0+R))
    input(2)=(Te(2)-341.93d0)/(98.41d0-68.78d0)
    input(3)=(Te(1)-341.93d0)/(98.41d0-68.78d0)
    open(1,FILE='pesos.dat',STATUS='OLD')
    do i=1,nhid
        read(1,*) (w1(i,j),j=1,nin)
    enddo
    do j=1,nhid
        read(1,*) (w2(i,j),i=1,nout)
    enddo
    do i=1,nhid
        if(i.le.nout) then
            read(1,*) b1(i),b2(i)
        else
            read(1,*) b1(i)
        endif
    enddo
    close(1)
    a1=0.d0;a2=0.d0;a3=0.d0
    a1=input
    do i=1,nhid
        do j=1,nin

```

```

                a2(i)=a2(i)+w1(i,j)*a1(j)
            enddo
        enddo
        a2=dtanh(a2+b1)
        do i=1,nout
            do j=1,nhid
                a3(i)=a3(i)+w2(i,j)*a2(j)
            enddo
        enddo
        a3=a3+b2
        outnet=a3
        cp(1,1)=outnet(1)
        cp(2,1)=outnet(2)
        cp(3,1)=1.d0-cp(1,1)-cp(2,1)
    end subroutine

```

Função para o acionamento da válvula magnética.

```

subroutine valve(t,Ta,tD,tL,dt,R,test)
    use portlib
    implicit none
    real(8)      R,t,Ta,tD,tL,dt
    logical(4)   test
    dt=timef()-t
    tL=Ta*R/(1.+R)-dt
    if(tL.gt.0.) then
        if(.not.test) then
            call sleepqq(int(tL*1000.))
        end if
        tL=tL+dt
    else
        tL=dt
    endif
    tD=Ta-tL
    if(tD.gt.1.) then
        call selectcanal(0)
        if(.not.test) then
            call sleepqq(int(tD*1000.))
        end if
    endif
    return
end subroutine

```

Função para a abertura das janelas onde os gráficos são inseridos.

```

subroutine mkwindows()
    use msflib
    implicit none
    integer(4)      result

```

```

        logical          mode
        type(qwinfo)     winfo
        type(windowconfig) screen
!
! Set highest resolution graphics mode.
    screen.numxpixels=-1
    screen.numypixels=-1
    screen.numtextcols=-1
    screen.numtextrows=-1
    screen.numcolors=-1
    screen.fontsize=-1
    screen.title= "MENSAGENS DE ERRO"C !BLANK
    mode=setwindowconfig(screen)
    if(.not.mode) stop 'ERROR: CANNOT SET GRAPHICS MODE'
    open(UNIT=12,FILE='USER',TITLE='TELA DE ACOMPANHAMENTO')
    open(UNIT=13,FILE='USER',TITLE='DADOS DE ENTRADA')
!
! Determine the minimum and maximum dimensions.
    mode=getwindowconfig(screen)
    winfo.TYPE = QWIN$MAX
    result=SETWSIZEQQ(QWIN$FRAMEWINDOW,winfo)
    return
end subroutine

```

Função responsável pela construção dos eixos dos gráficos de controle.

```

subroutine initgraph(tipo)
    use msflib
    implicit none
    integer          tipo
    integer(2)       i,ix,iy,stt
    character(5)     cy
    real(8)          ry
    type(xycoord)    xy
    type(fontinfo)   fi
    stt=focusqq(12)
    stt=setcolor(15)
    stt=initializefonts()
    if(stt.le.0) write(*,*)'INITIALIZEFONTS ERROR.'
    if(grstatus().ne.$GROK) write(*,*)'INITIALIZEFONTS GRSTATUS ERROR.'
    stt=setfont('t"Arial"h12w8i')
    stt=getfontinfo(fi)
    stt=rectangle($GBORDER,75,30,485,430)
    call setgtxtrotation(0)
    select case(tipo)
        case(1)
            call moveto(120,10,xy)
            call outgtext('TEMPERATURAS DE TOPO E FUNDO')
        case(2)
            call moveto(125,10,xy)

```

```

        call outgtext('COMPOSIÇÕES DE TOPO E FUNDO')
    end select
!
! Caracterização do eixo Y.
    ry=0.d0
    stt=setfont('t"Arial"h8w6')
    do i=0,400,40
        iy=430-i
        call moveto(70,iy,xy)
        stt=lineto(75,iy)
        call moveto(485,iy,xy)
        stt=lineto(491,iy)
        call moveto(41,iy-5,xy)
        write(cy,'(F5.1)') ry
        call outgtext(cy)
        call moveto(488,iy-5,xy)
        write(cy,'(F5.1)') ry
        call outgtext(cy)
        if(tipo.eq.1) then
            ry=ry+10.d0
        else
            ry=ry+1.d-1
        endif
    end do
!
! Caracterização do eixo X.
    do i=0,416,10
        ix=i+75
        call moveto(ix,430,xy)
        stt=lineto(ix,435)
    end do
!
! Título do eixo X.
    stt=setfont('t"Arial"h12w8i')
    call setgtextrotation(0)
    call moveto(250,456,xy)
    call outgtext("Tempo(s)")
!
! Título do eixo Y.
    call setgtextrotation(900)
    if(tipo.eq.1) then
        call moveto(15,260,xy)
        call outgtext("Temperatura(°C)")
    else
        call moveto(15,260,xy)
        call outgtext('Composição(%')
    endif
!
! Construção do gráfico de R.
    if(tipo.eq.2) then

```

```

        stt=setcolor(15)
        stt=rectangle($GBORDER,565,30,600,430)
        call moveto(582,30,xy)
        stt=lineto(582,430)
        call setgtextrotation(0)
        stt=setfont('t"Arial"h8w6')
        ry=0.d0
        do i=0,400,40
            iy=430-i
            call moveto(560,iy,xy)
            stt=lineto(565,iy)
            call moveto(600,iy,xy)
            stt=lineto(605,iy)
            call moveto(535,iy-5,xy)
            write(cy,'(F4.1)') ry
            call outgtext(cy)
            call moveto(605,iy-5,xy)
            write(cy,'(F5.1)') 10.d0*ry
            call outgtext(cy)
            ry=ry+1.d0
        end do
        stt=setfont('t"Arial"h12w8i')
        call setgtextrotation(900)
        call moveto(525,260,xy)
        call outgtext('Razão de refluxo')
    endif
    return
end subroutine

```

Função responsável pela plotagem dos dados na tela.

```

subroutine graphic(tipo,na,ced,nc,nt,nv,np,iparam,par,cp,sp,Te,R)
    use msflib
    implicit none
    integer      tipo,na,ced,nc,nt,nv,np,iparam(9)
    real(8)      par(nc,nv),cp(nc,nt),Te(nt),sp(nc),R
    integer(4)    result
    integer(2)    i,j,ct,ix,ixa,iya(3),iy(3),stt
    character(15) cy
    type(xycoord) xy
    save         ixa,iya,ct
    result=setactiveqq(12)
    call setgtextrotation(0)
    if(tipo.eq.1) then
        do i=1,iparam(2)
            iy(i)=430-int(Te(i)*4.0d0)
        enddo
    else
        if(na.eq.0) ct=0
        do i=1,iparam(2)

```

```

        iy(i)=430-int(cp(ced,i)*400.d0)
    enddo
    iy(3)=430-int(sp(ced)*400.d0)
end if
if(ct.eq.0) ix=75
if(iy(1).le.30) iy(1)=31
if(iy(2).le.30) iy(2)=31
if(iy(1).ge.430) iy(1)=429
if(iy(2).ge.430) iy(2)=429
if((ct.le.np).and.(ix.lt.485)) then
    ix=ix+10
    if(ct.eq.np) ix=ix-1
    if((ix.eq.85).and.(ct.eq.0)) then
        ixa=ix-9
        iya=iy
    end if
    ct=ct+1
else
    ct=1
    ix=85
    ixa=ix-9
    stt=setcolor(0)
    stt=rectangle($GFILLINTERIOR,76,31,484,429)
end if
stt=setcolor(10)
call moveto(ixa,iya(1),xy)
stt=lineto(ix,iy(1))
stt=setcolor(12)
call moveto(ixa,iya(2),xy)
stt=lineto(ix,iy(2))
ixa=ix
iya(1:2)=iy(1:2)
if(tipo.eq.2) then
!
! Set Point do comp. que esta sendo retirado.
    stt=setcolor(0)
    call moveto(76,iya(3),xy)
    stt=lineto(484,iya(3))
    if(iy(3).le.30) iy(3)=31
    if(iy(3).ge.430) iy(3)=429
    stt=setcolor(14)
    call moveto(76,iy(3),xy)
    stt=lineto(484,iy(3))
!
! Exibe as barras indicativas da razão de refluxo.
    stt=setcolor(0)
    stt=rectangle($GFILLINTERIOR,570,31,578,429)
    stt=rectangle($GFILLINTERIOR,587,31,595,429)
    if(R.lt.10.) then
        iy(4)=int(430-40.0d0*R)

```

```

        else
            iy(4)=31
        end if
        if(iy(4).eq.30) iy(4)=31
        if(iy(4).eq.430) iy(4)=429
        stt=setcolor(3)
        stt=rectangle($GFILLINTERIOR,570,iy(4),578,429)
        if(R.le.100.) then
            iy(4)=int(430-4.0d0*R)
        else
            iy(4)=31
        end if
        if(iy(4).eq.30) iy(4)=31
        if(iy(4).eq.430) iy(4)=429
        stt=rectangle($GFILLINTERIOR,587,iy(4),595,429)
        iya(3:4)=iy(3:4)
    !
    ! Exibe os parametros do modelo (Kp,Tp,td).
        stt=setcolor(0)
        stt=rectangle($GFILLINTERIOR,76,405,484,428)
        stt=setcolor(15)
        call moveto(133,405,xy)
        call outgtext('Kp')
        call moveto(271,405,xy)
        call outgtext('Tp')
        call moveto(409,405,xy)
        call outgtext('td')
        j=1
        do i=83,359,138
            call moveto(i,420,xy)
            write(cy,'(ES12.4)') par(ced,j)
            call outgtext(cy)
            j=j+1
        enddo
    !
    ! Exibe o numero de amostragem.
        if(na.ne.0) then
            stt=setcolor(0)
            call moveto(580,450,xy)
            write(cy,'(I4)') (na-1)
            call outgtext(cy)
        end if
        stt=setcolor(14)
        call moveto(580,450,xy)
        write(cy,'(I4)') na
        call outgtext(cy)
    end if
    return
end subroutine

```

Função responsável pela entrada das estimativas iniciais e demais parâmetros do modelo.

```

subroutine inputdata(iparam,rparam,fparam,nc,nv,par,sp,test)
  use msflib
  implicit none
  character(50)      fparam(5)
  integer            nc,nv,iparam(9)
  real(8)            rparam(9),par(nc,nv),sp(nc)
  logical(4)        test
  character(1)       key
  integer(2)         i,mode,opcao
  integer(4)         stt
  type(windowconfig) screen
  stt=focusqq(13)
  write(13,'(56H INÍCIO DE OPERAÇÃO DA COLUNA DE DESTILAÇÃO EM &
    & BATELADA.)')
  write(13,'(32H DESEJA TESTAR O PROGRAMA (S/N)?)')
  read(13,*) key
  if((key.eq.'S').or.(key.eq.'s')) then
    test=.true.
    write(13,'(26H NOME DO ARQUIVO DE TESTE:~)')
    read(13,'(A50)') fparam(5)
    open(UNIT=15,FILE=fparam(5),STATUS='UNKNOWN')
  else
    test=.false.
  end if
  write(13,'(18H ENTRADA DE DADOS:~)')
  write(13,'(18H (1) VIA ARQUIVO;~)')
  write(13,'(18H (2) VIA TECLADO;~)')
  write(13,'(23H (3) SAIR DO PROGRAMA.~)')
  write(13,'(27H QUAL SUA OPÇÃO (1,2 OU 3)?)')
  read(13,'(I2)') opcao
  write(13,'(35H NOME COMPLETO DO ARQUIVO DE DADOS:~)')
  read(13,'(A50)') fparam(1)
  open(UNIT=9,FILE=fparam(1),STATUS='UNKNOWN')
  write(13,'(35H NOME COMPLETO DO ARQUIVO DE SAIDA:~)')
  read(13,'(A50)') fparam(2)
  open(UNIT=10,FILE=fparam(2),STATUS='UNKNOWN')
  select case(opcao)
    case(1)
      read(9,*) (iparam(i),i=1,9)
      read(9,*) (rparam(i),i=1,4)
      read(9,*) (rparam(i),i=5,9)
      do i=1,iparam(1)
        read(9,*) par(i,1),par(i,2),par(i,3),sp(i)
      enddo
    case(2)
      write(13,'(34H DIGITE EM ORDEM (NC NT NV NP NM):~)')
      read(13,*) (iparam(i),i=1,5)
      write(13,'(30H INSTANTES DE AMOSTRAGEM (IA):~)')

```

```

read(13,*) iparam(6)
write(13,'(26H FUNÇÃO TRANSFÊNCIA (FTS):)')
write(13,'(29H (1) DATA-HOLD (ORDEM ZERO):)')
write(13,'(20H (2) SEM DATA-HOLD.)')
write(13,'(25H QUAL SUA OPÇÃO (1 OU 2)?)')
read(13,*) iparam(7)
write(13,'(25H ESTRATÉGIA DE CONTROLE :)')
write(13,'(23H (1) SELF-TUNING (PI):)')
write(13,'(27H (2) SELF-TUNING (DAHLIN):)')
write(13,'(18H (3) SELF-TUNING:;)')
write(13,'(22H (4) CONTROLADOR PAC.)')
write(13,'(29H QUAL SUA OPÇÃO (1,2,3 OU 4)?)')
read(13,*) iparam(8)
write(13,'(36H DIGITE EM ORDEM (TA DESV VLOW VUP):)')
read(13,*) (rparam(i),i=1,4)
write(13,'(36H DIGITE EM ORDEM (RMIN RMAX LOW UP):)')
read(13,*) (rparam(i),i=5,8)
if(iparam(8).eq.2) then
    write(13,'(22H DIGITE O VALOR DE mi:;)')
    read(13,*) rparam(9)
else
    rparam(9)=0.d0
end if
write(13,'(26H TRAJETORIA DO SET POINT :)')
write(13,'(16H (1) CONSTANTE:;)')
write(13,'(17H (2) ASCENDENTE:;)')
write(13,'(18H (3) DESCENDENTE:;)')
write(13,'(27H QUAL SUA OPÇÃO (1,2 OU 3)?)')
read(13,'(I2)') iparam(9)
write(9,'(9(I2,1X))') (iparam(i),i=1,9)
write(9,'(9(F7.3,1X))') (rparam(i),i=1,4)
write(9,'(9(F7.3,1X))') (rparam(i),i=5,9)
write(13,'(33H DIGITE EM ORDEM (KP TP TD YSP) :)')
do i=1,iparam(1)
    write(13,'(30H PARÂMETROS DO COMPONENTE NO.:,I2)') i
    read(13,*) par(i,1),par(i,2),par(i,3),sp(i)
    write(9,'(4(F7.3,1X))') par(i,1),par(i,2),par(i,3),sp(i)
end do
case(3)
    pause 'SAINDO DO PROGRAMA. TECLE ALGO.'
    call exit()
end select
close(9)
write(13,'(24H ARQUIVOS PARA O MÉTODO:;)')
write(13,'(26H NOME DO ARQUIVO DE DADOS:;)')
read(13,'(A50)') fparam(3)
write(13,'(26H NOME DO ARQUIVO DE SAIDA:;)')
read(13,'(A50)') fparam(4)
call CLEARSCREEN($GCLEARSCREEN)
write(13,'(A18)') 'DADOS FORNECIDOS:'

```

```

mi=rparam(9)
fts=iparam(7)
desv=rparam(2)
if(t.eq.0.d0) then
    yant=cp(ced,1)
    if(((cp(ced,1)-sp(ced)).gt.desv).and.(iparam(8).ne.4)) then
        cntrl=.false.
        R=0.d0
    else
        cntrl=.true.
        R=rparam(5)
    end if
    set=.false.
    icm=1
    return
else
    if(.not.cntrl) then
        if(((cp(ced,1)-sp(ced)).gt.desv).and.(iparam(8).ne.4)) then
            cntrl=.false.
            yant=cp(ced,1)
        else
            cntrl=.true.
        end if
        icm=1
        set=.false.
    end if
    if(cntrl) then
        if(na.eq.1) then
            R=rparam(5)
            spi=sp(ced)
        end if
    else
        R=0.d0
        return
    end if
end if
if(rfmax.and.(icm.gt.cm).and.rej) then
    if((cp(ced,1).lt.(sp(ced)-desv)).and.((ymed(icm-3)+ymed(icm-2)).gt. &
    & (ymed(icm-1)+ymed(icm))).and.(cp(ced,1).lt.yant)) then
        set=.false.
        rej=.true.
        pause 'ATENÇÃO!!! Vai para o rejeito.'
        call exit(2)
    else
        rej=.false.
    end if
end if
if(rej.or.(sp(ced).le.vmin)) then
    if((ced+1).lt.iparam(1)) then
        if((sp(ced).le.vmin).and.(sp(ced+1).le.vmin).and. &

```

```

        &      ((ced+2).lt.iparam(1))) then
            ced=ced+2
        else
            ced=ced+1
        end if
        pert=.false.
        rfmax=.false.
        rej=.false.
        yant=cp(ced,1)
        R=Rant-0.5d0
        na=1
        set=.true.

        ict=0
        icm=1
        tm=t+dbple(iparam(5)*Ta)
        ymed(icm)=vmin
    end if
end if
if(na.gt.iparam(6)) then
    cna=iparam(6)
    do i=1,(cna-1)
        u(i)=u(i+1)
        y(i)=y(i+1)
        erro(i)=erro(i+1)
    enddo
else
    cna=na
end if
select case(iparam(9))
    case(1)
        sp(ced)=sp(ced)
    case(2)
        sp(ced)=spi+dbple(na)/1.d3
    case(3)
        sp(ced)=spi-dble(na)/1.d3
end select
u(cna)=R
y(cna)=cp(ced,1)
erro(cna)=sp(ced)-cp(ced,1)
if((ced.ne.1).and.(cna.eq.1)) u(1)=vmin
if((cna.ge.3).and.(iparam(8).ne.4)) then
    call solve(par,nc,nv,ia,ced,na,cna,yss,uss,y,u,iparam,rparam,fparam)
end if
Kp=par(ced,1)
Tp=par(ced,2)
td=par(ced,3)+(Ta/2.d0)
nd=0
do while(.true.)
    if(td.ge.Ta) then

```

```

        td=td-Ta
        nd=nd+1
    else
        exit
    endif
enddo
if((R*rparam(8)).gt.Rant) pert=.false.
if((R*rparam(7)).lt.Rant) pert=.true.
Rant=R
if(set) then
    R=vmin
else
    if(cna.ge.3) then
        select case(iparam(8))
            case(1)
                Kc=(Tp/(Kp*td))*(0.9d0+(td/(Tp*12.d0)))
                Ti=td*((30.d0+3.d0*(td/Tp))/(9.d0+20.d0*(td/Tp)))
                fta=td/Tp
                amax=log((1.d0+exp(-Ta/Tp))/(exp(-Ta/Tp)* &
                & (3.d0+exp(-Ta/Tp))))
                if(fta.gt.amax) then
                    Kcu=(1.d0/(exp(-(Ta-td)/Tp)-exp(-Ta/Tp)))/Kp
                else
                    Kcu=(1.d0+exp(-Ta/Tp))/(1.d0+exp(-Ta/Tp)-2.d0* &
                    & exp(-(Ta-td)/Tp))/Kp
                end if
                if(Kc.gt.(Kcu/2.2d0)) Kc=Kcu/2.2d0
                R=Rant+Kc*((1.d0+(Ta/Ti))*erro(cna)-erro(cna-1))
                if(R.lt.rparam(5)) R=rparam(5)
            case(2)
                a1=exp(-Ta/Tp)
                c1=exp(-Ta/(60*mi))
                d1=1.d0-exp(-(dble(1+nd)*Ta-td)/(60*mi))
                d2=exp(-(dble(1+nd)*Ta-td)/(60*mi))-exp(-Ta/(60*mi))
                if(fts.eq.1) then
                    b1=Kp*(1.d0-exp(-(dble(1+nd)*Ta-td)/Tp))
                    b2=Kp*(exp(-(dble(1+nd)*Ta-td)/Tp)-exp(-Ta/Tp))
                    R=(d1/b1)*erro(cna)+((d2-a1*d1)/b1)* &
                    & erro(cna-1)-(a1*d2/b1)*erro(cna-2)+((c1*b1- &
                    & b2)/b1)*u(cna-1)+(c1*b2/b1)*u(cna-1- &
                    & nd)+((b1*d2+b2*d1)/b1)*u(cna-2-nd)+ &
                    & (b2*d2/b1)*u(cna-3-nd)
                else
                    b1=(Kp/Tp)*exp(-(dble(1+nd)*Ta-td)/Tp)
                    R=(d1/b1)*erro(cna)+((d2-a1*d1)/b1)* &
                    & erro(cna-1)-(a1*d2/b1)*erro(cna-2)+c1* &
                    & u(cna-1)+d1*u(cna-1-nd)+d2*u(cna-2-nd)
                end if
            case(3)
                a1=exp(-Ta/Tp)

```

```

        if(fts.eq.1) then
            b1=Kp*(1.d0-exp(-(db1e(1+nd)*Ta-td)/Tp))
            b2=Kp*(exp(-(db1e(1+nd)*Ta-td)/Tp)-exp(-Ta/Tp))
            bias=(yss-Kp*uss)*(1.d0-exp(-Ta/Tp))
            if(nd.eq.0) then
                R=(sp(ced)-a1*y(cna)-b2*u(cna-1)-bias)/b1
            else
                R=(sp(ced)-a1*sp(ced)-b2*u(cna-1)-bias)/b1
            end if
        else
            b1=(Kp/Tp)*exp(-(db1e(1+nd)*Ta-td)/Tp)
            bias=yss*(1.d0-a1)-(Kp*uss/Tp)* &
            & exp(-(db1e(1+nd)*Ta-td)/Tp)
            if(nd.eq.0) then
                R=(sp(ced)-a1*y(cna)-bias)/b1
            else
                R=(sp(ced)-a1*sp(ced)-bias)/b1
            end if
        end if
    case(4)
        R=0.435683d0*(cp(ced,2)**(-1.21302d0))
        Kc=22.4905d0*(cp(ced,2)**(-1.90183d0))
        Ti=0.0998669d0-(0.0166394d0*cp(ced,2))
        R=R+Kc*((1.d0+(Ta/Ti))*erro(cna)-erro(cna-1))
    end select
else
    if(pert) then
        R=rparam(8)*Rant
    else
        R=rparam(7)*Rant
    end if
end if
end if
if(dabs(R-Rant).gt.5.d0) then
    if(R.gt.Rant) then
        if(Rant.ge.Ta) then
            R=Rant+1.d0
        else
            R=Rant+5.d0
        end if
    else
        R=Rant-5.0d0
    end if
else
    if((dabs(R-Rant).gt.1.d0).and.(Rant.ge.Ta).and.(R.gt.Rant)) then
        R=Rant+1.d0
    end if
end if
if(R.gt.rparam(6)) R=rparam(6)
if(R.lt.rparam(5)) R=rparam(5)

```

```

ymed(icm)=ymed(icm)+cp(ced,1)
ict=ict+1
if(t.ge.tm) then
    ymed(icm)=ymed(icm)/ict
    ict=0
    if(icm.gt.cm) then
        icm=cm
        do i=1,(cm-1)
            ymed(i)=ymed(i+1)
        enddo
    else
        icm=icm+1
    end if
    tm=t+dbple(iparam(5)*Ta)
end if
yant=cp(ced,1)
na=na+1
return
end subroutine

```

Função que chama o método SQP da NAG para a identificação do processo.

```

subroutine solve(par,nc,nv,ia,ced,na,cna,yss,uss,y,u,iparam,rparam,fparam)
    implicit none
    character(50) fparam(5)
    integer      nc,nv,ia,ced,na,cna,iparam(9)
    real(8)      par(nc,nv),uss,yss,y(ia),u(ia),rparam(9)
    integer      cia,iam,nin,nout,ioptns,lda,ldcj,ldfj,ldr,maxn,nbnd, &
    &            liwork,lwork,ifail,inform,iter,i,j,n,m,nclin,ncnln
    parameter    (maxn=5,iam=20,ldfj=iam-2,lda=1,ldcj=1,nclin=1,ncnln=1, &
    &            ldr=maxn,lwork=20*maxn+ldfj*(maxn+3),nbnd=maxn+ &
    &            nclin+ncnln,liwork=3*maxn+nclin+2*ncnln+ldfj*(maxn+3))
    integer      istate(nbnd),iwork(liwork),iuser(iam)
    real(8)      Ta,sumsq,ubnd,lbnd,bl(maxn),bu(maxn),c(ldcj),f(ldfj), &
    &            cjac(ldcj,maxn),r(ldr,maxn),clamda(nbnd),work(lwork), &
    &            x(maxn),user(2*iam+6),fjac(ldfj,maxn),a(lda,maxn)
    parameter    (nin=1,nout=2,ioptns=nin,ubnd=1.d3,lbnd=1.d-5)
    external     e04udm,e04upf,e04uqf,e04urf,x04abf,objfun,confun
    open(UNIT=nin,FILE=fparam(3),STATUS='UNKNOWN')
    open(UNIT=nout,ACCESS='APPEND',FILE=fparam(4),STATUS='UNKNOWN')
    write(nout,*)
    write(nout,*) '##### &
    & #####'
    write(nout,'(1X,A,I3)') 'Solve results in NA = ',na
    read(nin,*)
    read(nin,*)
    call x04abf(1,nout)
    m=cna-2
    n=iparam(3)
    cia=iparam(6)

```

```

Ta=rparam(1)
uss=0.d0
yss=0.d0
if(cia.lt.10) then
    do i=1,int(cia/2)
        uss=uss+u(i)/dbple(int(cia/2))
        yss=yss+y(i)/dbple(int(cia/2))
    enddo
else
    do i=1,5
        uss=uss+u(i)/5.d0
        yss=yss+y(i)/5.d0
    enddo
end if
do i=1,n
    x(i)=par(ced,i)
    bu(i)=rparam(4)*x(i)
    bl(i)=rparam(3)*x(i)
    if(i.ne.3) then
        if(bu(i).gt.ubnd) bu(i)=ubnd
        if(bl(i).lt.lbnd) bl(i)=lbnd
    else
        if(bu(i).gt.(dbple(cna-2)*Ta)) bu(i)=dbple(cna-2)*Ta
        if(bl(i).lt.lbnd) bl(i)=lbnd
    end if
enddo
bl(n+nclin)=Ta
bu(n+nclin)=1.d2*Ta
bl(n+nclin+ncnln)=0.5d0*(x(1)**2.d0+x(2)**2.d0+x(3)**2.d0)
bu(n+nclin+ncnln)=2.0d0*(x(1)**2.d0+x(2)**2.d0+x(3)**2.d0)
a(1,1)=0.d0
a(1,2)=1.d0
a(1,3)=1.d0
iuser(1)=iparam(7)
iuser(2)=iparam(6)
user(1:iam)=y(1:iam)
user(iam+1:2*iam)=u(1:iam)
user(2*iam+1)=yss
user(2*iam+2)=uss
user(2*iam+3)=Ta
user(2*iam+4)=x(1)
user(2*iam+5)=x(2)
user(2*iam+6)=x(3)
call e04uqf(ioptns,inform)
if(inform.ne.0) then
    write(nout,*)
    write(nout,'(1X,A,I3)') 'E04UQF terminated with INFORM=',inform
    return
end if
close(nin)

```

```

        ifail=-1
        call e04upf(m,n,nclin,ncnln,lda,ldcj,ldfj,ldr,a,bl,bu,confun,objfun,iter,istate, &
        &          c,cjac,f,fjac,clamda,sumsq,r,x,iwork,liwork,work,lwork,iuser,user,ifail)
        do j=1,n
            par(ced,j)=x(j)
        enddo
        close(nout)
        return
end subroutine

```

Função objetivo para o método E04UPF da NAG.

```

subroutine objfun(mode,m,n,ldfj,x,f,fjac,nstate,iuser,user)
    implicit none
    integer      ldfj,m,mode,n,nstate,iuser(*)
    real(8)      f(ldfj),fjac(ldfj,n),x(n),user(*)
    integer      i,k,fts,nd,cia,iam
    parameter     (iam=20)
    real(8)      Ta,Kp,Tp,td,exp1,exp2,yss,uss,u(iam),y(iam),a1,b1,b2,bias
    fts=iuser(1)
    cia=iuser(2)
    yss=user(2*iam+1)
    uss=user(2*iam+2)
    Ta=user(2*iam+3)
    u=user(1:iam)
    y=user(iam+1:2*iam)
    nd=0
    td=x(3)
    do while(.true.)
        if(td.ge.Ta) then
            td=td-Ta
            nd=nd+1
        else
            if(m.eq.1) nd=0
            exit
        endif
    enddo
    Kp=x(1)
    Tp=x(2)
    td=x(3)
    if(dabs(Tp).lt.1.d-5) then
        exp1=0.0d0
        exp2=0.0d0
    else
        exp1=exp(-Ta/Tp)
        exp2=exp(-(dble(1+nd)*Ta-td)/Tp)
    end if
    a1=exp1
    k=3+nd
    select case(fts)

```

```

case(1)
  b1=Kp*(1.d0-exp2)
  b2=Kp*(exp2-exp1)
  bias=(yss-Kp*uss)*(1.d0-exp1)
  do i=1,m
    f(i)=y(k)-a1*y(k-1)-b1*u(k-1-nd)-b2*u(k-2-nd)-bias
    fjac(i,1)=-(1.d0-exp2)*u(k-1-nd)-(exp2- &
      & exp1)*u(k-2-nd)+uss*(1.d0-exp1)
    fjac(i,2)=(Kp*(dble(1+nd)*Ta-td)*exp2* &
      & (u(k-1-nd)-u(k-2-nd))-Ta*exp1* &
      & (y(k-1)-yss-Kp*(u(k-2-nd)-uss)))/Tp**2.d0
    fjac(i,3)=Kp*exp2*(u(k-1-nd)-u(k-2-nd))/Tp
    if(k.ge.cia) then
      k=cia
    else
      k=k+1
    end if
  enddo
case(2)
  b1=(Kp/Tp)*exp2
  bias=yss*(1.d0-exp1)-(Kp/Tp)*uss*exp2
  do i=1,m
    f(i)=y(k)-a1*y(k-1)-b1*u(k-1-nd)-bias
    fjac(i,1)=-((u(k-1-nd)-uss)*exp2)/Tp
    fjac(i,2)=(Kp/Tp**2.d0)*(exp2-((dble(1+nd)* &
      & Ta-td)/Tp)*exp2)*(u(k-1-nd)-uss)- &
      & (Ta/Tp**2.d0)*exp1*(y(k-1)-yss)
    fjac(i,3)=-Kp*(u(k-1-nd)-uss)*exp2/Tp**2.d0
    if(k.ge.cia) then
      k=cia
    else
      k=k+1
    end if
  enddo
end select
return
end subroutine

```

Sub-rotina das restições não-lineares.

```

subroutine confun(mode,ncnln,n,ldcj,needc,x,c,cjac,nstate,iuser,user)
  implicit none
  integer      mode,ncnln,n,ldcj,nstate,iam,needc(ncnln),iuser(*)
  parameter    (iam=20)
  real(8)      x(n),c(*),cjac(ldcj,*),user(*)
  if(needc(1).gt.0) then
    c(1)=x(1)**2.d0+x(2)**2.d0+x(3)**2.d0
    cjac(1,1)=2.d0*x(1)
    cjac(1,2)=2.d0*x(2)
    cjac(1,3)=2.d0*x(3)
  end if

```

```

        end if
        return
    end subroutine

```

Protótipos (ou interfaces) para as funções em C da placa ADA.

```

interface
    subroutine write_dig(dado)
        !ms$attributes C,ALIAS:'_write_dig' :: write_dig
        integer(1) dado
    end subroutine
end interface
!
interface
    integer(4) function read_anl(canal,offset)
        !ms$attributes C,ALIAS:'_read_anl' :: read_anl
        character(1) canal,offset
    end function
end interface

```

Funções da Placa conversora Analógico-Digital-Analógico.

```

#include "conio.h"
#define ADLSB 0          /* Porta de leitura do lsb do conv ad */
#define ADMSB 1          /* Porta de leitura do msb do conv ad */
#define ADOFF 2          /* Porta para ajuste de offset */
#define ADSTS 4          /* Porta de controle do modo de operacao */
#define DAMSB 6          /* Porta de escrita do msb do conversor da */
#define DALSB 7          /* Porta de escrita do lsb do conversor da */
#define CTL 8            /* Porta de controle do mux de ent/saida e dos sh */
#define base 0x220
#define IODIG 10         /* Entrada e saida digital */
#define TIMER0 12        /* Timer 0 do 8253 */
#define TIMER1 13        /* Timer 1 do 8253 */
#define TIMER2 14        /* Timer 2 do 8253 */
#define TIMCTL 15        /* Porta de controle do 8253 */
/* Definicao dos bits de importancia */
#define BSHEAN 0x10      /* Bit de controle do sample-hold entanl (1=sample) */
#define BSHSA 0x08       /* Bit de controle do sample-holds das saidas(1=sample)*/
#define MASC0_5 0x00     /* Mascara no modo de operacao 0-5V */
#define MASC1_5 0x29     /* Mascara no modo de operacao 1-5V */
#define MASC0_4 0x08     /* Mascara no modo de operacao 0-4V */
#define MASCBIP 0x23     /* Mascara no modo de operacao BIPOLAR */
#define TIME_OUT 25      /* Tempo de espera do fim da conversao A/D */
extern char erro;        /*Codigo de erro */
unsigned char modoper;    /* Byte que contem o modo de operacao da placa */
unsigned int read_anl(),adj_offset();
unsigned char offset;
void selchda();

```

```

void write_dig();
void write_anl();
/*-----*/
/* INICIO DAS ROTINAS DE TRATAMENTO DA AD/DA */
/*-----*/
/* SELECAO DO CANAL DO MUX DE ENTRADA E DA SAMPLE NA ENTRADA */
/*-----*/
selchad(canal)
unsigned char canal;
{
    unsigned char chad;
    chad = canal << 5;          /* Posiciona o end do mux (badchan ) */
    chad &= 0xe0;              /* Isola somente badchan0-2 */
    _outp(base + CTL,chad);    /* Selecciona o canal */
    modoper |= BSHEAN;         /* Introduz bit de sample */
    _outp(base + ADSTS,modoper); /* Sample-hold da inanl em sample */
    modoper &= ~BSHEAN;        /* Retira bit de sample */
    _outp(base + ADSTS,modoper); /* Sample-hold da inanl em hold */
    return(canal);
}
/*-----*/
/* SELECIONA O CANAL DO MUX DE SAIDA E DA SAMPLE NA SAIDA */
/*-----*/
void selchda(canal)
unsigned char canal;
{
    unsigned char chda;
    canal &= 0x07;              /* Isola */
    chda = (canal << 5);        /* Posiciona o end do mux (badchan0-2 ) */
    chda |= canal;              /* Soma c/ o mux do sample-hold (bdachan0-2) */
                                /* Selecciona o canal+sample-hold corresp */
    chda |= BSHSA;              /* Introduz o bit de sample */
    _outp(base + CTL,chda);     /* Coloca o sample-hold do canal em sample */
}
/*-----*/
/* ROTINA DE ESPERA DO FIM DA CONVERSAO */
/*-----*/
wait_eoc() {
    unsigned register int ciclos,status;
    for (ciclos = TIME_OUT;ciclos;ciclos --) { }
    return 0;
}
/*-----*/
/* ROTINA DE LEITURA DO CONVERSOR */
/* Parametro de entrada: numero do canal
   Parametro de saida: retorna o valor da conversao
                        se ocorrer erro retorna -1 */
/*-----*/
wait_eoc(void);
selchad(unsigned char canal);

```

```

unsigned int read_anl(canal,offset)
unsigned char canal,offset;
{
    unsigned int dado;
    unsigned int dadols,dadoms;
    selchad(canal);
    _outp(base+ADOFF,offset);    /* Normaliza o valor do offset */
    dadols = _inp(base+ADLSB);    /* Envia o start ao ad */
    wait_eoc();                  /* Delay para conversao */
    dadoms = _inp(base+ADMSB);    /* le os 4 bit`s mais signific. */
    dadols = _inp(base+ADLSB);    /* le os 8 bit's menos signific. */
    dado = (dadoms << 8) + dadols;
    return (dado);
}
/*-----*/
/* ESCRIVE O VALOR DE 'dado' NA SAIDA ANALOGICA DE 'canal' */
/*-----*/
void selchda(unsigned char canal);
void write_anl(dado,canal)
unsigned int dado;
unsigned char canal;
{
    unsigned char dadols,dadoms;
    dadols = dado;                /* Inicia o deslocamento do dado */
    dadoms = dado >> 8;           /* Desloca os dois bits mais signific. */
    _outp(base + DALSB,dadols);    /* Escreve byte menos significativo */
    _outp(base + DAMSB,dadoms);    /* Escreve byte mais significativo */
    selchda(canal);               /* Transf para o canal de saida desejado */
    return;
}
/*-----*/
/* ESCRIVE NAS ENTRADAS DIGITAIS */
/*-----*/
void write_dig(dado)
char dado;
{
    _outp(base + IODIG,dado);
}
/*-----*/
/* ALGORITMO PARA AJUSTE AUTOMATICO DE OFFSET */
/* Retorna os seguintes valores:
    100h -> Se nao existe o sinal de referencia de 3.500V na
           entrada analogica 7
    200h -> Se em 4 tentativas de ajustar o offset isto nao
           for conseguido
    Num -> De 0h a 0FFh que e' o valor para zerar o offset */
/*-----*/
unsigned int read_anl(unsigned char canal,unsigned char offset);
unsigned int adj_offset()
{

```

```

unsigned int in1,in2,i,inatual,trigger = 0x800;
unsigned char flag = 0,delay;
if(modoper & 0x01) /* modo 1 a 5 V ? */
    trigger = 0xa00; /* sim, armazene a00h como valor de comparacao */
else
    trigger = 0xb33;
inatual = read_anl(7,140); /* le a ent de refer. centrando o offset */
if(inatual < (trigger - 0x100)) /* existe a referencia de (3500mV)? */
    return(0x100); /* nao,retorne uma condicao de erro */
for(i=0;flag==0;++i) /* inicio do integ. p/ busca do pto otimo */
{
    inatual = read_anl(7,(i&0xff)); /* tente com i valor de offset */
    for(delay=0;delay<50;++delay) /* rotina para atraso */
        if(inatual == trigger) /* erro de leitura = zero? */
        {
            flag = 1; /* termine a execucao, ponto encontrado */
            if(i == 0x400) /* feita o scan 4 vezes sem sucesso? */
                return(0x200); /* termine e retorne condicao de erro */
        }
    return (i-1); /* termine e retorne o valor ajustado */
}
/*-----*/
/* ROTINA PARA SELECAO AUTOMATICA DO MODO DE OPERACAO
ACEITA OS SEGUINTEs PARAMETROS: */
/* n = 0 -> 0-5V (modo default)
n = 1 -> 1-5V
n = 2 -> 0-4V
n = 3 -> bipolar

OBS : modo de leitura do conversor : pooling */
/*-----*/
modo_oper(char n) {
    unsigned char masc;
    masc=MASC0_5; /* default operacao de 0-5V */
    switch(n) /* scan do modo de operacao alternativo */
    {
        case 1:
            masc=MASC1_5;
            break;
        case 2:
            masc=MASC0_4;
            break;
        case 3:
            masc=MASCBIP;
            break;
    }
    modoper = masc; /* armazena o modo de operacao setado */
    _outp(base+ADSTS,masc); /* envia a placa */
    return 0;
}
/*-----*/

```

REFERÊNCIA BIBLIOGRÁFICA

- ASTRÖM, K. J. Theory and Applications of Adaptive Control - A survey. *Automatica*, v. 19, n. 5, p. 471-486, 1983.
- BONNY, L. Strategies for handling mixtures in multicomponent batch distillations with slop recycle. *Chem. Engng. Processing*, v. 34, p. 401-414, 1995.
- BOSLEY, J. R.; EDGAR, T. F. An efficient dynamic model for batch distillation. *J. Proc. Cont.*, v. 4, n. 4, p. 195-204, 1994.
- BRITT, H. I., ROSEN, J. B. *Can. Chem. News*, v. 41, n. 8, p. 22, 1989.
- CONVERSE, A. O., GROSS, G. D. Optimal distillate-rate policy in batch distillation. *Ind. Engng. Chem. Fundam.*, v. 2, p. 217, 1963.
- COWARD, I. The time optimal problem in binary batch distillation. *Chem. Engng. Sci.*, v. 22, p. 503, 1967.
- CUNHA, A. P. *Automação e controle "on-line" de uma coluna de destilação em batelada*. Campinas: Faculdade de Engenharia Química, Universidade Estadual de Campinas, 1996, 115 p. Tese (Mestrado).
- CYBENKO, G. Approximation by superpositions of sigmoidal function. *Math. Control Signal Systems*, v.2, p. 303-314, 1989.
- DAVIDYAN, A. G., KIVA, V. N., MESKI, G. A., MORARI, M. Batch distillation in a column with a middle vessel. *Chem. Engng. Sci.*, v. 49, n. 18, p. 3033-3051, 1994.
- DIWEKAR, U. M. An efficient design method for binary, azeotropic, batch distillation columns, *AIChE Journal*, v. 37, n. 10, p. 1571-1578, 1991.
- DIWEKAR, U. M.; MALIK, R. K.; MADHAVAN, K. P. Optimal reflux rate policy determination for multicomponent batch distillation columns. *Comput. Chem. Engng.*, v. 11, n. 6, p. 629-637, 1987.
- FARHAT, S.; DOMENECH, S.; PIBOULEAU, L.; CZERNICKI, M. Optimal control of batch distillation via nonlinear programming. *Chem. Eng. Process.*, v. 29, p. 33-38, 1991.

- FILETI, A. M. F. *Controle Adaptativo e Preditivo com Redes Neurais de uma coluna de destilação em batelada*. Campinas: Faculdade de Engenharia Química, Universidade Estadual de Campinas, 1996, 201 p. Tese (Doutorado).
- FILETI, A. M. F. *Estratégias de controle em destilação batelada*. Campinas: Faculdade de Engenharia Química, Universidade Estadual de Campinas, 1992, 115 p. Tese (Mestrado).
- FLETCHER, R. *Practical methods of optimization*. New York: John Wiley & Sons, 1987, 436 p.
- HAGAN, M. T., MENHAJ, M. B. Training Feedforward Networks with the Marquardt Algorithm. *IEEE Transactions on Neural Networks*, v. 5, n. 6, p. 989-993, november 1994.
- HASEBE, S., KUROOKA, R., HASHIMOTO, I. Comparison of the separation performances of a multi-effect batch distillation system and a continuous distillation system. *Proceedings DYCORD+ '95*, p. 249-254, Denmark, 1995.
- JANG, M. J. Avail. Univ. Microfilms. Int n. DA9026755, Diss. Abstr. Int. B, v. 51, n. 5, p. 2491, 1990.
- KALMAN, R. E. Design of self-optimizing control system. *Trans. ASME*, v. 80, p. 468, 1958.
- KERKHOF, L. H. J.; VISSERS, H. J. M. On the profit of optimum control in batch distillation, *Chem. Engng. Sci.*, v. 33, p. 961-970, 1977.
- LUYBEN, W. L. Multicomponent batch distillation – 1. Ternary systems with slop recycle. *Ind. Eng. Chem. Res.*, v. 27, p. 642-647, 1988.
- LUYBEN, W. L. Some practical aspects of optimal batch distillation design. *Ind. Eng. Chem. Proc. Des. Dev.*, v. 10, n. 1, p. 54-59, 1971.
- MARQUARDT, D. An algorithm for least squares estimation of non-linear parameters. *J. Soc. Ind. Appl. Math.*, p. 431-441, 1963
- MORRIS, A. J., MONTAGUE, G. A., THAM, M. T., WILLIS, M. J., DI MASSIMO, C. Artificial neural networks. *Trans. IChemE*, v. 72, Part A, 1994.

- MORRIS, A. J., MONTAGUE, G. A., THAM, M. T., DI MASSIMO, C., WILLIS, M. J., On neural networks in chemical process control. *Proc. IEE*, v. 138, Part D, p. 256-266, 1991.
- MUJTABA, I. M., MACCHIETTO, S. Optimal operation of multicomponent batch distillation – Multiperiod formulation and solution. *Computers Chem. Engng.*, v. 17, n. 12, p. 1191-1207, 1993.
- MURTY, B. S. N., GANGIAH, K., HUSAIN, A. Performance of various methods in computing optimal control policies. *Chem. Engng. J.*, v. 19, p. 201, 1980.
- NILSSON, N. J. *Learning Machines*. New York: McGraw-Hill, 1965.
- OFFERS, H., DUSSEL, R., STICHLMAIR, J. Minimum energy requirement of distillation processes, *Computers Chem. Engng.*, v. 19, Suppl., p. S247-S252, 1995.
- PRICE, P. C. *Chem. Engng. Symp. Ser.*, v. 23, p. 96, 1967.
- PONTON, J. W., KLEMES, J. Alternatives to neural networks for inferential measurement. *Computers Chem. Engng.*, v. 17, n. 10, p. 991-1000, 1993.
- QUANTRILLE, T. E., LIU, Y. A. *Artificial intelligence in chemical engineering*. Academic Press, Inc., 1991.
- QUINTERO-MARMOL, E., LUYBEN, W. L. Inferential model-based control of multicomponent batch distillation. *Chem. Eng. Sci.*, v. 47, n. 4, p. 887-898, 1992.
- ROBINSON, E. R. The optional control of an industrial batch distillation column. *Chem. Engng. Sci.*, v. 25, p. 921, 1970.
- ROBINSON, E. R. The optimization of batch distillation operations. *Chem. Engng. Sci.*, v. 24, p. 1661, 1969.
- SHINSKEY, F. G. *Process Control Systems – Application, Design and Tuning*. New York: McGraw-Hill, 3rd. edition, 1979, 485 p.
- SKOGESTAD, S., WITTGENS, B., SORENSEN, E., LITTO, R. Multivessel batch distillation, *AIChE J.*, v. 43, n. 4, p. 971-978, 1997.

- SORENSEN, E. Alternative ways of operating a batch distillation column, ESCAPE, 1998.
- SORENSEN, E., SKOGESTAD, S. Comparison of regular and inverted batch distillation. *Chem. Eng. Sci.*, v. 51, n. 22, p. 4949-4962, 1996.
- STEPHANOPOULOS, G. *Chemical Process Control – An Introduction to theory and practice*. Pretence-Hall, Inc., 1984.
- THAM, M. T., MORRIS, A. J., MONTAGUE, G. A., LANT, P. A. Soft sensors for process estimation and inferential control, *J. Proc. Control*. v. 1, p. 3-14, 1991
- WILLIS, M. J., MONTAGUE, G. A., DI MASSIMO, C., THAM, M. T., MORRIS, A. J. Artificial neural networks in process estimation and control. *Automatica*, v. 28, n. 6, p. 1181-1187, 1992.
- WILLIS, M. J., DI MASSIMO, C., MONTAGUE, G. A., THAM, M. T., MORRIS, A. J. Artificial neural networks in process engineering. *Proc. IEE*, Part D, 138(3), p. 256-266, 1991.
- WITTENMARK, B. A self-tuning regulator. Thesis, TFRT-1003, Department of Automatic Control, Lund Institute of Tecnology, 1973.
- YU, G., YANG, Z., BAI, P., MO, Z. Huagong Xuebao, v. 40, n. 3, p. 271, 1989.
- YU, C. C.; LUYBEN, W. Control of Multicomponent distillation columns using rigorous composition estimators, *Chem. Enging. Symp. Series*, v. 104, p. A29-A69, 1987.
- ZHANG, J., MARTIN, E. B., MORRIS, A. J., KIPARISSIDES, C. Inferential estimation of polymer quality using stacked neural networks. *Computers Chem. Engng.*, v. 21, Suppl., p. S1025-S-1030, 1997.

ABSTRACT

Batch distillation is one of the most important separation processes used in many chemical industries, specially those related to the manufacture of fine and specialty chemicals. Due to the flexibility in purifying different mixtures under a variety of operational conditions, batch distillation is a process of valuable importance. The flexibility of the equipment coupled to the dynamic nature and the nonlinear process behaviour make the conventional control techniques inefficient and pose challenging control system design. From the above and looking for the constant overhead product purity operation, it is emphasized the importance of the development and experimental testing of suitable control systems for the automatic operation of such equipment.

The present work describes the development and experimental testing of a direct digital control system for a pilot batch distillation column. Conventional feedback control strategies were firstly implemented in a 486 DX 33MHz PC coupled to the column. It was then proved the inefficient results of conventional schemes for this transient process control.

Since the process under study claimed to more elaborated control systems, adaptive control algorithms were implemented. A Self Tuning Regulator (STR) based on a regressive analysis of the process was developed. An input/output discrete model was used to represent the process, which was approached to a first order plus time delay system. The on line process identification proceeds using the Successive Quadratic Programming method applied to Least Squares. The computational code was available in NAG Mathematical Library. A digital PI control law and the law obtained through Dahlin's Algorithm were individually tested. The performance of the developed STR was compared to a previously implemented Programmable Adaptive Controller (PAC). The results show that the STR is a suitable and an easy control strategy to be implemented when compared to the PAC, which requires strong off-line computational effort and is highly dependent on the knowledge of the feed mixture composition and properties.

Product and bottom composition were estimated from vapor temperature measurements and the equilibrium properties of the n-hexane/n-heptane binary system used. To allow experimental tests with a ternary system (n-hexane/ciclohexane/n-heptane), a composition soft-sensing system based on Neural Networks was developed.

The developed digital control system made the conventional batch distillation more efficient and easy to operate. Experiments performed on the pilot column confirm previous simulation results.