

UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA QUÍMICA

ÁREA DE CONCENTRAÇÃO

DESENVOLVIMENTO DE PROCESSOS QUÍMICOS

***AVALIAÇÃO E DESENVOLVIMENTO DE ALGORITMOS DE
CONTROLE APLICADO A UM PROCESSO EXTRATIVO DE
FERMENTAÇÃO ALCÓOLICA CONTÍNUA***

Autora: Elis Regina Duarte

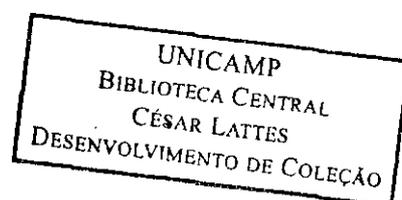
Orientador: Prof. Dr. Rubens Maciel Filho

Co-Orientador: Prof. Dr. Laércio Ender

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

Campinas - SP

Agosto de 2007



UNIDADE	<u>BC</u>
Nº CHAMADA:	<u>T/UNICAMP</u>
	<u>D85a</u>
V	Ex.
TOMBO DOCL	<u>74537</u>
PROC	<u>16.145-07</u>
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	<u>11,00</u>
DATA	<u>10/10/07</u>
BIB-ID	<u>246113</u>

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

D85a Duarte, Elis Regina
Avaliação e desenvolvimento de algoritmos de controle aplicado a um processo extrativo de fermentação alcoólica contínua. / Elis Regina Duarte.-- Campinas, SP: [s.n.], 2007.

Orientadores: Rubens Maciel Filho, Laércio Ender
Tese (Doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Química.

1. Álcool. 2. Controle preditivo. 3. Redes neurais (Computação). 4. Aprendizado do computador. I. Maciel Filho, Rubens. II. Ender, Laércio. III. Universidade Estadual de Campinas. Faculdade de Engenharia Química. IV. Título.

Título em Inglês: Development and evaluating the performance of predictive and adaptive controllers applied to an extractive fermentative process

Palavras-chave em Inglês: Ethanol, predictive control, adaptive control, Artificial neural network, On-line learning

Área de concentração: Desenvolvimento de processos Químicos

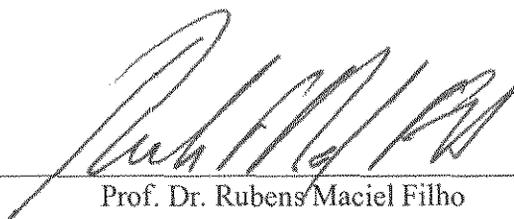
Titulação: Doutora em Engenharia Química

Banca examinadora: Daniel Ibraim Pires Atala, Roberto de Campos Giordano, Delba Nisi Cosme Melo e Caliane Bastos Costa

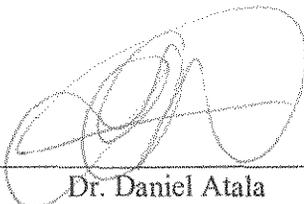
Data da defesa: 14/08/2007

Programa de Pós-Graduação: Engenharia Química

Tese de Doutorado defendida por Elis Regina Duarte e aprovada em 14 de agosto de 2007 pela banca examinadora constituída pelos doutores:



Prof. Dr. Rubens Maciel Filho
(Orientador)



Dr. Daniel Atala
Centro de Tecnologia Canavieira – CTC (Titular)



Prof. Dr. Roberto de Campos Giordano
Universidade Federal de São Carlos - UFSCar (Titular)



Dra. Cálhane Bastos Borba Costa
Pós-Doutoranda FAPESP (Titular)



Dra. Delba Nisi Cosme Melo
Pós-Doutoranda FAPESP (Titular)

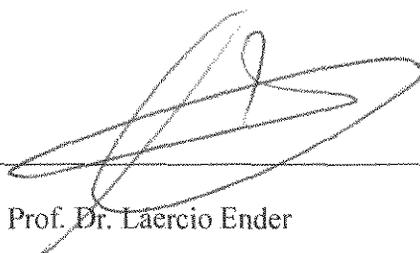
00749584

Este exemplar corresponde à versão final da Tese de Doutorado em Engenharia Química



Prof. Dr. Rubens Maciel Filho

Orientador



Prof. Dr. Laercio Ender

Co – Orientador

UNICAMP
BIBLIOTECA CENTRAL
CÉSAR LATTES
DESENVOLVIMENTO DE COLEÇÃO

Dedico esta Tese ao meu marido e grande
amor Raphael Henrique Korb Neotti.

AGRADECIMENTOS

Agradeço à Deus e todas forças angelicais por sua eterna bondade e demonstração de carinho e luz ao longo de toda jornada.

Agradeço aos meus pais Alicio Duarte e Catarina Duarte que não desistiram perante tantas dificuldades, especialmente por não desistirem de minha vida, quando nasci com problemas cardíacos, por estarem presentes e sendo minha fortaleza quando mais precisei. Obrigada por serem os melhores pais do mundo, me ensinar a acreditar e por me fazerem ver o lado bom da vida sempre.

Agradeço ao meu irmão Edielson George Duarte por todo carinho e incentivo.

Agradeço a minha sobrinha Camila Eduarda por me mostrar a simplicidade das coisas e o fascínio por aprender.

Agradeço ao meu marido Raphael Henrique Korb Neotti por toda paciência, carinho e dedicação, por me ensinar o verdadeiro significado da palavra amor, suportar meses de distância e todas as fases de *stress*. Fazer os momentos de minha vida mais coloridos e apaixonantes ao seu lado, dar-me incentivo e também por sua ajuda na parte computacional, que foi de fundamental importância para o desenvolvimento deste trabalho.

Agradeço ao meu orientador Rubens Maciel Filho por toda dedicação, incentivo e confiança.

Agradeço ao Daniel I. P. Atala por toda paciência e disponibilidade em me auxiliar no entendimento do processo fermentativo.

Agradeço pelos amigos campineiros, vindos de todos os lugares, que tornaram minha vida mais alegre, especialmente: a Sarita Rabelo por sua alegria contagiante e “causos” mineiros divertidíssimos, a Wanderléia Scarmen por me ajudar com seu testemunho a encontrar meu caminho espiritual, a Elisiane Wust Blumenauense que conheci em Campinas e hoje é uma grande amiga.

A minha família campineira Aline F. C. Passini e Elisiane Wust por todo carinho e companheirismo.

A família que mora em meu coração Maria das Graças, Márcio e o nosso anjinho Rafael por toda motivação e incentivo.

A todos meus antigos e novos companheiros de trabalho por toda amizade e ajuda, em especial aos colegas da UNICAMP, LOPCA, FURB, LFC e LCP.

Os meus mais sinceros agradecimentos a todos aqueles que contribuíram de uma forma ou de outra para que a realização deste trabalho fosse possível.

A CAPES pelo apoio financeiro.

“No princípio, ela o acompanhará por caminhos tortuosos, trazendo-lhe medo e temor; causar-lhe-á incomodo com a sua disciplina até poder fiar-se nele, e o experimentará com seus preceitos. A seguir, retornará diretamente para ele, alegrando-o e revelando-lhe seus segredos” (Ecl. 4, 17-18).

RESUMO

O objetivo deste trabalho foi desenvolver e avaliar diferentes algoritmos de controle para o processo extrativo de fermentação alcoólica contínua. Para isto foram comparados controladores do tipo preditivo e adaptativo.

Para o controle preditivo, foi avaliado o Controle por Matriz Dinâmica (DMC) e foi desenvolvido um algoritmo de controle preditivo baseado em modelo usando redes neurais artificiais (MPC Neural) com aprendizagem em tempo real das redes.

Para o controle adaptativo, foi proposto o aperfeiçoamento do algoritmo de controle CONDEG (Controle Neural Direto Baseado no Erro Global) Modificado, desenvolvido por Duarte (2004). O algoritmo está baseado em redes neurais artificiais, com aprendizagem em tempo real, de acordo com as alterações que ocorrem no processo. Os parâmetros de penalização das ações de controle, que são parâmetros de projeto do controlador, foram ajustados ao longo do tempo através da aplicação de um algoritmo do Filtro de Kalman.

Para o procedimento de investigação foi utilizada a simulação computacional para o qual todos os algoritmos de controle estudados foram implementados em linguagem de programação Fortran 90 e aplicados a um processo extrativo de fermentação alcoólica contínua para produção de etanol desenvolvido por Silva (1999). O modelo matemático utilizado foi desenvolvido por Costa *et al.* (2001).

As simulações em malha fechada realizadas utilizando os algoritmos propostos mostraram melhores resultados para os algoritmos de controle usando redes com aprendizagem ao longo do tempo e que o algoritmo de controle CONDEG Modificado usando filtro de Kalman com fator de velocidade associado foi eficiente e robusto, pois apresentou bons resultados em problemas dos tipos servos e regulador.

Palavras-Chaves: Etanol, Processo Extrativo, Controle Preditivo, Controle Adaptativo, Redes Neurais Artificiais, Aprendizagem “On-line”.

ABSTRACT

The objective of the present work is to develop and to evaluate the performance of predictive and adaptive controllers, applied to an extractive fermentative process.

As predictive controllers the Dynamical Matrix Control (DMC) and a model predictive control based on artificial neural networks with on-line learning were considered.

The adaptive controller is an improvement of the Modified Condeg strategy control (Direct Neural Control based on Global Error), developed by Duarte (2004). The strategy is based on artificial neural networks, with on-line learning, according to modifications that occur in the process. The control actions penalization parameters, that are in fact controller design parameters, are on-line adjusted through an algorithm based on Kalman filter.

The performance evaluation was carried out through computer simulation with all algorithms implemented in Fortran 90. As a case study, an extractive fermentation alcoholic process developed by Silva (1999) was taken into account with the mathematical model developed by Costa *et. al* (2001).

The results obtained from closed-loop simulations using the proposed algorithms showed better results for the neural networks with on-line learning. The Modified Condeg with Kalman Filter plus velocity factor is efficient and robust for servo and regulatory applications.

Key-Words: Ethanol, Extractive Process, Predictive Control, Adaptive Control, Artificial Neural Network, On-line Learning.

SUMÁRIO

AGRADECIMENTOS	ix
RESUMO	XIII
ABSTRACT.....	XV
SUMÁRIO.....	XVII
LISTA DE FIGURAS.....	XIX
NOMENCLATURA.....	XXI
1. INTRODUÇÃO.....	1
Organização do Trabalho.....	2
2. PROCESSO FERMENTATIVO PARA PRODUÇÃO DE ETANOL.....	4
2.1 Aspectos relevantes sobre o etanol como combustível	4
2.2 Processo Extrativo de Fermentação Alcoólica Contínua	8
2.2.1 Modelagem Matemática	10
2.2.2 Comportamento do Processo	17
2.2.3 Perturbações para Problema do Tipo Regulador e Servo.....	24
2.2.3.1 Problema do Tipo Regulador.....	24
2.2.3.2 Problema do Tipo Servo	27
2. Conclusões.....	28
3. REDES NEURAIS ARTIFICIAIS	29
3.1 História da Arte	29
3.1.1 Neurônio Artificial	32
3.2 Arquitetura das Redes Neurais Artificiais	35
3.3 Classificação das Redes.....	37
3.3.1 Aplicações na Área de Controle.....	37
3.3.2 Fluxo de Informações	38
3.3.3 Método de Identificação	39
3.4 Processo de Aprendizagem.....	39
3.4.1 Aprendizado Supervisionado.....	39
3.4.2 Aprendizado Não Supervisionado	39
3.5 Algoritmo de Aprendizagem <i>Backpropagation</i>	40
3.6 Controlador Neural – Dinâmica Inversa.....	47
3.7 Modelo Neural do Processo – Identificação.....	49
3.8 Aprendizagem <i>On-Line</i> e <i>Off-Line</i> das <i>RNA's</i>	50
3. Conclusões.....	54
4. ALGORITMOS DE CONTROLE.....	55
4.1 Controle de Processos.....	55

4.2 Controle Preditivo Baseado em Modelo.....	57
4.2.1 Controle por Matriz Dinâmica (DMC).....	60
4.2.2 Controle Baseado em Modelo Usando Redes (MPC Neural)	62
4.3 Controle Adaptativo	63
4.3.1 Controle Direto Baseado no Erro Global (Condeg Modificado).....	63
4.3.2 Condeg Modificado com Filtro de Kalman.....	67
4.3.2.1 Filtro de Kalman.....	67
4.3.2.2 Algoritmo Proposto para Estimar o Parâmetro de Penalização.....	71
4. Conclusões.....	74
5. RESULTADOS E DISCUSSÕES.....	76
5.1 Resultado para o Controle Preditivo.....	76
5.1.1 Problema do Tipo Servo	79
5.1.2 Problema do Tipo Regulador.....	81
5.1.3 Problema do Tipo Servo e Regulador.....	83
5.2 Resultado para o Controle Adaptativo	86
5.2.1 Problema do Tipo Servo	90
5.2.2 Problema do Tipo Regulador.....	94
5.2.3 Problema do Tipo Servo e Regulador.....	96
5.3 Resultado Comparativo dos Algoritmos Propostos.....	105
5. Conclusões.....	107
6. CONCLUSÕES E SUGESTÕES.....	108
6.1 Conclusões.....	108
6.2 Sugestões Para Trabalhos Futuros	111
7. REFERÊNCIAS BIBLIOGRÁFICAS	112

LISTA DE FIGURAS

Figura 2.1 Representação esquemática da planta de fermentação alcoólica extrativa	9
Figura 2.2 Perfil da Concentração de Substrato (S) para Perturbações Positivas	19
Figura 2.3 Perfil da Concentração de Produto(P) para Perturbações Positivas	20
Figura 2.4 Perfil da Concentração de Microrganismos (X) para Perturbações Positivas	20
Figura 2.5 Perfil da Concentração de Substrato (S) para Perturbações Negativas	21
Figura 2.6 Perfil da Concentração de Produto(P) para Perturbações Negativas	21
Figura 2.7 Perfil da Concentração de Microrganismos (X) para Perturbações Negativas	22
Figura 2.8. Definição das Variáveis Utilizadas	23
Figura 2.9. Perturbações No Processo Fermentativo para Variável S_0	24
Figura 2.10. Perturbações No Processo Fermentativo para Variável T_0	25
Figura 2.11. Resposta da Variável S para Perturbações em S_0 e T_0	25
Figura 2.12. Resposta da Variável P para Perturbações em S_0 e T_0	26
Figura 2.13. Resposta da Variável X para Perturbações em S_0 e T_0	27
Figura 3.1 Neurônio Artificial	32
Figura 3.2 Rede Neural Artificial (Tipo Feedforward)	33
Figura 3.3 Anatomia do i-ésimo neurônio artificial da k-ésima camada	34
Figura 3.4 Funções de ativação: (a) Função Sigmoidal; (b) Função Tangente Hiperbólica	35
Figura 3.5 Estruturas de Rede Feedforward (a,b,c) e Feedback (d,e)	38
Figura 3.6 Algoritmo Backpropagation	41
Figura 3.7 Arquitetura feedforward multivariável (Modelo Inverso)	49
Figura 3.8 Etapa 1 - Aprendizagem Off-line (Ajuste de Pesos com Dados Padrões)	51
Figura 3.9 Etapa 2 – Aprendizagem On-line (Ajuste de Peso em Tempo Real)	52
Figura 3.10 Esquema para Configuração da Aprendizagem On-line	52
Figura 4.1 Estrutura Básica de um Controle Preditivo (Fonte: Meleiro, 2002)	58
Figura 4.2 Algoritmo de Controle CONDEG Modificado	64
Figura 4.3 Algoritmo do Filtro de Kalman	69
Figura 5.1 MPC Neural com aprendizagem On-line e Off-line	77
Figura 5.2 Variáveis Controladas para Problema Servo em S	79
Figura 5.3 Variáveis Manipuladas para Problema Servo em S	79
Figura 5.4 Variáveis Controladas para Problema Regulador	81
Figura 5.5 Variáveis Manipuladas para Problema Regulador	80
Figura 5.6 Variáveis Controladas para Problema Servo S e Regulador	83
Figura 5.7 Variáveis Manipuladas para Problema Servo S e Regulador	84
Figura 5.8 Comparação de teta para Algoritmo do Filtro de Kalman 2	88
Figura 5.9 Comparação de teta para Algoritmo do Filtro de Kalman 4	88
Figura 5.10 Variáveis Controladas para Problema Servo S	90
Figura 5.11 Variáveis Manipuladas para Problema Servo S	91
Figura 5.12 Variáveis Controladas para Problema Servo P	92
Figura 5.13 Variáveis Manipuladas para Problema Servo P	93
Figura 5.14 Variáveis Controladas para Problema Regulador	94
Figura 5.15 Variáveis Manipuladas para Problema Regulador	95
Figura 5.16 Variáveis Controladas para Problema Servo S e Regulador	96
Figura 5.17 Variáveis Manipuladas para Problema Servo S e Regulador	97
Figura 5.18 Variáveis Controladas para Problema Servo P e Regulador	98
Figura 5.19 Variáveis Manipuladas para Problema Servo P e Regulador	99

Figura 5.20 Variáveis Controladas para Problema Servo X e Regulador	100
Figura 5.21 Variáveis Manipuladas para Problema Servo X e Regulador	101
Figura 5.22 Penalizações das Ações de Controle para Diferentes Valores de θ	104
Figura 5.23 Resultado Comparativo para Problema Servo em P	105
Figura 5.24 Resultado Comparativo para Problema Regulador	106
Figura 5.25 Resultado Comparativo para Problema Regulador e Servo em P	106

NOMENCLATURA

C_p	Calor específico (J/kgK)
d_j	Resposta alvo de um neurônio j da camada de saída
$\hat{e}(k)$	Vetor de erro global estimado da malha fechada, para o instante de tempo k
$e(t)$	erro para o instante de tempo t
F_w	Vazão de água (m ³ /h)
F_p	Vazão de purga (m ³ /h)
F_{pi}	Vazão da purga (m ³ /h)
F	Vazão de alimentação do fermentador (m ³ /h)
F_R	Vazão de reciclo de células (m ³ /h)
K_c	Constante Proporcional do PID
K_{eq}	Equilíbrio líquido – vapor
N_p	Número de padrões
n_k	Número de neurônios da camada k
n	Número de camadas ocultas
P	Concentração de etanol (kg/m ³)
$R_{microorg}$	Razão de reciclo de microrganismos
R_{flash}	Razão de reciclo do tanque flash
S	Concentração de substrato no fermentador (kg/m ³)

$S_i^{(k)}$	Função que indica a soma ponderada de todas as entradas de um neurônio i da camada k
S_0	Concentração de substrato na corrente de alimentação (kg/m^3)
T	Temperatura ($^{\circ}\text{C}$)
$u(t)$	Ação de controle no tempo t
$u(i, j)$	Ação de controle da malha i no tempo j
$U(k)$	Vetor de ações de controle
V	Volume do fermentador (m^3)
$w_{i,j}^{(k)}$	Peso da conexão entre o neurônio j da camada $(k-1)$ e o neurônio i da camada k
X	Concentração de microrganismos (kg/m^3)
X_v	Concentração de células viáveis (kg/m^3)
X_d	Concentração de células mortas (kg/m^3)
X_t	Concentração de células totais (kg/m^3)
$X_i^{(k)}$	Saída (ou ativação) de um neurônio i da camada k
$Y_w(k)$	Vetor de trajetória de referência da malha fechada
$y(t)$	Saída de um sistema no instante t
$y_{\text{setpoint}}(t)$	<i>Setpoint</i> de um sistema no instante t
$Y(k)$	Vetor de respostas da malha fechada
$\hat{Y}(k)$	Vetor de previsões da malha fechada

$y_r(i)$	Referência desejada para a malha de controle i
$Y_r(k)$	Vetor de referências da malha fechada
$y_w(i, j)$	Trajétoria de referência da malha i no tempo j
ΔH	Calor de reação ($2,167 \times 10^3$ J/kg de ART)
ε	Função objetivo para treinamento
$\delta_i^{(k)}$	Erro equivalente na aprendizagem
η	Taxa de aprendizagem
β	Conjunto de padrões passados usados na rede neural
α_i	Parâmetro de ajuste da trajetória de referência para a malha i
$\Delta u(i, j)$	Incremento na ação de controle i no tempo j
δu_j	Erro estimado na ação de controle da malha fechada j
δy_j	Erro estimado na resposta da malha fechada j ;
$\delta U(k)$	Vetor de erros estimados nas ações de controle da malha fechada
$\delta Y(k)$	Vetor de erros estimados nas respostas da malha fechada
$\lambda(i)$	Penalização na ação de controle da malha i
τ	Tempo de residência do fermentador
ρ_m	Densidade do meio fermentativo (1000 kg/m^3)
τ_I	Constante Integral do PID
τ_D	Constante Derivativa do PID

Subscritos

E	Corrente que segue para tanque flash
C	Corrente que segue para cuba
R	Corrente de reciclo
F	Corrente de alimentação

OBS: Todas as demais variáveis, parâmetros e símbolos empregados no texto são definidos ao longo do mesmo, à medida que são utilizados, a fim de facilitar a leitura.

INTRODUÇÃO

O etanol produzido a partir da cana-de-açúcar é, de longe, o biocombustível que apresenta a melhor relação custo/benefício atualmente no mercado. O Brasil tem hoje uma das melhores tecnologias para produção de etanol no mundo, fruto de evolução nos processos de fermentação e da eficiência energética da cana-de-açúcar em termos de sua produtividade de açúcares fermentescíveis.

A continuidade de estudos nessa linha de pesquisa para desenvolvimento de novas tecnologias e ampliação da produção de etanol é de suma importância para se ter processos eficientes e torná-los ainda mais competitivos. Dentro deste contexto, a utilização de um processo extrativo de fermentação que tem eficiência superior ao processo fermentativo convencional é justificável.

De uma maneira geral, os processos biotecnológicos são caracterizados por apresentarem um comportamento dinâmico complexo, como resposta inversa, tempo morto e forte não-linearidade, características inerentes à natureza complexa do metabolismo dos microrganismos, que são extremamente sensíveis à variações nas condições do meio fermentativo. Por estes motivos, a modelagem, simulação e controle destes processos configuram um desafio com grandes oportunidades.

Tendo como base as considerações feitas acima, emerge a necessidade do desenvolvimento de algoritmos de controle avançado para atuarem de forma efetiva nos processos biotecnológicos, em especial no processo extrativo de fermentação alcoólica, os quais deverão ter a capacidade de adaptação em função das perturbações externas que podem ocorrer.

Uma técnica de controle que permite esta característica de adaptação é o algoritmo de controle usando otimização em tempo real. Este algoritmo permite que o controlador tenha seus parâmetros atualizados ao longo do tempo e perceba as modificações no sistema. Para o desenvolvimento desses algoritmos de controle uma ferramenta que vem sendo muito empregada são as redes neurais artificiais (RNA's).

As RNA's têm se tornado foco de muita atenção principalmente devido à capacidade de aprendizado, ampla gama de aplicabilidade e facilidade com que armazenam conhecimento experimental e o torna disponível para o uso em aplicações de engenharia de processos.

O objetivo principal que norteia o desenvolvimento do presente trabalho foi desenvolver e avaliar diferentes algoritmos de controle para o processo extrativo de fermentação alcoólica. Os algoritmos de controle estudados foram:

- Controle por matriz dinâmica (DMC) clássico;
- Controle preditivo usando RNA's como modelo interno (MPC Neural);
- Algoritmo de controle Condeg Modificado com ajuste automático dos parâmetros do controlador usando filtro Kalman.

Para o procedimento de investigação foi utilizada a simulação computacional do processo extrativo de fermentação alcoólica. A análise computacional foi feita com simulações, utilizando um modelo para o processo de fermentação alcoólica acoplado a um tanque *flash* à vácuo, desenvolvido por Silva (1999). O modelo matemático usado foi baseado naquele desenvolvido por Costa *et al.* (2001) e os parâmetros cinéticos do processo fermentativo foram determinados experimentalmente por Atala *et al.* (2001) através de experimentos em batelada alimentada estendida. Todos os algoritmos de controle analisados foram implementadas em linguagem Fortran 90.

ORGANIZAÇÃO DO TRABALHO

O Capítulo 2 trata da produção de etanol, com ênfase no processo extrativo de fermentação alcóolica, que será o caso de estudo deste trabalho.

O Capítulo 3 apresenta como tema as redes neurais artificiais (RNA's), tendo como foco a aplicação das RNA's em controle de processos. Também serão enfatizadas as

características das redes aplicadas no desenvolvimento deste trabalho, como arquitetura e algoritmo de aprendizagem.

O Capítulo 4 aborda o controle de processos, onde o principal objetivo é apresentar os algoritmos de controle usados para o desenvolvimento deste trabalho, sendo estes: DMC, MPC Neural, Condeg Modificado e Condeg Modificado com filtro Kalman.

No Capítulo 5 são discutidos os resultados obtidos para cada um dos algoritmos de controle aplicados no processo de fermentação.

As conclusões finais e as sugestões para o prosseguimento das investigações tratadas nesse trabalho são apresentadas no Capítulo 6.

PROCESSO FERMENTATIVO PARA PRODUÇÃO DE ETANOL

Neste capítulo são apresentadas as vantagens da utilização de etanol como biocombustível, destacando as pesquisas na área de produção por via fermentativa e apresentado o processo extrativo de fermentação alcoólica, como caso de estudo.

2.1 ASPECTOS RELEVANTES SOBRE O ETANOL COMO COMBUSTÍVEL

As principais causas do aquecimento global são as emissões e o aumento da concentração de dióxido de carbono na atmosfera, principalmente por causa do uso de combustíveis fósseis, como o petróleo. O dióxido de carbono se concentra na atmosfera impedindo a saída do calor, causando o efeito estufa.

O uso de etanol como combustível apresenta uma alternativa vantajosa para diminuição do efeito estufa. Macedo (1997) realizou um estudo sobre as emissões dos gases de efeito estufa, evitadas pelo uso do etanol e bagaço da cana-de-açúcar no Brasil durante o período de 1996. De acordo com o autor, o carbono emitido à atmosfera quando o etanol e o bagaço são consumidos como combustíveis são compensados por uma quantidade equivalente de carbono seqüestrado pela cana-de-açúcar durante seu crescimento pela fotossíntese, fazendo com que a aplicação do etanol tenha menor impacto no efeito estufa. Ainda que a colheita seja feita com queimadas em alguns lugares, o ciclo é vantajoso.

Além disso, a tecnologia para destilar o etanol de cana e misturá-lo à gasolina é comparativamente barata e fácil de obter (Constance, 2007). Praticamente todos os novos automóveis e a maioria dos mais antigos podem funcionar com gasolina misturada a até 10% de etanol.

De acordo com Carnicelli (2004), o aumento do consumo de álcool combustível ganhou força com o lançamento em 2003 do primeiro veículo bi-combustível, movido a álcool ou gasolina. A participação desses veículos em 2003 foi de 3,4% e em 2006 de 78,1% das vendas totais (Globo Notícias, 2007).

Os processos de fermentação alcóolica brasileiros surgiram nos velhos engenhos de cana-de-açúcar como processos de produção de aguardente a partir do caldo-de-açúcar. No início do século XIX surgiu a produção do etanol por via fermentativa em escala industrial. Indústrias vinculadas às usinas de fabricação de açúcar transformavam o melaço, considerado um subproduto sem grande valor comercial a um produto de maior valor agregado, o etanol (Andrieta, 1994).

O processo de fermentação alcóolica foi aprimorado na década de 1930 na França, através da produção de álcool em regime de batelada alimentada (processo Melle – Boinot). No Brasil, a tecnologia começou a ser empregada somente em meados da década de 1970.

A produção de etanol no Brasil teve seu grande impulso após a primeira grande crise do petróleo em 1973, sendo em 1975 criado o Proálcool, um programa federal, administrado pelo Ministério da Indústria e Comércio através da Comissão Executiva Nacional do Álcool, que tinha como objetivo o aumento de safras agro-energéticas e a capacidade industrial de transformação, visando à obtenção de álcool para substituir o petróleo e seus derivados (Santos *et al.*, 2000). Programas similares para o desenvolvimento de tecnologias para utilização do etanol como uma alternativa viável para combustível foram criados nos Estados Unidos e Canadá (Wheals *et al.*, 1999).

Na primeira década do programa Proálcool, o governo avaliou o balanço positivo: na safra de 1985 foram produzidos 11 bilhões de litros de álcool, gerando muitos empregos diretos e indiretos. Os carros movidos a álcool e a mistura de álcool a gasolina reduziram o nível de poluição ambiental (Neiva, 1987).

Os esforços empreendidos por empresários e pesquisadores nesse período foram direcionados para otimização dos processos de batelada alimentada e o desenvolvimento e

implementação de processos contínuos em escala industrial (Andrieta e Stupiello, 1990a, Andrieta e Stupiello, 1990b; Wheals *et al.*, 1999).

De acordo com Meleiro (2002), no Brasil, o etanol chegou a ser pensado como fonte alternativa para frota automotiva nacional e a solução para dependência da importação de petróleo da indústria brasileira. A análise das características físico-químicas do etanol o classificou como excelente combustível automotivo, pois apresenta diversas características superiores quando comparado com a gasolina, entre elas: maior octanagem, menores emissões evaporativas devido à pressão de vapor inferior, menor combustão no ar, que reduz a severidade e número de fogo nos veículos.

Segundo Atala (2004), a partir de 1986, os preços do petróleo se estabilizaram, com tendência de queda. A Petrobrás, em alguns casos, chegou a pagar mais caro pelo litro de álcool do que o preço cobrado nos postos de abastecimento. Os preços do diesel e da gasolina foram artificialmente elevados para cobrir parte do déficit, levando o Proálcool a um processo de estagnação. A produção de álcool se estabilizou em 12 bilhões de litros por safra. Surgindo nesse período uma escassez de álcool carburante, pois a fabricação de carros a álcool atingiu ao fim da década de 80, cerca de 95% dos veículos comercializados.

Em 1994, quando entrou em vigor a lei de proteção ambiental que obrigou a mistura de 22% de álcool à gasolina, o déficit chegou a um bilhão de litros. A introdução da mistura gasolina/álcool teve um impacto imediato na qualidade do ar das grandes cidades, pois eliminava a necessidade de aditivos como o chumbo. Além disso, os hidrocarbonetos aromáticos também foram eliminados e o conteúdo de enxofre da gasolina foi diminuído. Nos automóveis movidos a álcool puro, a emissão de enxofre foi eliminada. Outra comparação importante leva em consideração as emissões de monóxido de carbono (CO) que, antes de 1980, quando era utilizada gasolina pura, eram superiores a 50 g/km e em 1995 reduzidas para menos de 5,8 g/km (Moreira e Goldemberg, 2002).

O grande erro do programa Proálcool segundo Encarnação Júnior (2002) foi o mesmo não ser auto-suficiente, ou seja, não se atingiu viabilidade econômica, pois o custo da produção do álcool era superior ao seu preço de venda.

Atualmente o Brasil produz 17,5 bilhões de litros e diversos esforços estão sendo realizados para aumentar ainda mais esse valor, desde 2004 o governo tem mostrado interesse em incentivar o aumento de sua produção, sua exportação e incentivos fiscais para sua estocagem (Atala, 2004; Agência-Udop, 2004; Cunha, 2004; Agência-Brasil, 2004).

Segundo Atala (2004) a produção de etanol por via fermentativa é o processo mais utilizado e aplicado pelas indústrias produtoras desta matéria prima. No Brasil, a cana-de-açúcar se destaca pela disponibilidade e custo, como principal meio utilizado na fermentação, pois se trata de uma rica fonte de carbono com elevada concentração de açúcares de sacarose, seguido pela glicose e frutose.

As plantas contínuas apresentam uma maior produtividade comparada com as plantas descontínuas e semi-contínuas, algumas pesquisas nessa área foram desenvolvidas no Brasil por Rodrigues *et al.* (1992), Andrieta e Maugeri Filho (1994).

Os sistemas contínuos são caracterizados por possuir uma linha de retirada do vinho e outra de alimentação do substrato no tanque de fermentação, que pode ser com ou sem recirculação de células (Siqueira, 1997).

Atala (2004) cita as vantagens e desvantagens dos processos fermentativos contínuos. Este tipo de processo apresenta uma maior produtividade, uniformidade do produto e adaptabilidade ao controle automático, ausência de paradas para carga, descarga e limpeza dos tanques de fermentação e também pelas células estarem mais adaptadas ao meio (Alves, 1996). Os pontos negativos de acordo com Siqueira (1997) são: a dificuldade de manutenção de um meio estéril por muito tempo e a ocorrência de mutantes não interessantes ao processo de fermentação especialmente quando estes são conduzidos por longos períodos de tempo.

2.2 PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

O etanol que se acumula no meio fermentativo inibe a atividade metabólica do microorganismo, e de acordo com Alves (1996) é o primeiro fator a ser reconhecido como inibidor da fermentação alcoólica e que as leveduras dos gêneros *Saccharomyces* e *Schizosaccharomyces* são as consideradas de melhor resistência ao efeito tóxico do álcool.

Devido a esta inibição da atividade metabólica do microorganismo a utilização de técnicas de extração do etanol do meio fermentativo assim que ele é produzido, melhora o desempenho do processo.

Silva (1997) propôs um sistema que acopla o fermentador a um evaporador *Flash* operando a uma temperatura que elimine a necessidade de trocadores de calor no estágio fermentativo, o que reduz drasticamente os custos fixos e de manutenção. A remoção parcial do etanol produzido até níveis de 40 g/L no meio fermentativo, reduz significativamente o poder inibitório exercido na atividade metabólica do microorganismo, por outro lado, garante a ação anti-séptica do meio.

O evaporador *flash* equivale a um estágio de uma coluna de destilação. O sistema opera em pressões reduzidas (vácuo), facilitando a separação da mistura etanol/água, em temperaturas baixas.

Silva *et al.* (1999) propuseram um processo extrativo de fermentação alcoólica com este princípio e que apresentou alta produtividade comparada aos processos convencionais.

Para o desenvolvimento dos algoritmos de controle através da análise computacional, o processo extrativo de fermentação alcoólica utilizado é o proposto por Silva *et al.* (1999), representado na Figura 2.1.

A planta é formada por quatro unidades interligadas:

- *Fermentador* – unidade onde o etanol é produzido;

- *Centrífuga* – unidade responsável pela separação das fases sólida (microorganismo) e líquida (mosto e etanol);
- *Cuba* – unidade de tratamento dos microrganismos;
- *Tanque Flash* – unidade de separação da mistura etanol – mosto (fase aquosa).

O tanque flash é utilizado para extrair continuamente parte do etanol formado, aumentando desta forma a produtividade, como discutido anteriormente.

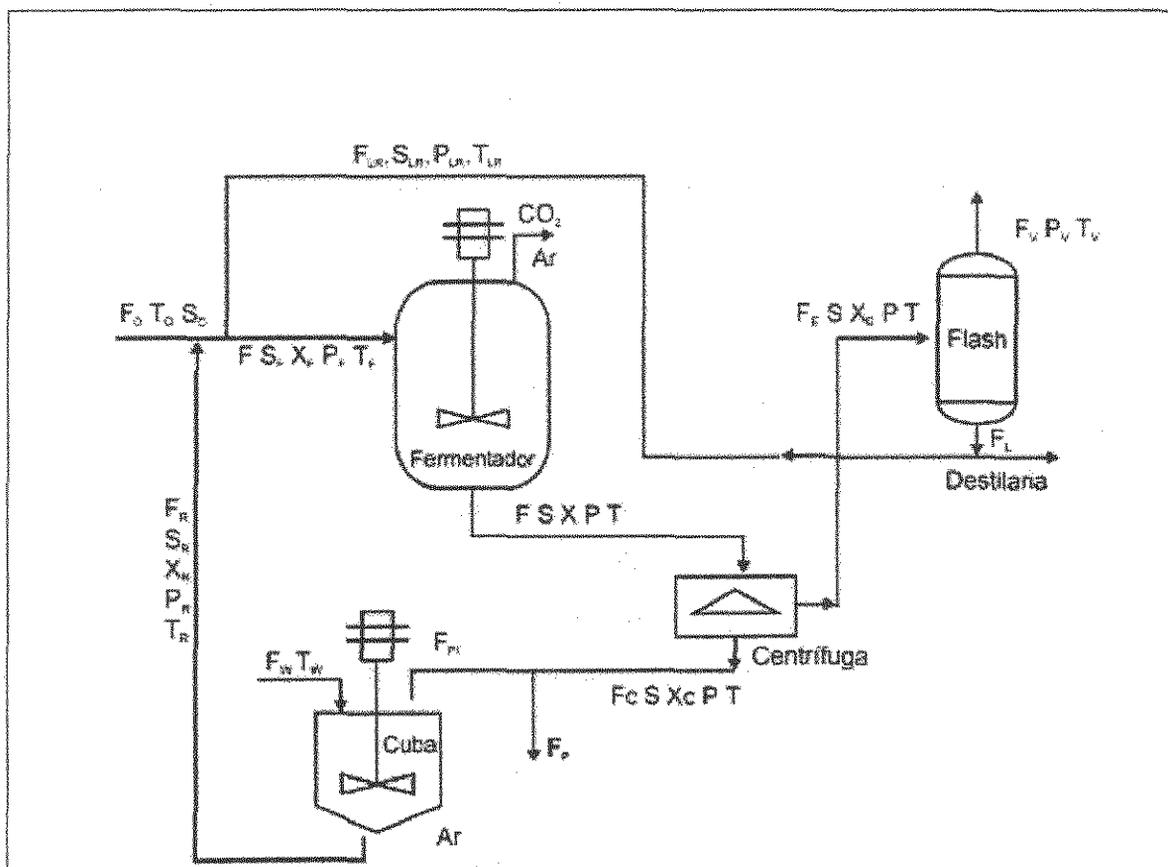


Figura 2.1 Representação esquemática da planta de fermentação alcoólica extrativa

De acordo com Meleiro (2002), o processo inicia-se com a alimentação de substrato no fermentador, que contém a concentração de microrganismos adequada para dar início ao processo fermentativo. A corrente de saída do fermentador, contendo substrato, células e etanol, é enviada para centrífuga e separada em duas fases: fase pesada, contendo

células e etanol, é enviada para centrífuga e separada em duas fases: fase pesada, contendo maior parte das células (microrganismos); e a fase leve, praticamente isenta de células. A fase pesada, após sofrer uma purga para eliminar parte das células é enviada para a cuba. Na cuba, o creme de leveduras é diluído com água e acidulado com ácido sulfúrico, sob forte agitação, para diminuir a contaminação do processo por bactérias e desfloculação da levedura. A suspensão de células tratadas é enviada para o fermentador para um novo ciclo do processo.

Após o estado estacionário ser atingido o sistema de separação a vácuo é ativado e a fase leve que sai da centrífuga é enviada para o tanque *flash*, onde parte do etanol é separado da mistura aquosa.

A corrente vaporizada e parte da corrente líquida que deixam o *flash* são enviadas para uma coluna de retificação. A outra parte da corrente líquida oriunda do flash retorna ao fermentador com o objetivo de manter a concentração de etanol em um nível mínimo no qual ele apresente propriedades anti-sépticas.

2.2.1 Modelagem Matemática

A modelagem matemática do processo considerado consiste em balanços de massa e energia. Todos os equipamentos, com exceção do fermentador, são modelados assumindo a condição de estado pseudo-estacionário. Segundo Costa *et al.* (2002) esta aproximação é adequada porque a dinâmica do fermentador é muito lenta quando comparada com as dinâmicas das demais unidades, sendo, portanto, a dominante do sistema.

O modelo matemático para o fermentador utilizado foi desenvolvido por Costa *et al.* (2001), que consideraram a queda de viabilidade celular ao longo da fermentação, de modo que a concentração de microrganismos foi dividida em duas partes: uma ativa (células viáveis) e outra inativa (células mortas). Os parâmetros cinéticos utilizados na modelagem do mesmo equipamento foram determinados experimentalmente por Atala *et al.* (2001).

Segundo Costa *et al.* (2001), para se determinarem as concentrações e as vazões de entrada para o fermentador, fez-se um balanço de massa global no processo assumindo-se as seguintes aproximações:

- As concentrações de substrato e de produto na saída da centrífuga, nas fases leve e pesada, são iguais às concentrações na saída do fermentador;
- A concentração de biomassa na corrente de reciclo é fixa. Para isto, ajusta-se a vazão de água que dilui o creme de leveduras;
- A vazão de reciclo de células, cujo valor é determinado pela razão de reciclo de microrganismos é mantida constante através da variação da vazão de purga. A purga serve para retirar produtos secundários acumulados no fermentador e permite uma constante renovação celular.

De acordo com estas aproximações, o modelo matemático do processo é descrito pelas equações descritas a seguir:

- Vazão de Alimentação do fermentador:

$$F = F_O + F_{LR} + F_R \quad (2.1)$$

- Vazão de Reciclo de células:

$$F_R = F \times R_{microrg} \quad (2.2)$$

Substituindo o valor de F_R (equação 2.2) na equação 2.1, tem-se:

$$F = \frac{F_O + F_{LR}}{(1 - R_{microrg})} \quad (2.3)$$

- Tempo de Residência do Fermentador:

$$\tau = \frac{V}{F} \quad (2.4)$$

onde V é o volume do reator.

- Balanço de Biomassa (Microrganismos) para Centrífuga:

$$FX = F_E X_E + F_C X_C \quad (2.5)$$

onde X representa a concentração de microrganismos e os índices E e C estão associados às correntes que seguem para o flash e para a cuba, respectivamente.

- Balanço Global Centrífuga:

$$F_C = F - F_E \quad (2.6)$$

- Vazão de Alimentação do Tanque Flash:

$$F_E = \frac{F(X - X_C)}{X_E - X_C} \quad (2.7)$$

- Vazão de Alimentação da Cuba:

$$F_{Cl} = \frac{F_R X_R}{X_C} \quad (2.8)$$

- Vazão de Água de Diluição que Alimenta a Cuba:

$$F_W = F_R + F_{Cl} \quad (2.9)$$

- Concentração de Substrato na Corrente de Reciclo:

$$S_R = \frac{F_{Cl} S}{F_R} \quad (2.10)$$

- Concentração de Etanol na Corrente de Reciclo:

$$P_R = \frac{F_{C1}P}{F_R} \quad (2.11)$$

- Balanço de Massa no Tanque Flash:

$$F_E = F_V + F_L \quad (2.12)$$

$$F_E X_{Ei} = F_V y_i + F_L x_i \quad (2.13)$$

- Equilíbrio Líquido Vapor da Mistura Etanol e Água:

$$k_{eqi} = \frac{y_i}{x_i} = \gamma_i \frac{P_i^{sat}}{P} \quad (2.14)$$

onde o valor de P_i^{sat} é dado pela equação de Antoine e o valor de γ_i é calculado com o modelo *NRTL* (Silva, 1997).

As concentrações de substrato, etanol e microrganismos presentes na corrente que retorna para o fermentador, vindas do tanque flash seguem as equações 2.15, 2.16 e 2.17, respectivamente.

$$S_{LR} = \frac{F_E S}{F_L} \quad (2.15)$$

$$P_{LR} = \frac{F_E P - F_V P_V}{F_L} \quad (2.16)$$

$$X_{LR} = \frac{F_E X_E}{F_L} \quad (2.17)$$

- Vazão de Reciclo do tanque Flash;

$$F_{LR} = R_{flash} F_L \quad (2.18)$$

onde R_{flash} é a razão de reciclo do tanque Flash.

UNICAMP BIBLIOTECA CENTRAL CÉSAR LATTES DESENVOLVIMENTO DE COLEÇÃO

As concentrações de substrato, etanol e microrganismos na corrente de alimentação do fermentador são dadas pelas equações 2.19, 2.20 e 2.21, respectivamente.

$$S_F = \frac{F_O S_O + F_{LR} S_{LR} + F_R S_R}{F} \quad (2.19)$$

$$P_F = \frac{F_R P_R + F_{LR} P_{LR}}{F} \quad (2.20)$$

$$X_F = \frac{F_R X_R + F_{LR} X_{LR}}{F} \quad (2.21)$$

Os balanços de energia em torno da Cuba e do Fermentador fornecem as temperaturas nas correntes de reciclo da Cuba e de alimentação do Fermentador, respectivamente:

$$T_R = \frac{F_W T_W + F_{Cl} T}{F_R} \quad (2.22)$$

$$T_F = \frac{F_O T_O + F_R T_R + F_{LR} T_{LR}}{F} \quad (2.23)$$

Assumindo que o volume do reator é constante e considerando o volume ocupado pela célula (2.26) e a concentração intracelular (2.27), as equações de balanço de massa e energia para o fermentador, utilizando o modelo intrínseco, são escritas da seguinte forma:

$$\frac{dX_v}{dt} = r_x - r_d - \frac{F}{V} (X_v - X_{vF}) \quad (2.24)$$

$$\frac{dX_d}{dt} = r_d - \frac{F}{V} (X_d - X_{dF}) \quad (2.25)$$

$$\frac{d\left(1 - \frac{X_t}{\rho}\right)SV}{dt} = F(S_F - S) - r_s V \quad (2.26)$$

$$\frac{d\left[\left(1 - \frac{X_t}{\rho}\right)PV + \frac{X_t}{\rho}\Gamma PV\right]}{dt} = r_p V + F(P_F - P) \quad (2.27)$$

$$\frac{dT}{dt} = \frac{F}{V}(T_F - T) + \frac{\Delta H r_s}{\rho_m C_p} \quad (2.28)$$

onde:

X_v é a concentração de células viáveis;

X_d é a concentração de células mortas;

X_t é a concentração de células totais;

S é a concentração de substrato;

Sendo que o indicador F indica na alimentação;

P é a concentração de produto;

T é a temperatura do fermentador;

ΔH é o calor de reação ($2,167 \cdot 10^3$ J/kg de ART);

ρ_m é a densidade do meio fermentativo (1000 kg/m³);

C_p é o calor específico ($4,183 \cdot 10^3$ J/kgK);

F é a vazão de alimentação, V é o volume do fermentador, ρ é a relação de massa seca pelo volume de massa úmida e Γ é a relação entre as concentrações de etanol intra e extra-celular.

- Taxa Cinética de Crescimento

$$r_x = \mu_{\max} \frac{S}{K_S + S} \exp(-K_i S) \left(1 - \frac{X_t}{X_{\max}}\right)^m \left(1 - \frac{P_t}{P_{\max}}\right)^n X_v \quad (2.29)$$

- Taxa de Morte

$$r_d = K_{dT} X_v \exp(K_{dP} P) \quad (2.30)$$

- Consumo de Substrato

$$r_s = \frac{r_x}{y_x} + m_x X_v \quad (2.31)$$

- Formação de Produto

$$r_p = Y_{pX} r_x + m_p X_v \quad (2.32)$$

Os parâmetros cinéticos utilizados foram determinados experimentalmente, para faixa de 28 a 40°C por Atala *et al.* (2001) e são descritos na Tabela 2.1.

Tabela 2.1 Parâmetros Cinéticos em Função da Temperatura (em °C)

Parâmetros	Expressão ou Valor
μ_{\max}	$1,57 \exp\left(\frac{-41,47}{T}\right) - 1,29 \cdot 10^4 \exp\left(\frac{-431,4}{T}\right)$
X_{\max}	$-0,3279 T^2 + 18,484 T - 191,06$
P_{\max}	$-0,4421 T^2 + 26,41 T - 279,75$
B	$2,704 \exp(-0,1225 T)$
Y_{PX}	$0,2556 \exp(-0,1086 T)$
K_s	4,1
K_i	$1,393 \cdot 10^{-4} \exp(-0,1004 T)$
m_p	0,1
m_x	0,2
m	1
n	1,5
K_{dP}	$7,421 \cdot 10^{-3} T^2 - 0,4654 T + 7,69$
K_{dT}	$4 \cdot 10^{13} \exp\left(\frac{41947}{1,987(T + 273,15)}\right)$
ρ	390
Γ	0,78

2.2.2 Comportamento do Processo

Os parâmetros estacionários adotados para as simulações são apresentados na Tabela 2.2, que seguem Meleiro (2002).

Tabela 2.2 Parâmetros Estacionários Adotados

<i>Variável</i>	<i>Valor Estacionário</i>
To	30 ° C
So	180 kg/m ³
Fo	100 kg/m ³
R _{microorg}	0,35
R _{flash}	0,4
S	5,49 kg/m ³
P	41,12 kg/m ³
X	34,81 kg/m ³

Para observar o comportamento dinâmico do fermentador foram feitas perturbações positivas e negativas nas variáveis de entrada: vazão de alimentação (Fo), razão de reciclo de microorganismo (R_{microorg}), razão de reciclo do tanque flash (R_{flash}), temperatura de alimentação (To) e Concentração de substrato na alimentação (So).

A Tabela 2.3 mostra o valor de cada variável após as perturbações positivas (+) e negativas (-) dos valores estacionários.

Tabela 2.3 Valores das Variáveis de Entrada após Perturbações Especificadas

<i>Variável</i>	<i>Perturbação (+)</i>	<i>Estacionário</i>	<i>Perturbação (-)</i>
To	35° C	30° C	25° C
So	200 kg/m ³	180 kg/m ³	160 kg/m ³
Fo	120 m ³ /h	100 m ³ /h	80 m ³ /h
R _{microorg}	0,5	0,35	0,2
R _{flash}	0,5	0,4	0,2

Nas Figuras 2.2, 2.3 e 2.4, pode se observar os perfis de concentração do substrato (S), produto (P) e microorganismo (X), respectivamente, para as perturbações positivas dos valores estacionários, conforme descrito na Tabela 2.3.

Nas Figuras 2.5, 2.6 e 2.7 pode-se observar os perfis de concentração do substrato (S), produto (P) e microorganismo (X), respectivamente, para as perturbações negativas dos valores estacionários, conforme descrito na Tabela 2.3.

Nestas Figuras pode-se observar o comportamento dinâmico do fermentador para as perturbações impostas nas variáveis de entrada, onde se pode observar a dinâmica do processo estudado bem como a presença de não-linearidades.

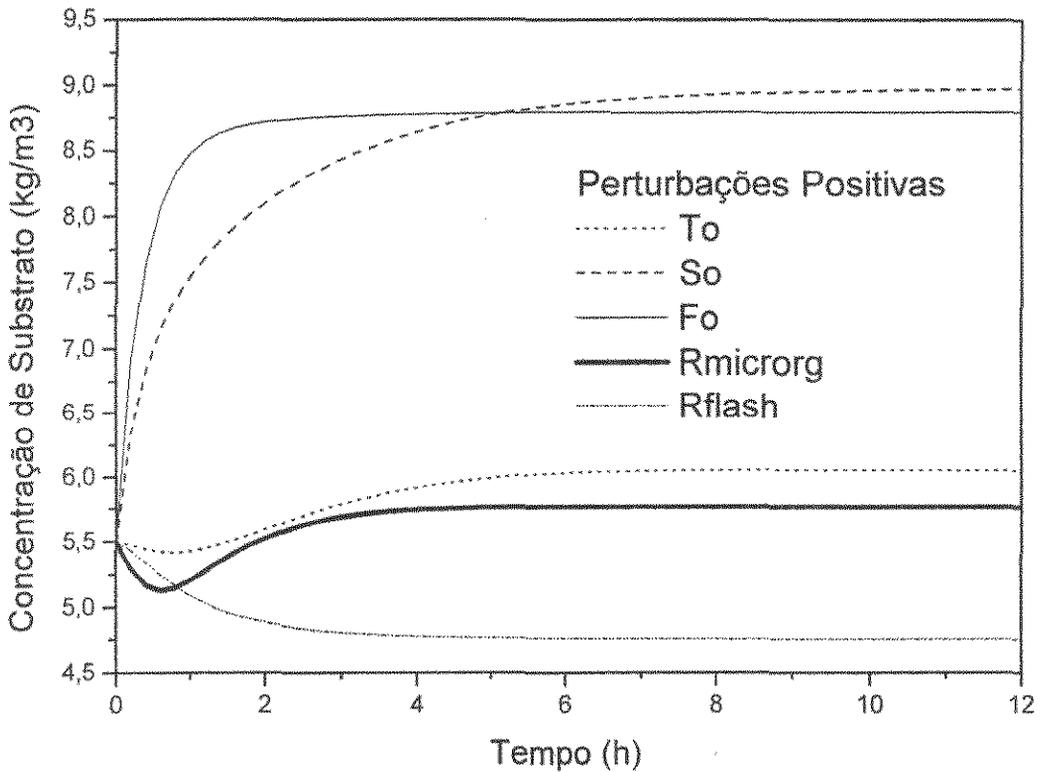


Figura 2.2 Perfil da Concentração de Substrato (S) para Perturbações Positivas

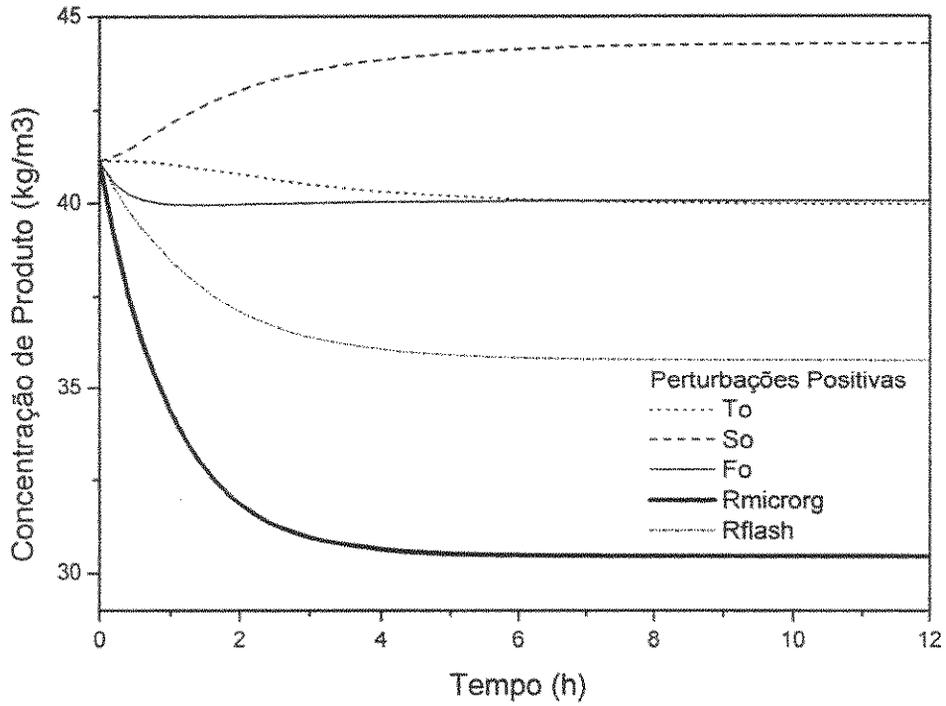


Figura 2.3 Perfil da Concentração de Produto(P) para Perturbações Positivas

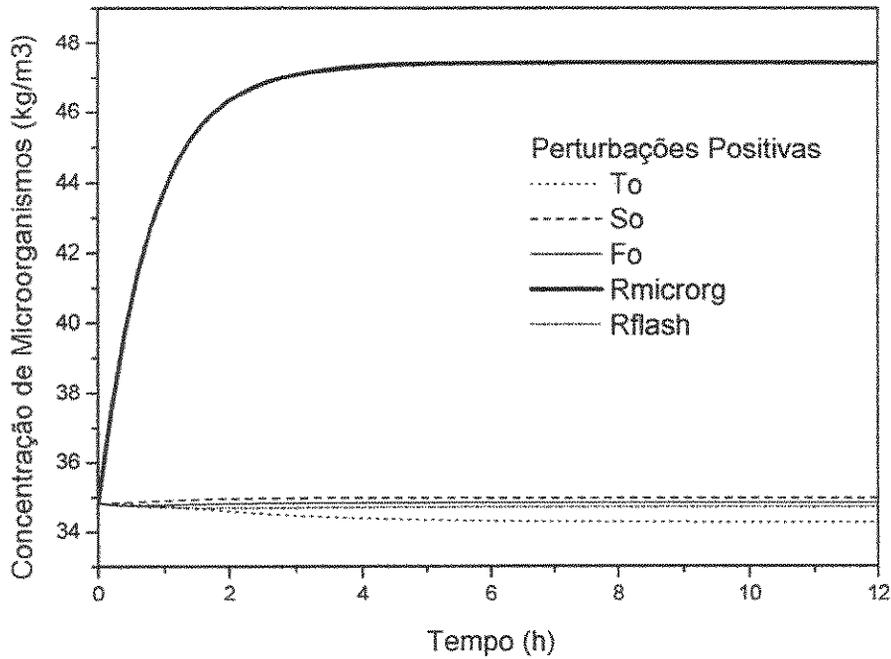


Figura 2.4 Perfil da Concentração de Microorganismos (X) para Perturbações Positivas

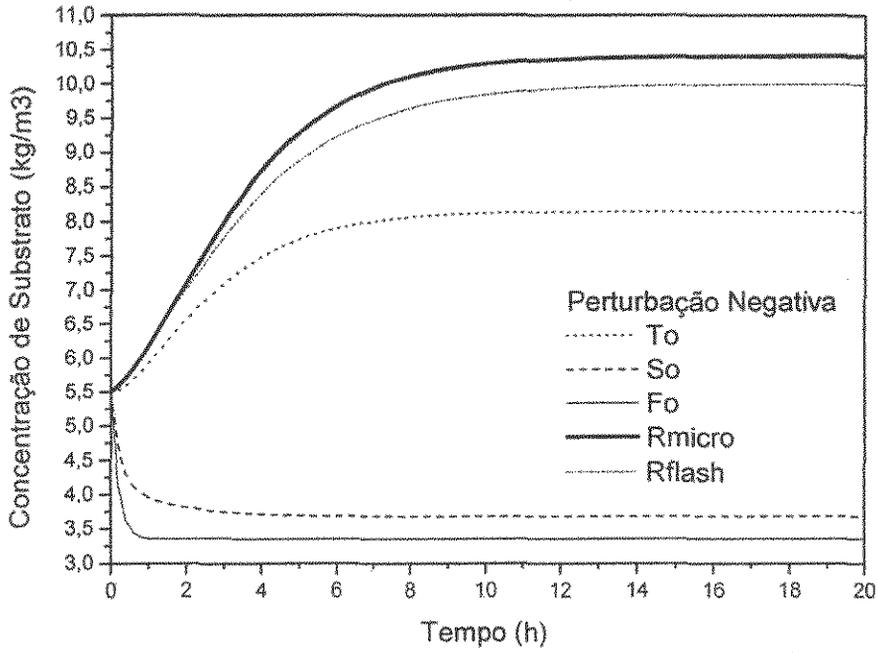


Figura 2.5 Perfil da Concentração de Substrato (S) para Perturbações Negativas

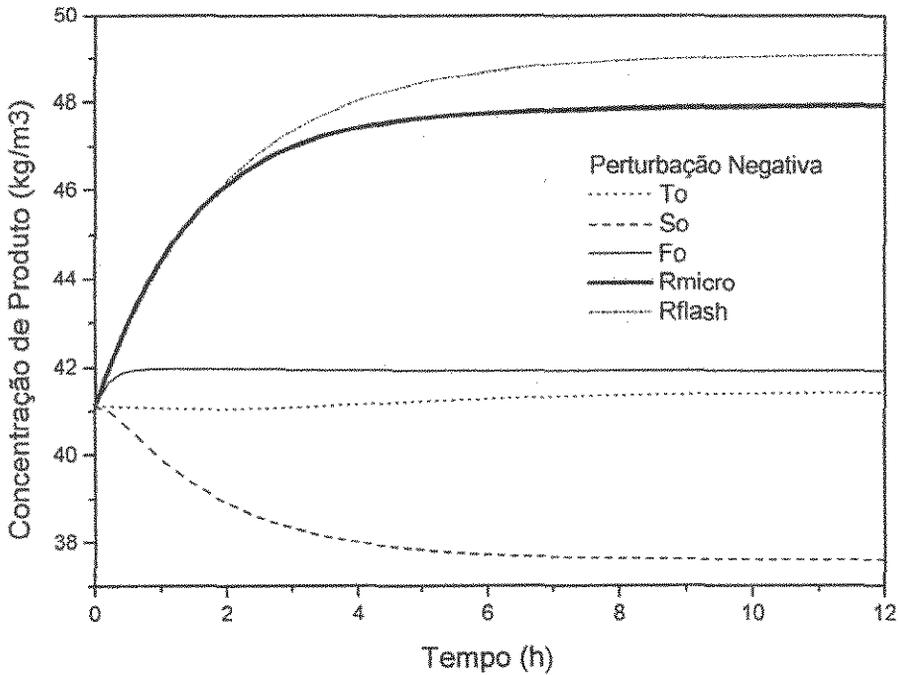


Figura 2.6 Perfil da Concentração de Produto (P) para Perturbações Negativas

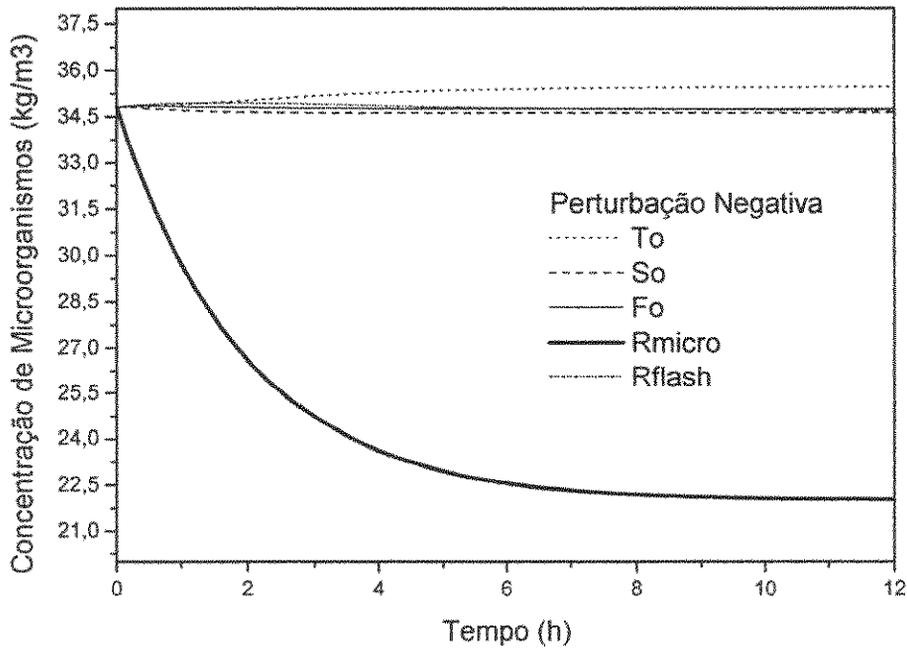


Figura 2.7 Perfil da Concentração de Microrganismos (X) para Perturbações Negativas

Para determinação das variáveis de entrada, saída e perturbação para este processo fermentativo, Silva *et al.* (1999) utilizaram os métodos de planejamento fatorial e análise da superfície de resposta gerada a partir de dados de simulação para determinar as variáveis de processo mais relevantes.

Costa *et al.* (2001) também fizeram um estudo completo utilizando a mesma técnica combinada com a simulação dinâmica do processo para determinar como as variáveis de saída (controladas) variam no tempo de acordo com mudanças impostas nas variáveis de entrada, sendo estas variáveis manipuladas e perturbadas.

De acordo com estes estudos foram escolhidas as principais variáveis de interesse para este caso de estudo, conforme descrito a seguir e esquematizado na Figura 2.8.

Variáveis de Entrada:

- Vazão da Corrente de Alimentação: F_0 (m³/h);

- Razão de Reciclo de Microrganismos: R_{microorg} (adimensional);
- Razão de Reciclo do Tanque Flash: R_{flash} (adimensional).

Variáveis de Saída:

- Concentração de Etanol no Fermentador: P (kg/m^3);
- Concentração de Substrato no Fermentador: S (kg/m^3);
- Concentração de Microrganismos no Fermentador: X (kg/m^3).

Variável Perturbação:

- Concentração de Substrato na Corrente de Alimentação: S_0 (kg/m^3);
- Temperatura na Corrente de Alimentação: T_0 ($^{\circ}\text{C}$).

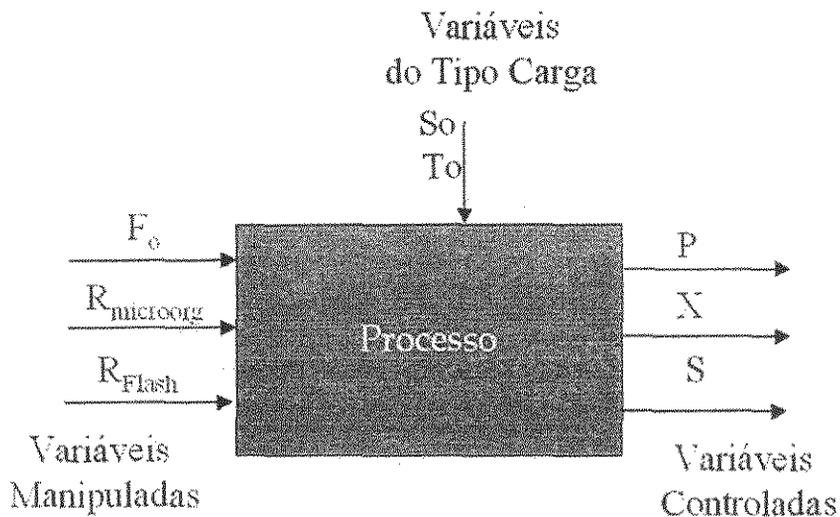


Figura 2.8. Definição das Variáveis Utilizadas

O período de amostragem foi escolhido em função do tempo médio de análise de um cromatógrafo (equipamento utilizado para medir as concentrações de ART e etanol), pois o turbidímetro que mede a concentração celular opera em intervalos pequenos.

Segundo Andrieta (1994) doze minutos é o tempo de amostragem adequado para o processo.

2.2.3 Perturbações para Problema do Tipo Regulador e Servo

Para verificar o desempenho dos algoritmos analisados, foram estudados problemas dos tipos, servo e regulador.

No problema do tipo servo é realizada uma mudança no valor do estado estacionário desejado (*setpoint*) e o objetivo do controle é levar a saída ao novo estado estacionário.

Para o problema do tipo regulador a perturbação, ou carga, varia de maneira imprevisível e o objetivo do controle é manter a saída num valor desejado.

2.2.3.1 Problema do Tipo Regulador

Como problemas reguladores foram usadas as variáveis perturbações S_0 e T_0 . Estas foram submetidas a oscilações dentro de suas faixas de operação assumindo valores diferentes ao longo do tempo em todas as simulações, gerando uma oscilação no processo e a necessidade de um controlador que seja robusto e efetivo. As Figuras 2.9 e 2.10 descrevem as perturbações-cargas impostas em S_0 e T_0 ao longo do tempo, respectivamente.

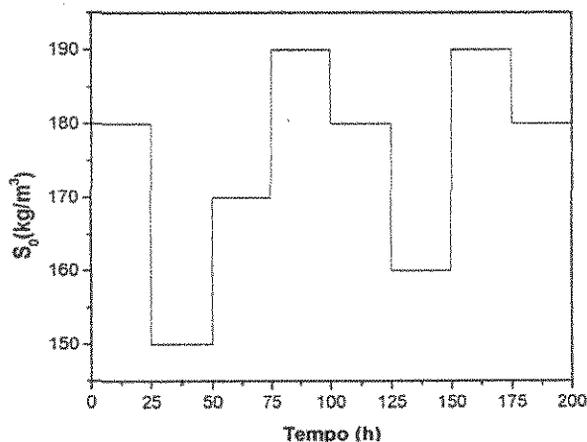


Figura 2.9. Perturbações No Processo Fermentativo para Variável S_0

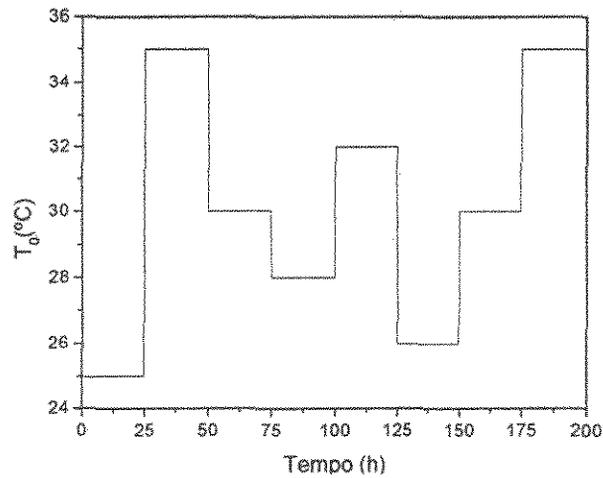


Figura 2.10. Perturbações No Processo Fermentativo para Variável T_0

Os resultados do processo em malha aberta para a variável de saída S , para perturbações agindo de forma independente e conjunta das variáveis S_0 e T_0 são apresentados na Figura 2.11.

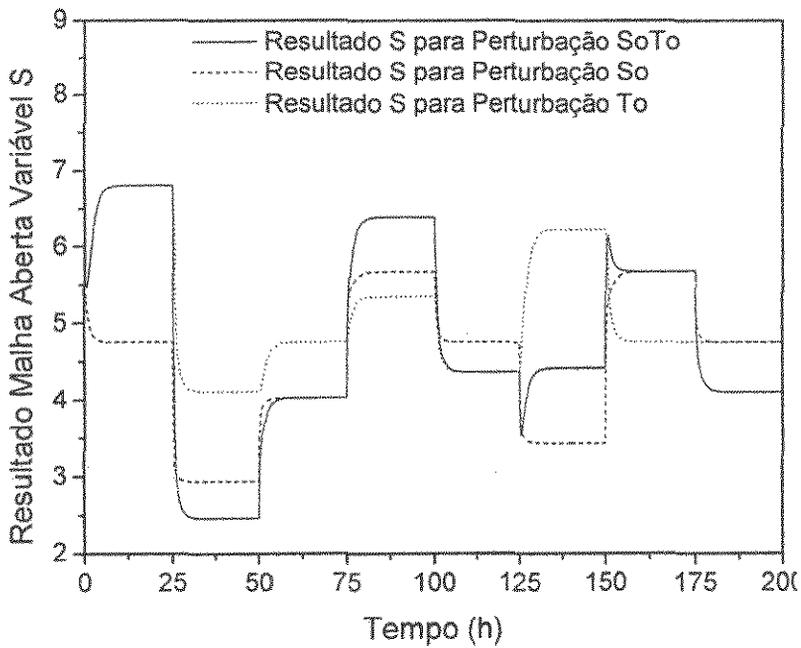


Figura 2.11. Resposta da Variável S para Perturbações em S_0 e T_0

Analisando os resultados apresentados na Figura 2.11, observa-se que, como esperado, a variação da quantidade de substrato inicial (S_0) interfere diretamente na quantidade de substrato final (S) e a variável temperatura (T_0) também influencia o consumo de substrato no processo (S). Por isso, a variação conjunta dessas duas variáveis (S_0 e T_0) gera uma resposta em malha aberta diferenciada de acordo com as perturbações impostas ao processo fermentativo.

Na Figura 2.12 são apresentados resultados da variável P para as perturbações de S_0 e T_0 de forma individual e conjunta para o processo fermentativo em malha aberta. Como se observa, a temperatura inicial não tem muita influência quando comparada com a concentração de substrato inicial no processo para esta variável.

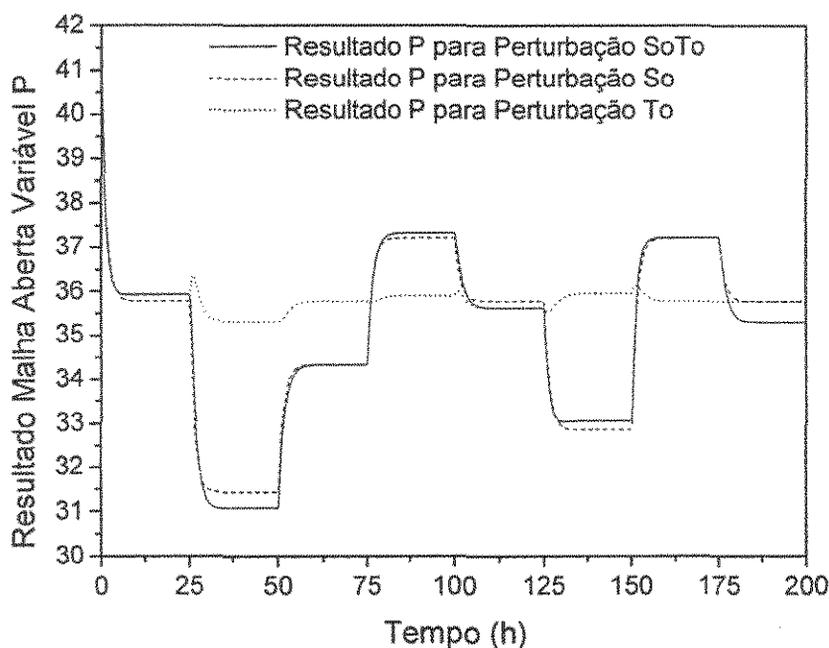


Figura 2.12. Resposta da Variável P para Perturbações em S_0 e T_0

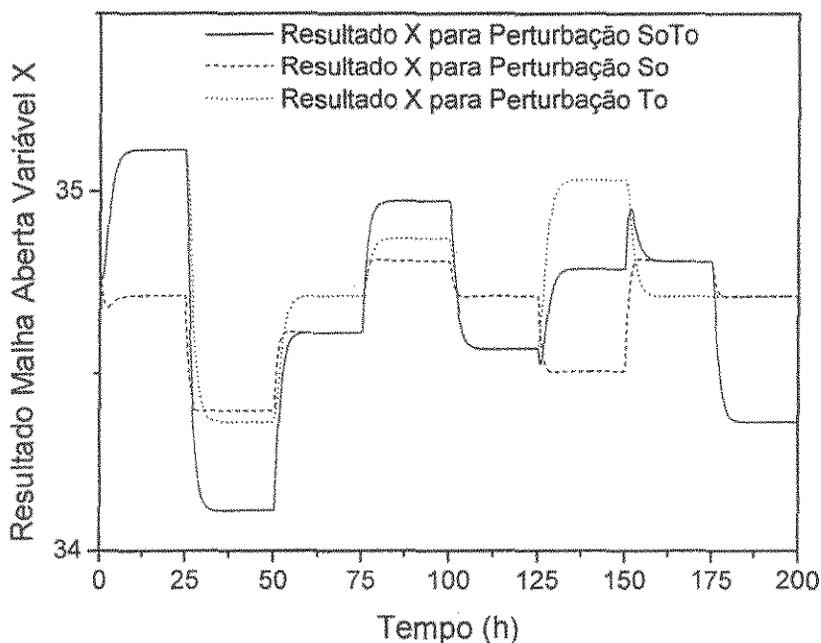


Figura 2.13. Resposta da Variável X para Perturbações em So e To

Como se observa na Figura 2.13, a temperatura inicial (T_0) e a concentração inicial de substrato (S_0) influenciam diretamente a concentração final de microrganismos (X). De forma similar ao resultado obtido para variável S , as mudanças nas variáveis (S_0 e T_0) geram respostas em malha aberta diferenciadas de acordo com as perturbações impostas ao processo fermentativo.

2.3.3.2 Problema do Tipo Servo

Para problemas do tipo servo os desvios nas variáveis, concentração de substrato (S), concentração de produto (P) e concentração de microrganismos (X) no fermentador estão apresentados na Tabela 2.4.

Tabela 2.4 Desvios nas Referências do Processo Fermentativo

<i>Tempo (h)</i>	<i>S</i>	<i>P</i>	<i>X</i>
0-50	10,0	45,0	37,0
50-100	3,0	40,0	32,0
100-150	8,0	43,0	36,0
150-200	5,5	42,0	34,8

CONCLUSÕES

Neste capítulo foram apresentadas as vantagens e aspectos importantes sobre o uso de etanol como combustível, especialmente foi apresentado o processo extrativo de fermentação alcoólica contínua, que será aplicado no desenvolvimento deste trabalho.

Pode-se observar através da dinâmica do processo, a complexidade inerente ao metabolismo dos microrganismos, o que dificulta o controle e principalmente justifica o uso de modelos não-lineares para representação do processo.

São apresentados também os tipos de problemas que serão aplicados no processo para avaliar os algoritmos de controle. Sendo estes problemas do tipo servo (para as variáveis S, P e X), regulador (para as variáveis do tipo carga T_0 e S_0) e problemas dos tipos, servo e regulador de forma simultânea.

REDES NEURAIS ARTIFICIAIS

O capítulo 3 apresenta uma introdução sobre redes neurais artificiais (RNA's) e sua aplicação na área de controle de processos. O objetivo principal deste capítulo é apresentar uma base teórica necessária para o entendimento sobre RNA's além de definir a topologia e o algoritmo de aprendizagem das redes que serão utilizadas nos algoritmos de controle que utilizam redes (NPC Neural e Condeg Modificado).

3.1 HISTÓRIA DA ARTE

As *RNA's* são apresentadas como modelos matemáticos cujas características gerais em termos estruturais (neurônios artificiais interligados entre si formando uma rede) e de comportamento (aprendizagem, reconhecimento de características relevantes em ambientes com ruído, operações em paralelo) guardam uma vaga similaridade com o cérebro humano. Dessa maneira, as reais potencialidades das *RNA's* podem ser evidenciadas, ao mesmo tempo em que expectativas errôneas são refutadas (De Souza Jr, 1993).

A operação de uma rede neural artificial é constituída por três etapas:

- Etapa de Aprendizagem – Ajuste dos pesos sinápticos;
- Etapa de Representação – Estimativa para valores de saída;
- Etapa de Generalização – Resposta para novos padrões, não apresentados na etapa de aprendizagem.

O trabalho pioneiro sobre redes neurais artificiais foi realizado no ano de 1943, por McCulloch e Pitts. Eles desenvolveram um estudo sobre o comportamento do neurônio biológico, com o objetivo de criar um modelo matemático para este. As conclusões de suas pesquisas geraram um modelo de elemento de rede neurológica e seu mecanismo de cálculo que foram uma inspiração para diversos trabalhos posteriores (McCulloch e Pitts, 1943).

Em 1949, Donald Hebb, no livro *The Organization of Behavior* (A Organização do Comportamento), apresentou pela primeira vez uma explicação filosófica para o aprendizado através da modificação da junção sináptica dos neurônios.

O primeiro modelo de rede neural implementado foi o *Perceptron*, por Frank Rosenblatt em 1958. Em seu livro *Principles of Neurodynamics* mostra o modelo dos *Perceptrons*, onde os neurônios eram organizados em camada de entrada e saída, onde os pesos das conexões adaptados a fim de se atingir a eficiência sináptica. Este trabalho estabeleceu a base para os algoritmos de treinamento de modelos não supervisionados, como o de *Kohonen*, e para modelos supervisionados como o *backpropagation*.

Neste período também surgiu o primeiro neurocomputador a obter sucesso (Mark I *Perceptron*), criado por Rosenblatt, Charles Wightman e outros (Eberhart e Dobbins, 1990).

Vários trabalhos foram desenvolvidos durante este período; parecia que as *RNA's* (Redes Neurais Artificiais) poderiam fazer qualquer coisa, e este entusiasmo exagerado de muitos pesquisadores, que imaginavam máquinas tão poderosas quanto ao cérebro humano tirou quase toda credibilidade dos estudos desta área e causou grande aborrecimento a teóricos de outras áreas. Em 1969, Minsky e Papert escreveram o livro *Perceptrons* fazendo uma crítica ao modelo e demonstrando matematicamente as suas limitações. Após a publicação desta obra iniciaram-se os anos negros das *RNA's*.

O renascimento da área ocorreu na década de 80, com o trabalho de John Hopfield, que criou o modelo de Hopfield em 1982, caracterizado pelo tipo *feedback*, isto é, existe uma conexão das entradas com as saídas. A rede de Hopfield pode ser comparada a um modelo físico, onde, a cada troca de estado da rede a energia da mesma diminui. Portanto, a fase de aprendizagem chegará ao fim no momento em que a rede tiver sua energia minimizada (Hopfield, 1982).

Em 1986 foi publicado o livro *Parallel Distributed Processing* de David Rumelhart e James McClelland, que trouxe grandes contribuições para área. Também nesta época foi

desenvolvido o algoritmo de aprendizado para redes de múltiplas camadas denominado *Backpropagation* (Rumelhart e McClelland, 1986).

A partir destes trabalhos, as *RNA's* tiveram uma grande redescoberta no mundo científico, sendo hoje considerada multidisciplinar, pois existem pesquisas nas áreas de engenharia, matemática, administração, biomédica entre outras. É importante reconhecer as capacidades e limitações de cada modelo e não criar expectativas alarmantes como no passado.

Hecht-Nielsen (1990) destaca as aplicações de *RNA's* aos seguintes casos:

- Processamento de sensores, tais como reconhecimento de padrões (por exemplo, reconhecimento de letras impressas ou manuscritas) e compreensão de imagens;
- Extração de informações a partir de dados brutos;
- Controle de processos.

Na engenharia química são muito utilizadas devido à natureza não-linear dos processos químicos – são encontradas nos três domínios citados, e muitas vezes essas categorias se sobrepõem, como no caso em que sinais de sensores são processados para uso em controle.

No controle de processos aplicando *RNA's* pode-se citar Hoskins e Himmelblau (1988), Hunt *et al.* (1992), Psychogios e Ungar (1991), de Souza Jr (1993), Liew *et al.* (1996), Hussain (1999), Ramirez-Beltran e Jackson (1999), Broderick *et al.* (1999), Hang *et al.* (1999), Narendra e Chen (2001), Ender (2002), Meleiro (2002), Duarte (2004), Duarte *et al.* (2004a, 2004b, 2006, 2007), Gadkar *et al.* (2005), Kadi (2006), Darsono e Labadie (2006), Furferi *et al.* (2007), Hung e Chung (2007), Yoo *et al.* (2007), Uraikul *et al.* (2007), Al-Alawi *et al.* (2007) e Jillson e Ydstie (2007).

3.1.1 Neurônio Artificial

Da mesma maneira que as redes neurais biológicas, as redes neurais artificiais são organizadas na forma de elementos individuais simples, que são os neurônios artificiais. Fazendo uma analogia com o neurônio biológico, o conjunto de dados de entrada (X_1, X_2, \dots, X_p) correspondem aos dendritos e o conjunto de dados de saída (y) correspondem ao axônio (Figura 3.1).

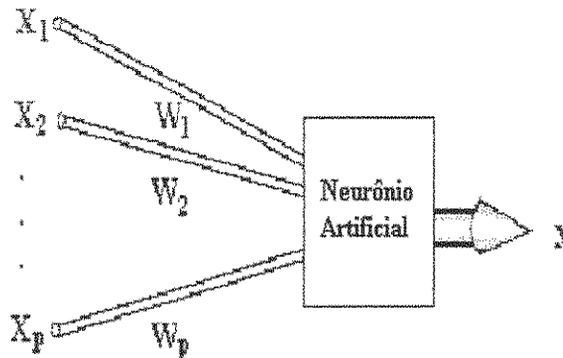


Figura 3.1 Neurônio Artificial

Os neurônios se interconectam uns aos outros através das sinapses formando as redes neurais artificiais (Figura 3.2). São os pesos das sinapses que determinam a magnitude do efeito do sinal recebido. Este sinal pode ser excitatório ou inibitório.

As redes neurais artificiais têm a capacidade de auto-organização, ou plasticidade, ou seja, através de um processo de aprendizado, é possível alterar os pesos das sinapses entre os neurônios artificiais como ocorre nos neurônios biológicos. Os pesos são os detentores da maior parte das informações que uma rede neural pode acumular. Isto implica que a fase de ajuste (aprendizagem) destes pesos tem uma influência direta na performance da rede.

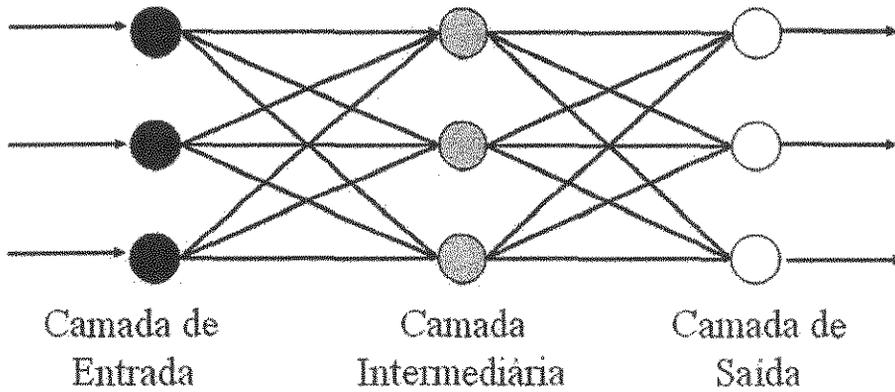


Figura 3.2 Rede Neural Artificial (Tipo Feedforward)

Considerando que um neurônio “i” qualquer da camada “k”, recebe um conjunto de informações $X_j^{(k-1)}$ ($j=1, \dots, n_{k-1}$), correspondentes as saídas dos n_{k-1} neurônios da camada anterior, ponderadas, cada uma, pelo correspondente peso da conexão $w_{i,j}^{(k)}$. O neurônio soma essas entradas ponderadas e o valor resultante é por sua vez somado a um limite interno de ativação, um “bias” que pode ser descrito por $w_{i,0}^{(k)}$. A este sinal resultante, o neurônio “i”, produz uma resposta $X_i^{(k)}$, de acordo com uma função limite ou função de ativação $f(\bullet)$, conforme mostrado na Figura 3.3, matematicamente pode-se expressar por:

$$S_i^{(k)} = w_{i,0}^{(k)} + \sum_{j=1}^{(n_{k-1})} w_{i,j}^{(k)} X_j^{(k-1)} \quad (3.1)$$

$$X_i^{(k)} = f(S_i^{(k)}) \quad (3.2)$$

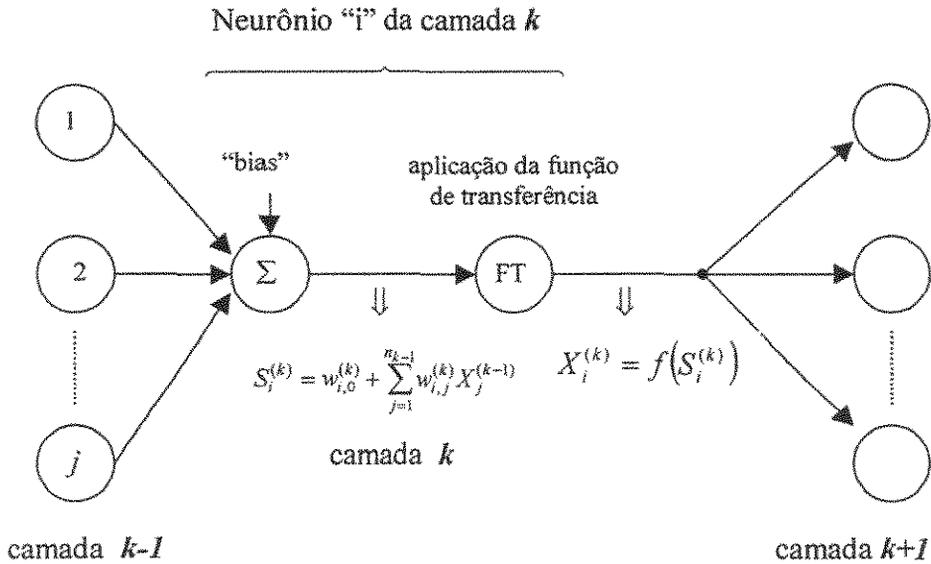


Figura 3.3 Anatomia do i-ésimo neurônio artificial da k-ésima camada

De acordo com Bakshi e Stephanopoulos (1993), para a ativação dos neurônios pode-se utilizar dois tipos de funções, a local e a global. A função de ativação local é aquela que só apresenta uma resposta ativa para valores de entrada na vizinhança imediata do seu centro de ativação. Um exemplo é a função gaussiana. A função de ativação global produz uma saída ativa para um largo intervalo de entradas. Por exemplo, as funções: linear, degrau limiar, sigmoidal e tangente hiperbólica.

Funções de ativação lineares praticamente não são utilizadas, pois as redes basicamente são usadas para modelagem de comportamentos não-lineares e o uso de camadas internas com neurônios com ativação linear não aumenta o poder computacional da rede multicamada comparando-a com uma rede sem camadas internas (Rumelhart e McClelland, 1986).

Na literatura observou-se o uso das funções sigmoidal e tangente hiperbólica com resultados satisfatórios utilizando o método de aprendizagem *backpropagation* entre eles, Duarte *et al.* (2004), Ender e Maciel Filho (2001a), Kaufmann e Andersson (2000), Moreno (2001). Nesta tese serão utilizadas como funções de ativação as sigmoidal e tangente hiperbólica. A Figura 3.4 mostra o comportamento das funções respectivas.

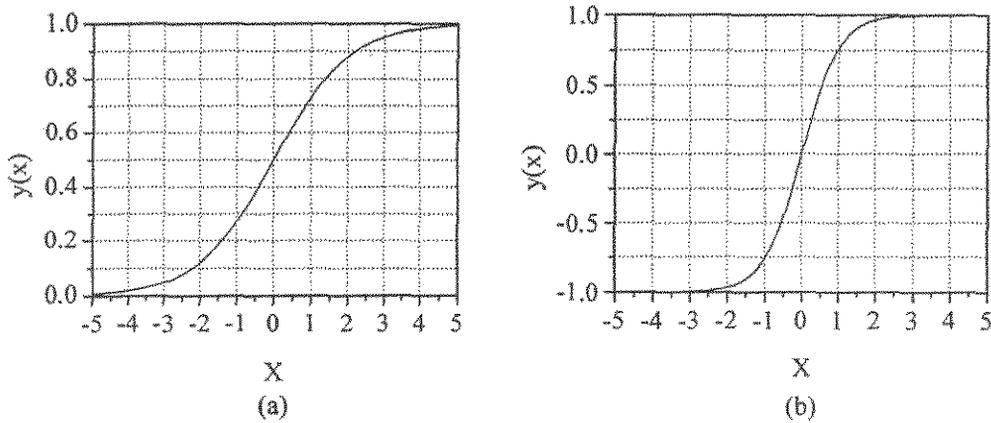


Figura 3.4 Funções de ativação: (a) Função Sigmoidal; (b) Função Tangente Hiperbólica

A diferença básica entre a função sigmoidal e a função tangente hiperbólica, é que a primeira é “positiva” (saída total variando de 0 a 1) e a segunda é “média-zero” (saída total variando de -1 a 1) (Hunt *et al.*, 1992).

As funções seguem as equações (3.3) e (3.4):

- função sigmoidal:
$$f(S_i^{(k)}) = \frac{1}{1 + e^{-S_i^{(k)}}} \quad (3.3)$$

- função tangente hiperbólica:
$$f(S_i^{(k)}) = \tanh(S_i^{(k)}) \quad (3.4)$$

3.2 ARQUITETURA DAS REDES NEURAS ARTIFICIAIS

Uma rede neural artificial é formada por diversas camadas como visto na Figura 3.2. Basicamente apresenta uma camada de entrada, camadas intermediárias (ocultas) e camada de saída. Os neurônios da camada de entrada são responsáveis pela distribuição dos dados de entrada para a próxima camada (camada intermediária). Nas camadas intermediárias as informações são processadas e propagadas até a última camada (camada de saída).

Segundo Henrique *et al.* (2000) dois aspectos importantes a serem determinados são os números de camadas intermediárias e os números de neurônios das camadas intermediárias a serem utilizados. Por não existirem padrões, estes valores variam de acordo com a função que se deseja determinar. O que se observa é que aumentando a estrutura da rede (número de camadas e neurônios) ocorre uma maior precisão dos resultados; neste caso ocorre um aumento do tempo computacional requerido.

Segundo Hecht-Nielsen (1989), uma função contínua pode ser aproximada para qualquer grau de precisão usando uma rede neural *backpropagation* com três camadas, desde que haja um número suficiente de neurônios ativos na camada oculta.

Cybenko (1989) mostrou que qualquer representação de um conjunto de dados de \mathbb{R}^p para \mathbb{R}^q pode ser obtido com duas camadas internas. Hornik *et al.* (1990) também mostraram que qualquer mapeamento pode ser obtido, para um grau de precisão, usando uma rede neural com uma camada interna com um número suficiente de neurônios.

Segundo Hecht-Nielsen (1989), o matemático russo Kolmogorov (1957), publicou um teorema relativo a representação de funções contínuas que pode ser diretamente adaptado às redes neurais multicamadas, reformulando este teorema da seguinte maneira (Kovács, 1996):

Teorema – Kolmogorov-Nielsen: Dada uma função contínua arbitrária $f: [0,1]^n \rightarrow \mathbb{R}^m$, $f(x) = y$, existe sempre para f , uma implementação exata com uma rede neural de três camadas, sendo a camada de entrada um vetor de dimensão n , a camada oculta composta por $(2n+1)$ neurônios, e a camada de saída com m neurônios representando as m componentes do vetor y .

De acordo com Loesch *et al.* (1996), vários autores sugerem que em redes de 3 camadas com mais entradas que saídas o número ideal de elementos na camada oculta seja algo em torno da média geométrica entre as camadas de entrada e de saída.

Para o desenvolvimento deste trabalho, de acordo com estudos prévios desenvolvidos por Duarte (2004), foram utilizadas redes com três camadas (uma camada de

entrada, uma intermediária e uma de saída), com três passados, sendo a camada intermediária composta por 10 neurônios. Para redes que representam o modelo do processo foram utilizadas funções de ativação sigmoideal e para redes que atuam como controlador a função tangente hiperbólica.

3.3 CLASSIFICAÇÃO DAS REDES

3.3.1 Aplicações na Área de Controle

Baughman e Liu (1995) classificaram as redes em três categorias na aplicação em controle:

- Controle Direto: A rede neural é treinada para atuar como controlador e as ações de controle são determinadas diretamente;
- Controle Inverso: A rede neural é treinada como modelo inverso de processo, predizendo a ação de controle necessária para produzir a resposta desejada do processo;
- Controle Indireto: A rede neural é treinada para atuar como modelo do processo, ou seja, para determinar o *setpoint* para um controlador local, assim afetando indiretamente o processo controlado.

Psichogios e Ungar (1991) classificaram a aplicação das redes neurais como método direto e método indireto. No método direto, a rede neural é treinada com dados de entrada / saída do sistema para representar a dinâmica inversa deste processo. Ou seja, dado o estado atual do sistema dinâmico e o estado desejado (*setpoint*) para o próximo tempo de amostragem, a rede neural é treinada a produzir a ação de controle que leva o sistema para o estado desejado. O modelo inverso resultante poderá ser usado como controlador em uma estrutura *feedforward* típica. No método indireto, a rede neural é treinada com dados do sistema dinâmico para representar dinâmicas futuras, ou seja, conhecida o estado atual e a ação de controle atual, a rede aprende a predizer o novo estado do sistema.

3.3.2 Fluxo de Informações

Quanto ao fluxo das informações as *RNA's* podem ser da forma acíclica (*Feedforward*) ou da forma cíclica (*Feedback*) (Baughman e Liu, 1995). As conexões do tipo *feedback* são usualmente utilizadas em rede neurais recorrentes, como exemplo a rede de Hopfield (Narendra e Parthasarathy, 1990). A Figura 3.5 traz exemplos de estruturas *feedforward* (a, b, c) e *feedback* (d) (Fonte lanet, 2003).

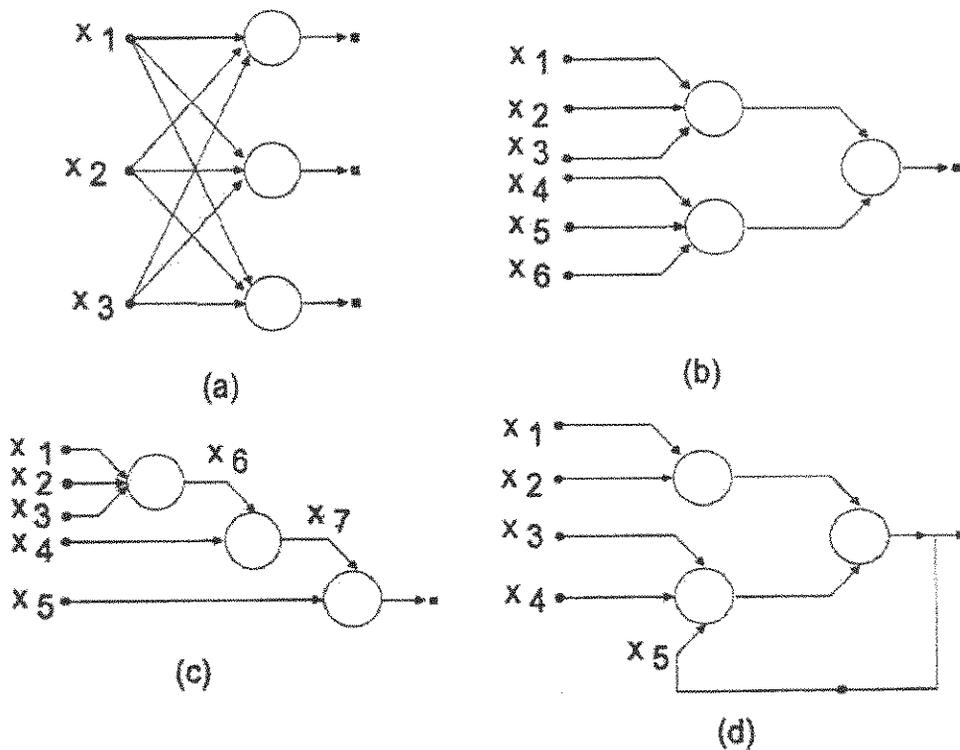


Figura 3.5 Estruturas de Rede Feedforward (a,b,c) e Feedback (d,e)

Segundo Bulsari (1995) nas redes com conexões do tipo *feedforward*, todos os sinais se propagam através das camadas para “frente”. Não há conexões laterais ou como o sinal retroceder. Este tipo de rede possui a capacidade de generalização, habilidade para aproximar funções complexas, dados com ruídos e são apropriadas para a modelagem dinâmica, onde se deseja mapear uma resposta de saída baseada em um sinal de entrada. As

redes usadas nos algoritmos de controle propostos são constituídas por este tipo de arquitetura.

3.3.3 Método de Identificação

Existem duas classificações de acordo com o método de identificação:

Paralelo: resposta predita pela rede é realimentada na mesma;

Série-Paralelo: saída do processo realimenta a rede.

3.4 PROCESSO DE APRENDIZAGEM

A aprendizagem ou treinamento é o processo onde os pesos e *biases* da rede são modificados (Simpson, 1990). O processo de aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas.

3.4.1 Aprendizado Supervisionado

Esta forma de aprendizagem utiliza um agente externo que indica à rede um comportamento bom ou ruim de acordo com o padrão de entrada, como se existisse um professor, fornecendo a resposta correta para a rede a ser treinada. A rede aprende por experiência, ou seja, um conjunto de pares de exemplos de entrada e saída são apresentados à mesma, assim obtém-se um sinal de erro com a resposta atual da rede e a saída desejada, com o qual ela pode adaptar seus pesos.

3.4.2 Aprendizado Não Supervisionado

Aprendizado não supervisionado ocorre de um modo auto-organizado não sendo necessário um professor para ordenar as mudanças sinápticas. Os parâmetros da rede são dirigidos pelos dados de entrada, de forma a criar classificações de acordo com os atributos percebidos. Essas classes são funções da distribuição estatística dos dados de entrada. Uma vez que a rede aprendeu os padrões estatísticos e formou representações internas dos atributos da entrada, ela é capaz de criar novas classes automaticamente (Braga *et al.*, 2000).

Os métodos de treinamento mais empregados em aplicações como: regressores, inferência e modelo, são os supervisionados, pois um modelo interno do processo que rege os dados fornecidos à rede é criado pela mesma, de modo que ela pode fazer previsões para novas entradas e reconhecer padrões, mesmo com ruídos (Rumelhart e McClelland (1986), Giarratano *et al.* (1990), De Souza Jr (1993)).

3.5 ALGORITMO DE APRENDIZAGEM *BACKPROPAGATION*

As redes treinadas pelo método *backpropagation* (retropropagação) são multicamadas e não apresentam conexões laterais entre neurônios da mesma camada, sendo que seus sinais são do tipo *feedforward* (Rumelhart e McClelland, 1986).

Durante o treinamento com o algoritmo *backpropagation*, a rede opera em uma seqüência de dois passos. No primeiro passo (Figura 3.6), um padrão é apresentado à camada de entrada da rede. A atividade resultante flui através da rede, camada por camada, até que a resposta seja produzida pela camada de saída. No segundo passo, a saída obtida é comparada à saída desejada para esse padrão particular. Se o valor da saída calculada não estiver correto, o erro é calculado. Este erro é então propagado a partir da camada de saída até a camada de entrada (daí o nome *backpropagation*), e os pesos das conexões dos neurônios da camada interna vão sendo modificados conforme o erro é retropropagado.

Este tipo de algoritmo utiliza a regra delta generalizada; esta usa em essência a regra da cadeia do cálculo diferencial. Seu objetivo é reduzir o erro quadrático da rede, empregando o método do passo na direção oposta do gradiente, sobre a superfície de erros, onde se deseja minimizar um erro que é função de todos os pesos da rede (Gozdziejewski Jr. e Ender, 1998).

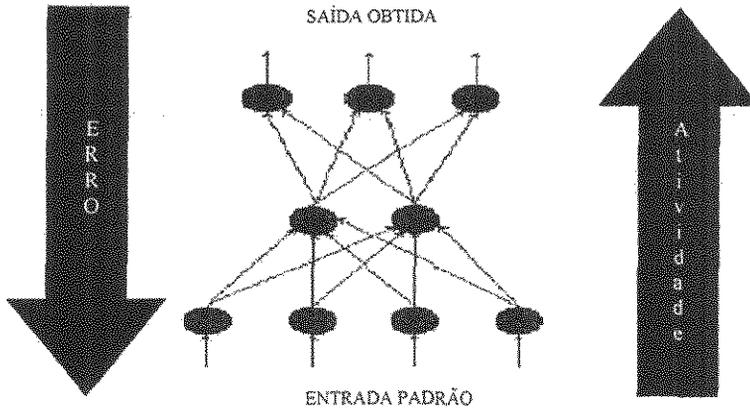


Figura 3.6 Algoritmo Backpropagation

O erro quadrático calculado na rede é dado por:

$$\varepsilon = \sum_{l=1}^{n_k} (d_l - X_l^{(k)})^2 \tag{3.5}$$

sendo que n_k o número de neurônios da camada de saída, $X_l^{(k)}$ a l -ésima saída da rede e d_l a saída desejada.

Considerando um neurônio “ i ” qualquer da camada de saída “ k ”, em uma determinada iteração. O gradiente do erro para este elemento de processamento é:

$$\vec{\nabla}_{i,j}^{(k)} \varepsilon = \frac{\partial \varepsilon}{\partial w_{i,j}^{(k)}} \tag{3.6}$$

onde $\vec{\nabla}_{i,j}^{(k)} \varepsilon$ denota o gradiente do erro quadrático do i -ésimo neurônio da k -ésima camada gerado pela j -ésima conexão sináptica.

Sabe-se que:

$$X_i^{(k)} = f(S_i^{(k)}) \tag{3.7}$$

$$S_i^{(k)} = \sum_{j=0}^{n_{k-1}} w_{i,j}^{(k)} X_j^{(k-1)} \quad (3.8)$$

sendo que para $j = 0$, $X_j^{(k-1)} = +1$ que representa a conexão extra (*bias*) e n_{k-1} é o número de neurônios da camada $k - 1$. Então:

$$\varepsilon = \sum_{l=1}^{n_k} \left(d_l - f(S_l^{(k)}) \right)^2 \quad (3.9)$$

$$\frac{\partial \varepsilon}{\partial w_{i,j}^{(k)}} = \frac{\partial \left(\sum_{l=1}^{n_k} \left(d_l - f(S_l^{(k)}) \right)^2 \right)}{\partial w_{i,j}^{(k)}} \quad (3.10)$$

Sabe-se que quando $i \neq l$ as derivadas serão nulas, portanto:

$$\frac{\partial \varepsilon}{\partial w_{i,j}^{(k)}} = -2 \left(d_i - f(S_i^{(k)}) \right) \frac{\partial f(S_i^{(k)})}{\partial w_{i,j}^{(k)}} \quad (3.11)$$

Aplicando a regra da cadeia, chega-se a:

$$\frac{\partial \varepsilon}{\partial w_{i,j}^{(k)}} = -2 \left(d_i - f(S_i^{(k)}) \right) \frac{\partial f(S_i^{(k)})}{\partial S_i^{(k)}} \frac{\partial S_i^{(k)}}{\partial w_{i,j}^{(k)}} \quad (3.12)$$

Pela equação (3.8) pode-se determinar:

$$\frac{\partial S_i^{(k)}}{\partial w_{i,j}^{(k)}} = \frac{\partial \left(\sum_{l=0}^{n_{k-1}} w_{i,l}^{(k)} X_l^{(k-1)} \right)}{\partial w_{i,j}^{(k)}} \quad (3.13)$$

A derivada em questão na equação anterior é nula quando $l \neq j$, desta forma:

$$\frac{\partial S_i^{(k)}}{\partial w_{i,j}^{(k)}} = X_j^{(k-1)} \quad (3.14)$$

A equação (3.12) pode ser reescrita:

$$\frac{\partial \varepsilon}{\partial w_{i,j}^{(k)}} = -2(d_i - X_i^{(k)})f'(S_i^{(k)})X_j^{(k-1)} \quad (3.15)$$

Definindo:

$$\delta_i^{(k)} = (d_i - X_i^{(k)})f'(S_i^{(k)}) \quad (3.16)$$

Segundo Duarte (2004), Ender (2002), Nguyen e Widrow (1990) e Chen (1990):

$\delta_i^{(k)}$ pode ser interpretado como *erro equivalente*, o qual será igual ao erro $d_i - X_i^{(k)}$ se $f(S_i^{(k)})$ for a função identidade. Substituindo tem-se:

$$\vec{\nabla}_{i,j}^{(k)} \varepsilon = \frac{\partial \varepsilon}{\partial w_{i,j}^{(k)}} = -2\delta_i^{(k)}X_j^{(k-1)} \quad (3.17)$$

Desta forma, os pesos podem ser reajustados na direção oposta ao gradiente do erro, da seguinte forma:

$$w_{i,j}^{(k) \text{ new}} = w_{i,j}^{(k) \text{ old}} + \eta \left(-\vec{\nabla}_{i,j}^{(k)} \varepsilon \right) \quad (3.18)$$

Ou substituindo a equação (3.17) na equação (3.18), tem-se:

$$w_{i,j}^{(k) \text{ new}} = w_{i,j}^{(k) \text{ old}} + 2\eta\delta_i^{(k)}X_j^{(k-1)} \quad (3.19)$$

onde $0 < \eta \ll 1$ é o termo de taxa de aprendizagem.

A forma de determinação de $\delta_i^{(k)}$, apresentada na equação (3.16), é válida para o ajuste dos pesos dos neurônios da camada de saída. Para ajustar os pesos das camadas ocultas, o erro ε deverá ser retropropagado até a camada oculta em análise, conforme exposto a seguir.

Considerando um neurônio “ i ” de uma camada oculta “ k ”, em uma rede neural de 3 camadas, ou seja, uma camada de entrada ($k-1$), uma camada oculta (k) e uma camada de saída ($k+1$). Pela regra da cadeia, pode-se escrever:

$$\frac{\partial \varepsilon}{\partial w_{i,j}^{(k)}} = \frac{\partial \varepsilon}{\partial S_i^{(k)}} \frac{\partial S_i^{(k)}}{\partial w_{i,j}^{(k)}} \quad (3.20)$$

Analisando primeiramente o termo $\frac{\partial \varepsilon}{\partial S_i^{(k)}}$,

$$\frac{\partial \varepsilon}{\partial S_i^{(k)}} = \frac{\partial \left(\sum_{l=1}^{n_{k+1}} (d_l - X_l^{(k+1)})^2 \right)}{\partial S_i^{(k)}} \quad (3.21)$$

$$\frac{\partial \varepsilon}{\partial S_i^{(k)}} = -2 \sum_{l=1}^{n_{k+1}} (d_l - f(S_l^{(k+1)})) \frac{\partial f(S_l^{(k+1)})}{\partial S_l^{(k+1)}} \frac{\partial S_l^{(k+1)}}{\partial S_i^{(k)}} \quad (3.22)$$

Porém:

$$\frac{\partial S_l^{(k+1)}}{\partial S_i^{(k)}} = \frac{\partial \left(\sum_{j=0}^{n_k} w_{l,j}^{(k+1)} f(S_j^{(k)}) \right)}{\partial S_i^{(k)}} \quad (3.23)$$

Sabe-se que as derivadas serão nulas para $j \neq i$, portanto:

$$\frac{\partial S_l^{(k+1)}}{\partial S_i^{(k)}} = w_{l,i}^{(k+1)} f'(S_i^{(k)}) \quad (3.24)$$

Então:

$$\frac{\partial \varepsilon}{\partial S_i^{(k)}} = -2 \sum_{l=1}^{n_{k+1}} (d_l - X_l^{(k+1)}) f'(S_l^{(k+1)}) w_{l,i}^{(k+1)} f'(S_i^{(k)}) \quad (3.25)$$

Pela equação (3.16), pode-se afirmar:

$$\delta_i^{(k+1)} = (d_i - X_i^{(k+1)}) f'(S_i^{(k+1)}) \quad (3.26)$$

Portanto:

$$\frac{\partial \varepsilon}{\partial S_i^{(k)}} = -2 f'(S_i^{(k)}) \sum_{l=1}^{n_{k+1}} \delta_l^{(k+1)} w_{l,i}^{(k+1)} \quad (3.27)$$

Analisando agora o termo $\frac{\partial S_i^{(k)}}{\partial w_{i,j}^{(k)}}$,

$$\frac{\partial S_i^{(k)}}{\partial w_{i,j}^{(k)}} = \frac{\partial \left(\sum_{m=1}^{n_{k-1}} w_{i,m}^{(k)} X_m^{(k-1)} \right)}{\partial w_{i,j}^{(k)}} \quad (3.28)$$

As derivadas para $m \neq j$ são nulas, então:

$$\frac{\partial S_i^{(k)}}{\partial w_{i,j}^{(k)}} = X_j^{(k-1)} \quad (3.29)$$

Portanto a derivada do erro em relação aos pesos da camada oculta pode ser determinada da seguinte forma:

$$\frac{\partial \varepsilon}{\partial w_{i,j}^{(k)}} = -2 f'(S_i^{(k)}) \sum_{l=1}^{n_{k+1}} (\delta_l^{(k+1)} w_{l,j}^{(k+1)}) X_j^{(k-1)} \quad (3.30)$$

Então:

$$\frac{\partial \varepsilon}{\partial w_{i,j}^{(k)}} = -2 \delta_i^{(k)} X_j^{(k-1)} \quad (3.31)$$

Onde se tem para a camada intermediária:

$$\delta_i^{(k)} = f'(S_i^{(k)}) \sum_{l=1}^{n_{k+1}} (\delta_l^{(k+1)} w_{lj}^{(k+1)}) \quad (3.32)$$

Esta forma para o cálculo de $\delta_i^{(k)}$ é usada para retropropagar o erro para as demais camadas ocultas, caso a rede neural apresentar $n > 1$ camadas ocultas. A equação para o ajuste dos pesos fica da mesma forma que apresentada na equação (3.19), ou seja:

$$w_{i,j}^{(k) \text{ new}} = w_{i,j}^{(k) \text{ old}} + 2\eta \delta_i^{(k)} X_j^{(k-1)} \quad (3.33)$$

A inicialização dos pesos das interconexões e os *biases* são realizados com números aleatórios e pequenos. Desta forma, procura-se evitar os problemas de simetria (que haveria caso se começasse com pesos iguais) e saturação (produzidos por pesos grandes). No desenvolvimento desta tese, adotaram-se pesos iniciais aleatórios no intervalo entre -1 e 1 , conforme ENDER (2002), Rumelhart e McClelland (1986).

Segundo Ender (2002), para cada padrão p ($p=1, 2, \dots, N_p$; onde N_p é o número de padrões), tem-se um conjunto de entradas e um conjunto de saídas desejadas, d_i , que são usadas no critério de erro. Na implementação da aprendizagem *backpropagation*, pode-se atualizar os pesos após a apresentação de cada padrão (aprendizagem por padrão) ou após a apresentação de todos os padrões (aprendizagem batelada) (Rumelhart e McClelland (1986). A forma de implementação adotada foi a aprendizagem por padrão.

Depois do treinamento, pode-se usar a rede, onde se tem apenas a fase em que as entradas passam através da rede para frente (*feedforward*), correspondendo às equações (3.1) e (3.2).

Para o desenvolvimento desta tese, adotou-se o controle direto, quando a rede neural atua como controlador e controle indireto, quando as redes neurais são usadas em modelos de predição de estados ou dinâmicas futuras. A seguir serão apresentadas redes atuando como controlar neural e como modelo direto do processo.

3.6 CONTROLADOR NEURAL – DINÂMICA INVERSA

Um processo pode ser descrito na seguinte forma discreta (Narendra e Parthasarathy (1990), Chen e Billings (1992), Nahas *et al.* (1992), SU *et al.* (1992), Bulsari (1995), Ender (2002) e Duarte (2004)):

$$Y(k+1) = f(Y(k), \dots, Y(k-n+1), U(k), \dots, U(k-m+1)) \quad (3.34)$$

com:

$$\begin{aligned} Y(k) &= [y(1, k), \dots, y(i, k)]^T \\ U(k) &= [u(1, k), \dots, u(i, k)]^T \end{aligned} \quad i = \text{número de malhas de controle} \quad (3.35)$$

onde $U(k)$ e $Y(k)$ ($k=0, 1, \dots, n$) são os vetores de perturbações e respostas no instante de tempo k , respectivamente, e $f(\bullet)$ é uma função desconhecida. Considerando o desenvolvimento de uma função aproximada para $f(\bullet)$ pelo uso de redes neurais, onde $Y(k+1)$ e $Y(k), \dots, Y(k-n+1), U(k), \dots, U(k-m+1)$ são respostas e perturbações da rede neural respectivamente, e W são os pesos das conexões e os limiares da rede neural. Esta rede neural pode gerar um modelo do processo, pelo ajuste de W , denominado modelo de controle indireto:

$$\hat{Y}(k+1) = \hat{f}(Y(k), \dots, Y(k-n+1), U(k), \dots, U(k-m+1); W) \quad (3.36)$$

Resolvendo a equação (3.34) em relação a $U(k)$, obtém-se:

$$U(k) = \phi(Y(k+1), Y(k), \dots, Y(k-n+1), U(k-1), \dots, U(k-m+1)) \quad (3.37)$$

onde $\phi(\bullet)$ é uma função desconhecida, que representa a dinâmica inversa do processo. Como a função $\phi(\bullet)$ é desconhecida, bem como a função $f(\bullet)$, não é possível avaliar a equação (3.37). Neste caso, utiliza-se uma rede neural para gerar uma aproximação da função ϕ , da seguinte maneira:

$$\hat{U}(k) = \hat{\phi}(Y(k+1), Y(k), \dots, Y(k-n+1), U(k-1), \dots, U(k-m+1); W) \quad (3.38)$$

Considerando-se um problema de controle em que a resposta $Y(k)$ procura acompanhar as referências da malha fechada, $Y_r(k)$. Então para o próximo tempo de amostragem:

$$Y(k+1) = Y_r(k+1) \quad (3.39)$$

Substituindo a Equação (3.39) na Equação (3.38), a ação de controle desejada para o processo é dada por:

$$\hat{U}(k) = \hat{\phi}(Y_r(k+1), Y(k), \dots, Y(k-n+1), U(k-1), \dots, U(k-m+1); W) \quad (3.40)$$

Este modelo representa o mapeamento de perturbações/respostas não-lineares através de redes neurais, denominado de modelo de controle direto, que terá suficiente exatidão se um grande número de dados estiver disponível (Psychogios e Ungar (1991), Bulsari, (1995)).

Conforme a Figura 3.7, as entradas desta rede são formadas pelas referências das variáveis manipuladas para o próximo tempo de amostragem das variáveis manipuladas e controladas passadas. Como resposta da rede obtém-se as ações de controle para o próximo tempo de amostragem, onde β representa o número de informações passadas do processo usadas como entrada na rede.

O controle neural baseado em modelo inverso tem como objetivo conduzir a variável controlada para a referência desejada do processo. Segundo Ender (2002), este tipo de modelo não apresenta resultados eficientes para comportamentos dinâmicos, podendo gerar *off-set*.

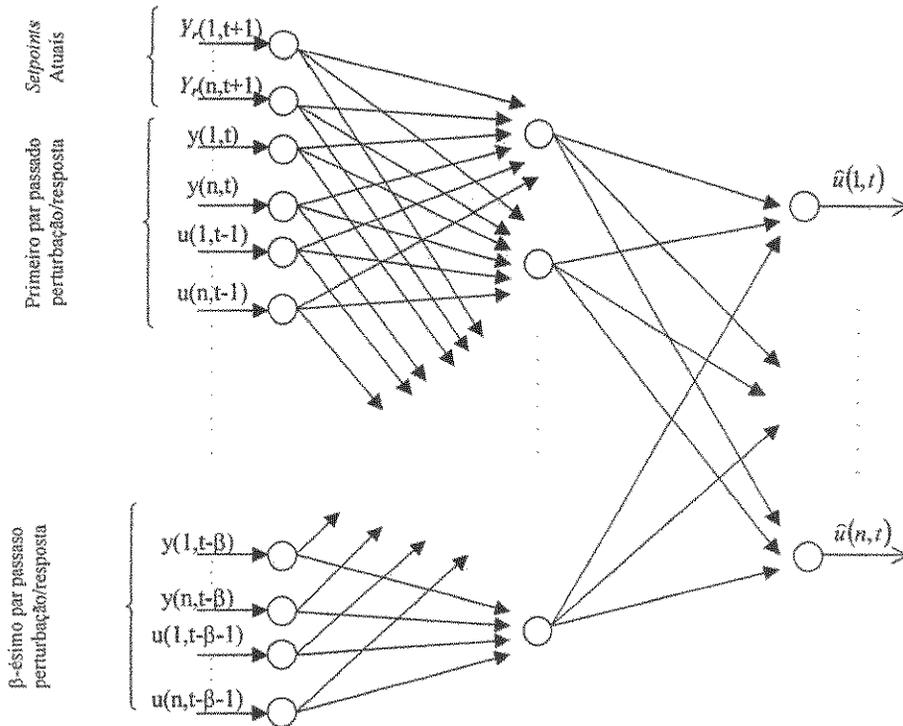


Figura 3.7 Arquitetura feedforward multivariável (Modelo Inverso)

3.7 MODELO NEURAL DO PROCESSO – IDENTIFICAÇÃO

As redes neurais artificiais são amplamente empregadas na identificação de processos, gerando modelos a serem usados no projeto de controladores não-lineares. No algoritmo de controle proposto, denominou-se esta rede como Modelo Neural do Processo. Utilizou-se o método de identificação série - paralelo, conforme descrito anteriormente.

De acordo com os conceitos apresentados neste capítulo, determinou-se a arquitetura das redes utilizadas no algoritmo de controle proposto com aprendizagem *on-line* das redes. Estas serão compostas por 3 camadas, estrutura *feedforward* e como funções de ativação serão utilizadas as funções sigmoidal e tangente hiperbólica. O algoritmo de aprendizagem utilizado será do tipo supervisionado, *backpropagation* apropriado para o controle de processos devido à capacidade de representar processos dinâmicos e generalizar mesmo com ruídos.

3.8 APRENDIZAGEM *ON-LINE* E *OFF-LINE* DAS *RNA'S*

Aprendizagem *off-line* é aquela realizada com um conjunto de informações passadas do processo (dados padrões), ou seja, pares de entrada e saída (Perturbações/Respostas) obtidos através de perturbações realizadas no processo. Estes dados são apresentados várias vezes à rede neural, até a representação apropriada destes padrões.

Aprendizagem *on-line* é aquela realizada com dados obtidos do processo em tempo real e estes são alimentados ao conjunto de dados de aprendizagem. Esta aprendizagem permite a rede neural “aprender” o comportamento do processo num momento de não-linearidades fortes ou mapear uma região ainda não apresentada à rede neural, durante a sua utilização.

As aprendizagens *on-line* e *off-line* são essencialmente idênticas do ponto de vista matemático, diferindo apenas na forma de obtenção dos padrões de aprendizagem. Na aprendizagem *off-line*, o conjunto de padrões é fixo e na *on-line* é flutuante.

A aprendizagem *on-line* apresenta limitações devido ao número de iterações necessárias para atender o critério de tolerância adotado e à necessidade de aprendizagem em tempo real. Para contornar estas dificuldades, limitou-se o número máximo de iterações, usando um vetor de padrões formado por um número pré-estabelecido das últimas informações passadas de perturbações/respostas do sistema em questão. Além disto, para garantir a boa representação dinâmica do processo através de redes neurais, adotou-se uma estratégia formada por três redes neurais que atuam paralelamente (Ender e Maciel Filho, 2001a,b). Essa estratégia foi utilizada para o controlador e modelo do processo, pois todas as redes foram submetidas à aprendizagem *on-line*. Esta estratégia possui uma etapa *off-line* e outra *on-line* conforme descrito a seguir.

Numa primeira etapa, as redes devem ser treinadas com os dados padrões da aprendizagem (*off-line*), conforme a Figura 3.8.

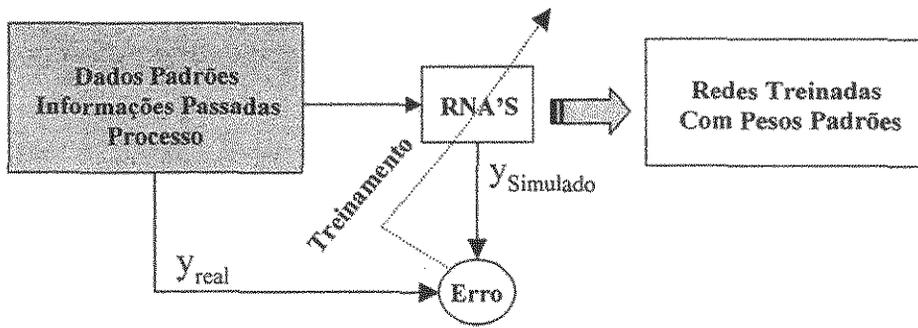


Figura 3.8 Etapa 1 - Aprendizagem Off-line (Ajuste de Pesos com Dados Padrões)

Na segunda etapa de aprendizagem, a primeira rede neural (*RNA 1*) é denominada de Padrão, pois não terá seus pesos atualizados, ou seja, seus pesos são os obtidos através da aprendizagem *off-line* (pesos padrões). A segunda rede neural (*RNA 2*) é denominada de Base-Padrão, pois inicia a aprendizagem com os pesos padrões (obtidos na primeira etapa da aprendizagem *off-line*) e tem os mesmos sendo atualizados ao longo do tempo através da aprendizagem *on-line*. Sempre que a *RNA 1* apresentar o menor erro quadrático em relação às outras redes, a *RNA 2* terá seus pesos substituídos pelos pesos padrões e inicia novamente o processo de ajuste de pesos (aprendizagem *on-line*). A terceira rede (*RNA 3*) é denominada de Nova, pois inicia o processo de aprendizagem com os pesos padrões e tem seus pesos sendo atualizados ao longo do tempo (Figura 3.9).

A diferença na representação destas redes é que a rede treinada apenas com aprendizagem *off-line* (*RNA 1*), por ter em seu vetor de treinamento um conjunto de dados de várias perturbações e maior número de dados consegue representar bem o perfil do processo; pode-se dizer que tem uma maior “memória” do processo. Já as redes treinadas com dados *on-line* (*RNA's 2 e 3*) possuem uma pequena “memória” ou seja, conseguem representar muito bem as últimas perturbações, mas não guardam um histórico do processo.

Isto pode ser considerado uma limitação se o processo apresentar uma mesma perturbação, pois a rede não conseguirá lembrar desta perturbação, não irá representar o processo se não for novamente treinada. Mas este é o objetivo da aprendizagem *on-line* permitir que a rede seja treinada para novas perturbações ao longo do tempo.

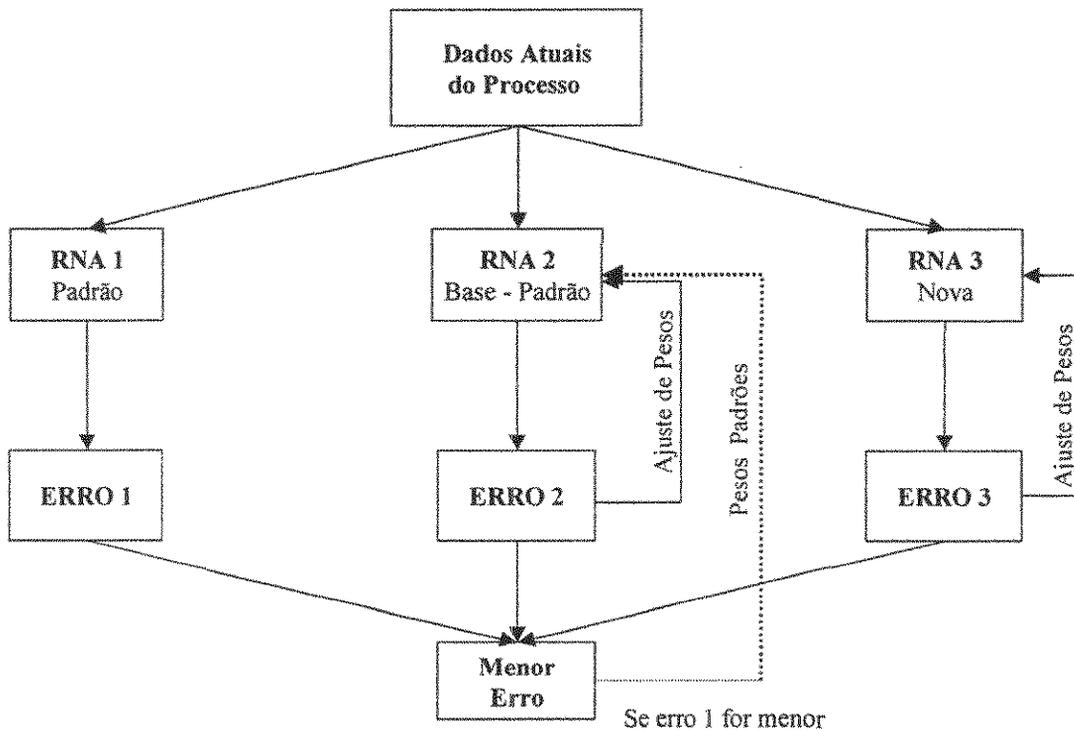


Figura 3.9 Etapa 2 – Aprendizagem On-line (Ajuste de Peso em Tempo Real)

A estratégia de aprendizagem *on-line* usando três redes neurais permite o acompanhamento das variações do processo, mesmo quando o número de iterações não for suficiente para convergência da aprendizagem *on-line*, pois utiliza-se a *RNA* padrão ou *RNA* base padrão. As três redes apresentam as seguintes características (Duarte, 2004):

- Rede Padrão (RNA 1) – Apresenta uma “memória” geral do processo, ou seja, consegue representar bem o processo como um todo, mas muitas vezes não representa com exatidão uma região específica do processo;
- Rede Base-Padrão (RNA 2) – É uma rede intermediária entre as outras redes. Permite a retomada da aprendizagem *on-line* a partir dos pesos padrões sempre que estes apresentarem melhores resultados;
- Rede Nova (RNA 3) – Apresenta uma “memória” local, ou seja, é a rede que melhor representa o processo nos últimos tempos de amostragem quando o número de iterações é suficiente. Como tem seus pesos modificados ao

longo do tempo não consegue “lembrar” de modificações impostas para um tempo passado muito grande.

Para o desenvolvimento deste trabalho, foi utilizada a configuração de aprendizagem proposta por Duarte (2004). Esta configuração considera durante a aprendizagem, o vetor teste formado por todo conjunto de dados de informações do processo, (padrões do tempo $(t-b)$ até (t)). O vetor de aprendizagem é formado pelos últimos tempos de amostragem do processo, (padrões do tempo $(t-a)$ até (t)), conforme o esquema da Figura 3.10.

Em todas as simulações realizadas com esta configuração foram utilizados 25 pares de informações do processo para fase de aprendizagem e 50 pares para fase de teste das redes, exceto quando determinado um valor diferenciado para o vetor de teste.

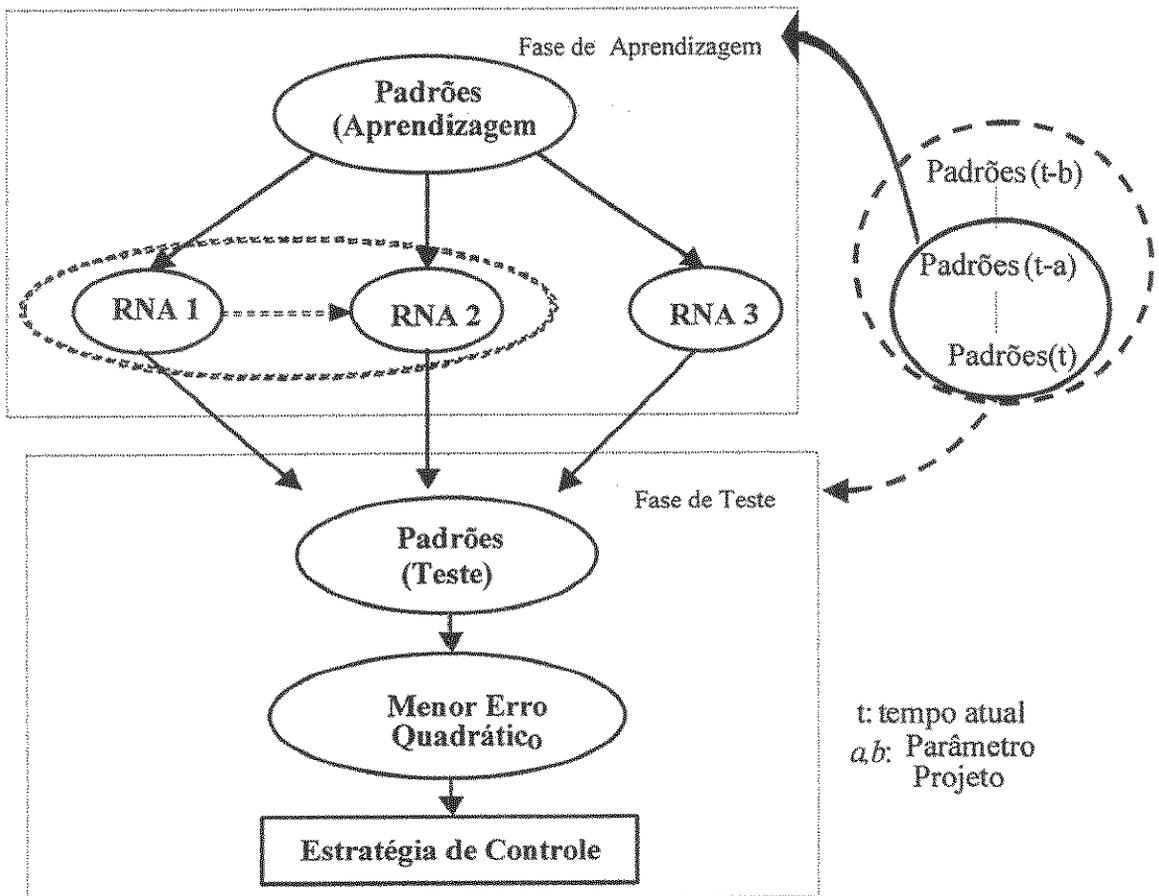


Figura 3.10 Esquema para Configuração da Aprendizagem On-line

CONCLUSÕES

Para o desenvolvimento deste trabalho, foram utilizadas redes neurais nos algoritmos Condeg Modificado e NPC Neural, para os quais elas foram utilizadas como controle direto, quando a rede neural atua como controlador, e controle indireto, quando as redes neurais são usadas em dinâmicas futuras e modelos de predição, respectivamente.

Conforme apresentado neste capítulo, todas as redes que serão estudadas apresentam arquitetura do tipo *feedforward*, com três camadas, funções de ativação sigmoidal e tangente hiperbólica. O algoritmo de aprendizagem foi do tipo supervisionado, *Backpropagation* apropriado para o controle de processos devido à capacidade de representar processos dinâmicos.

Foi apresentada a estratégia de aprendizagem *on-line*, que utiliza três redes em paralelo e que foi aplicada em todas as redes utilizadas nos algoritmos de controle, por permitirem que as redes neurais tenham uma aprendizagem ao longo do tempo.

ALGORITMOS DE CONTROLE

Este capítulo aborda o controle de processos, e o principal objetivo é apresentar os algoritmos de controle usados para o desenvolvimento deste trabalho, entre eles o DMC (controle por matriz dinâmica), MPC Neural (controle preditivo baseado em modelo neural), Condeg Modificado (controle baseado no erro global) e Condeg Modificado com filtro de Kalman.

4.1 CONTROLE DE PROCESSOS

Controlar um processo significa atuar sobre ele, ou sobre as condições a que o processo está sujeito, de modo a atingir algum objetivo, como por exemplo, manter o processo próximo de um determinado estado estacionário, mesmo que os efeitos externos tentem desviá-lo desta condição (Stepanopoulos, 1984).

O controle é realizado principalmente para:

- Garantir que restrições de segurança e ambientais não sejam violadas;
- Garantir certas condições operacionais;
- Garantir produtos finais com maior qualidade e menor custo.

A maioria das plantas químicas dificilmente opera em estado estacionário, apesar de ser uma condição de operação desejável, nem sempre é atingido ou mantido por muito tempo. Por isso existe a necessidade de monitoramento da operação destas plantas e de intervenção para garantir a satisfação dos objetivos operacionais.

Os processos químicos, bioquímicos e petroquímicos podem apresentar algumas características que dificultam o controle, dentre as quais se podem citar: a presença de não-linearidades, interação entre as variáveis controladas e manipuladas, variáveis de estado e

perturbações não mensuráveis, parâmetros incertos e variando no tempo, tempo morto e atraso nas medidas (Bequette, 1991).

Segundo Assis (2004), tradicionalmente o controle dos processos não-lineares tem sido feito aplicando a teoria de controle linear a um modelo linearizado do processo na etapa de projeto do controlador. Se o processo for operado continuamente no estado estacionário em torno do qual houve a linearização, o modelo linear obtido não difere muito do processo real. Porém, é comum que as condições operacionais do processo mudem com o tempo, dependendo de vários fatores, desde econômicos, qualidade e fonte da matéria-prima, desativação de catalisadores, mudanças genéticas em microrganismos. Nas novas condições de operação, o controle existente pode já não ser mais adequado, levando às instabilidades no processo, perda de rendimento, presença de *off sets*, dentre outras características indesejáveis. Tendo em vista tal desempenho, é natural que se tenha pensado em desenvolver uma teoria de controle que levasse em consideração as não-linearidades do processo na etapa de projeto do controlador, surgindo assim a teoria de controle não-linear.

Uma etapa importante no desenvolvimento, implementação e análise de controladores se refere à representação adequada do comportamento do sistema. Dentro deste contexto, os modelos dinâmicos de processos são ferramentas básicas para diversos propósitos, tais como, simulação, controle, predição de estados, identificação de sistemas, entre outros. Os modelos dinâmicos lineares, devido à sua versatilidade e simplicidade, têm sido muito utilizados, porém não são adequados para descrever processos com multiplicidade de estados estacionários e as não-linearidades inerentes à maioria dos processos da Engenharia Química.

Como efeito da globalização surge a necessidade de produtos com menor custo e maior rendimento, promovendo um avanço tecnológico significativo em diversas áreas. O controle e otimização de processos são ferramentas que auxiliam nessa nova concepção, trazendo muitas melhorias, como produtos com maior qualidade, menor custo e um melhor aproveitamento dos recursos.

A literatura tem mostrado que é necessário o desenvolvimento de algoritmos de controle baseados nos conceitos de controle avançado para se obter a operação do sistema a

altos níveis de desempenho, conciliado as condições seguras. Com esta crescente necessidade de se desenvolver controladores de processos com desempenho cada vez melhor, emerge a necessidade do desenvolvimento de algoritmos de controle não-lineares para atuarem de forma mais efetiva nos processos não-lineares, os quais deverão ter a capacidade de adaptação em função das perturbações externas que podem ocorrer.

Dentro deste contexto, emergiram os algoritmos de controle usando otimizações em tempo real, onde diversas ferramentas são empregadas objetivando desenvolver técnicas para o controle de processos considerados com dinâmica complexa.

As Redes Neurais Artificiais (*RNA's*) têm se tornado recentemente foco de muita atenção, principalmente devido à sua capacidade de aprendizado, à ampla gama de aplicabilidade e à facilidade com que armazena conhecimento experimental e o torna disponível para o uso em aplicações em engenharia de processos.

Muitas pesquisas foram desenvolvidas na última década na identificação e controle de processos aplicando as redes neurais artificiais ou *RNA's* por estas características e principalmente pela grande capacidade de representação de modelos não-lineares. Para maior detalhamento, alguns autores podem ser consultados, entre eles, Baugman e Liu (1995), Hussain (1999).

A utilização de *RNA's* no controle de bioprocessos também é bastante estudada, principalmente pela complexidade destes processos. Dentre estes estudos a utilização das redes pode ser utilizada para modelar ou controlar o processo ou ambos os casos. (Boskovic e Narrendra, 1995; Braake *et al.* 1998; Vilaça *et al.* 2000, Costa *et al.* 2002, Meleiro 2002, Gadkar *et al.*, 2005 e Duarte *et al.* 2007).

4.2 CONTROLE PREDITIVO BASEADO EM MODELO

O controle preditivo baseado em modelo, também conhecido como *Model Predictive Control* (MPC) fundamenta-se no uso de um modelo dinâmico do processo como parte do controlador.

De acordo com Meleiro (2002), o conceito básico deste algoritmo de controle é que um modelo dinâmico do processo e as medidas de interesse são utilizados para prever o comportamento futuro do processo. As ações de controle são obtidas minimizando uma função do erro entre a resposta predita do processo e a trajetória desejada (*setpoint*).

Segundo os autores Ogunnaike e Ray (1994), o algoritmo MPC teve um grande crescimento em suas aplicações na década de 1990 e pode ser considerado o algoritmo de controle multi-variável mais empregado nos processos industriais.

Comparado com o controle clássico, o controle preditivo apresenta vantagens como o controle de processos multivariáveis, possibilidade de se utilizar restrições e principalmente em processos com resposta inversa e tempo morto (Marchetti *et al.* 1983).

O princípio geral do controle preditivo (MPC) é apresentado na Figura 4.1 (Meleiro, 2002). No instante de amostragem k , o controlador recebe informação sobre o estado atual do sistema quando, baseado no modelo do processo, calcula as ações de controle futuras. A trajetória de entradas futuras é, então, determinada segundo um critério de otimização, e a primeira ação de controle é implementada até a próxima amostragem.

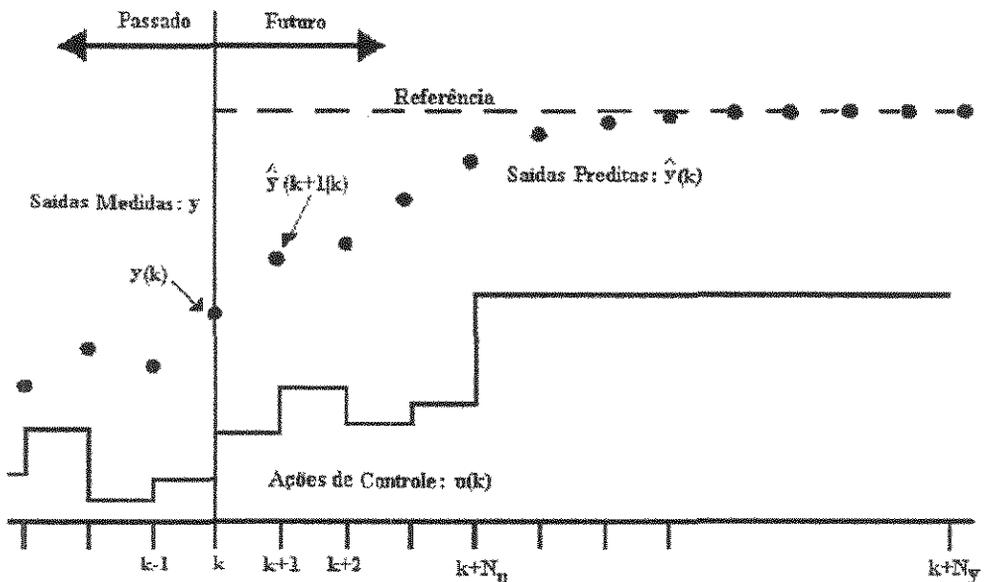


Figura 4.1 Estrutura Básica de um Controle Preditivo (Fonte: Meleiro, 2002)

O algoritmo MPC pode ser dividido em vários tipos de controle, dependendo do tipo de modelo utilizado pelo controlador, função objetivo empregada e tratamento das restrições impostas ao processo. Pode-se citar entre os diversos tipos:

- DMC – Controle por Matriz Dinâmica (Cutler e Ramaker, 1979);
- MAC – Controle Algorítmico por Modelo (Richalet *et al.*, 1978);
- GPC – Controle Preditivo Generalizado (Clarke *et al.* 1987).

Dentre os tipos de controladores baseados em modelos, o controle por matriz dinâmica (*Dynamic Matrix Control - DMC*), é um dos controladores preditivos mais utilizados na indústria química, por isso foi um dos algoritmos de controle utilizados no desenvolvimento deste trabalho.

De acordo com Meleiro (2002) o desempenho inadequado apresentado pelos controladores lineares quando usados em sistemas não-lineares ou mesmo a sistemas moderadamente não-lineares que operam em uma faixa ampla de condições é um dos principais fatores pelo interesse em projeto de controladores não-lineares. Esta abordagem é genericamente chamada de *Controle Preditivo Baseado em Modelo Não-Linear* (NMPC). Esta formulação do problema de controle é análoga à abordagem linear, exceto que a predição do comportamento futuro do processo é feita por um modelo dinâmico não-linear (Henson, 1998). A técnica predominante para modelos dinâmicos não-lineares a partir de dados entrada-saída são as redes neurais artificiais (Morari e Lee, 1999).

No desenvolvimento deste trabalho foram utilizadas duas técnicas de controle preditivo para análise do processo fermentativo, uma com modelo linear que considerou o DMC clássico e outra com modelo não-linear (NMPC) que considerou uma modificação do DMC, com uma rede neural para representar o modelo dinâmico do processo, denominada MPC Neural. Os algoritmos são descritas a seguir, nos subitens 4.2.1 e 4.2.2, respectivamente.

4.2.1 Controle por Matriz Dinâmica (DMC)

O controle por matriz dinâmica (DMC) foi originalmente desenvolvido na *Shell Oil Company*, cuja aplicação inicial data de 1973. Os autores Cutler e Ramaker (1979) apresentaram o DMC multi-variável sem restrições aplicado a um sistema de forno – pré aquecedor industrial. Em 1980, os autores Prett e Gillete descreveram uma versão multi-variável com restrições para uma unidade de craqueamento catalítico. Cutler (1982) apresentou modificações do DMC para aplicações em processos auto reguladores.

As principais características do algoritmo de controle DMC de acordo com Qin e Badgwell (1997) são: o modelo linear da planta, a minimização de uma função objetivo de desempenho quadrático sobre um horizonte de predição finito e o comportamento futuro da saída da planta especificado de modo a seguir a saída de referência o mais próximo possível.

De acordo com Meleiro (2002) o modelo utilizado pelo DMC geralmente é obtido a partir da aplicação de um degrau em cada variável de entrada do processo, seguido da correspondente medição das respostas das variáveis de saída. Estas respostas ao degrau são utilizadas para identificar o modelo de convolução do processo. Através do modelo de resposta ao degrau, podem-se prever as mudanças futuras das saídas do processo como uma combinação linear das variações nas entradas.

Em cada instante de amostragem, o valor da resposta ao degrau da variável de saída é medido e chamado de coeficiente de resposta ao degrau a_i . A diferença entre dois coeficientes de resposta ao degrau sucessivo é chamado de coeficiente de resposta ao impulso e é dado pela equação (4.1) (Newell e Lee, 1989):

$$h_i = a_i - a_{i-1} \quad (4.1)$$

O valor predito da variável de saída \hat{y} , no instante de amostragem k , pode ser expresso de acordo com a equação (4.2):

$$\hat{y}_k = \sum_{i=1}^{N_m} h_i u_{k-i} \quad (4.2)$$

Onde N_m é o horizonte de modelo, que corresponde ao número de coeficientes da resposta ao impulso necessário para representar 90-95% da resposta estacionária do sistema. No apêndice A, são apresentados os algoritmos DMC, para sistemas monovariáveis (SISO) e Multivariáveis (MIMO).

De acordo com Rezende (2003), o projeto de um controlador DMC consiste em estimar os valores dos parâmetros de forma a permitir um melhor desempenho do controlador. Os principais parâmetros são descritos a seguir.

O horizonte de modelo (N_m) que corresponde ao tempo de resposta em malha aberta para atingir 95-99% do estado estacionário.

O horizonte de predição (N_y) que representa o número de predições que é usado nos cálculos de otimização. Um aumento de N_y resulta em uma ação de controle mais conservativa, a qual tem um efeito estabilizante, mas também aumenta o esforço computacional.

O horizonte de controle (N_u) como dito anteriormente é o número de ações de controle futuras que são calculadas na otimização para reduzir os erros preditos. À medida que esta variável aumenta o esforço computacional também é aumentado. Segundo Seborg (1999), uma recomendação é escolher N_u igual ao tempo de resposta em malha aberta para atingir 60% do estado estacionário. Para $N_u > 5$, o valor de N_y não é tão crucial já que somente a primeira ação de controle é implementada. Contudo, um valor muito grande de N_u resulta em excessiva ação de controle. Um valor pequeno de N_u conduz a um controlador robusto, relativamente sensível a erros de modelagem.

O fator de supressão (f) também é um parâmetro estimado. Valores grandes desse parâmetro penalizam muito a magnitude de Δu ocasionando, então, um controle menos vigoroso. Quando $f=0$, o ganho do controlador é muito sensível a N_u , principalmente devido ao mau condicionamento da matriz $A^T A$, e N_u deve ser tomado como um valor baixo. A

aproximação de Maurath *et al.* (1988) utiliza o método da decomposição do valor singular para eliminar o mau condicionamento de $A^T A$ com $f=0$, mas este não é um problema sério com $N_u=1$.

4.2.2 Controle Baseado em Modelo Usando Redes (MPC Neural)

Segundo Henson e Seborg (1997), a motivação para o desenvolvimento de algoritmos de identificação e controle voltados para sistemas não-lineares são:

- Avanços na teoria de sistemas não-lineares que possibilitaram o projeto de controladores não-lineares;
- Desenvolvimento de métodos de identificação eficientes para modelos não-lineares empíricos;
- Melhoria na capacidade dos *softwares e hardwares* voltados para controle através de computador, nos quais é possível incorporar modelos não-lineares complexos nos sistemas de controle de processos.

De acordo com Meleiro (2002), um aspecto muito importante do NMPC é o tipo de modelo de processo a ser usado. Os modelos determinísticos têm a capacidade de interpretação física e uma grande faixa de aplicabilidade. Contudo nem sempre estão disponíveis e frequentemente apresentam parâmetros e/ou variáveis de processos que não podem ser medidos. Devido a isto o uso dos modelos empíricos ou semi-empíricos, desenvolvidos a partir de dados experimentais, vem crescendo. Modelos dinâmicos não lineares a partir de dados entrada-saída tem sido foco de diversos estudos, onde a técnica predominante são as redes neurais artificiais (Morari e Lee, 1999).

A literatura apresenta uma série de trabalhos aplicando modelo dinâmico neural em algoritmos de controle preditivos, dentre os quais se podem citar: Psychogios e Ungar (1991), Lee e Park (1992), De Souza Jr (1993), Morris *et al.* (1994), Najim *et al.* (1997), Vega *et al.* (2000), Santos *et al.* (2000a), Ender e Maciel Filho (2001a), Nikravesh *et al.* (2000), Benne *et al.* (2000).

O algoritmo de controle MPC Neural é uma variação do controle DMC clássico, onde ao invés de um modelo linear de controle será incorporada uma rede neural previamente treinada, conforme descrito no capítulo 3. Esta rede irá representar o comportamento dinâmico do processo fermentativo, realizando as previsões ao longo do horizonte de tempos de amostragem, sendo acionada tantas vezes for o valor do horizonte de predição.

De acordo com Ender (2001a), é de fundamental importância que a rede treinada consiga representar adequadamente o processo. Para isto os pares de treinamento devem conter informações significativas e consistentes em toda a faixa de operação.

4.3 CONTROLE ADAPTATIVO

De acordo com Assis (2001), o controle adaptativo surge da observação direta de que todos os seres da natureza possuem capacidades adaptativas sem as quais seria praticamente impossível a vida na Terra, devido à diversidade de situações adversas entradas, tais como sazonalidades no clima, mudanças no *habitat*.

O objetivo do controle adaptativo é ajustar os parâmetros do controlador de forma a compensar possíveis perturbações dinâmicas no sistema, causadas por variações nas condições operacionais do processo.

Conceitualmente as redes neurais também se adaptam a certo conjunto de dados, pelo ajuste de seus pesos, por isso existe certa conexão entre o controle adaptativo e as redes neurais artificiais (Ballini e Von Zuben, 2000).

4.3.1 Controle Direto Baseado no Erro Global (Condeg Modificado)

O algoritmo denominado Controle Neural Direto baseado no Erro Global (CoNDEG) Modificado foi proposto por Duarte (2004). O presente algoritmo está baseado no modelo inverso e no controle adaptativo, apresentando um mecanismo diferenciado para ajustar o controlador que é baseado em redes neurais artificiais. A principal vantagem do uso do algoritmo é a atualização dos parâmetros do controlador ao longo do tempo (*on-*

line), através de uma estratégia de aprendizagem usando três redes conforme descrito no capítulo 3, que permite um controle mais efetivo, pois acompanha as variações dinâmicas do processo.

A Figura 4.2 mostra o esquema do algoritmo de controle, onde são utilizadas redes neurais artificiais atuando como controlador neural (modelo direto), e modelo neural do processo (modelo indireto). O qual é utilizado na rotina de otimização do controlador. A rede neural do controlador é otimizada *on-line*, a cada tempo de amostragem, usando o erro global estimado para malha fechada, obtido através do modelo neural do processo.

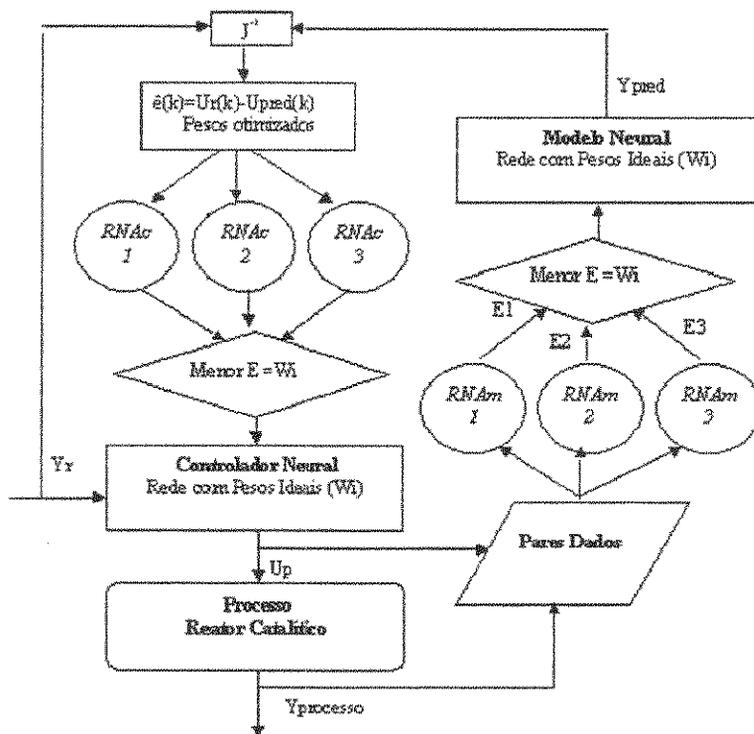


Figura 4.2 Algoritmo de Controle CONDEG Modificado

Na rotina de otimização, o projeto do controlador utiliza as mesmas entradas que serão utilizadas no controlador neural para cada instante de amostragem. Esta rotina ajusta os pesos da rede neural do controlador de tal forma a minimizar o erro global estimado, onde a resposta da malha fechada é proveniente do modelo neural do processo.

O erro global não pode ser diretamente aplicado na aprendizagem das redes neurais do controlador, pois estes estão associados às respostas do processo e não às ações de controle, as quais são as respostas da rede neural a ser treinada. Por isso são retropropagados através de matriz Jacobiana do processo.

Para o ajuste do controlador, primeiramente foram retropropagados e normalizados individualmente os valores de referência (Y_r) e valores preditos da rede neural para a malha fechada (\hat{Y}) e posteriormente gerados os erros. Desta forma o erro utilizado na aprendizagem pode assumir valores positivos e negativos e podem-se utilizar as funções de ativação sigmoideal e tangente hiperbólica, para treinamento das redes, conforme descrito a seguir.

A matriz Jacobiana pode ser determinada a partir do modelo neural pela variação de cada variável manipulada, em torno do seu ponto de operação e medindo a variação nas respostas do processo. Considerando a matriz jacobiana:

$$J = \begin{pmatrix} \frac{\partial y_1}{\partial u_1} & \dots & \frac{\partial y_1}{\partial u_i} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_i}{\partial u_1} & \dots & \frac{\partial y_i}{\partial u_i} \end{pmatrix} \quad (4.3)$$

Considerando que as variáveis y_j e u_j são as respostas estimadas da malha fechada e a ação de controle que gerou esta resposta estimada da malha fechada, respectivamente, pode-se escrever:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_i \end{pmatrix} = \begin{pmatrix} \frac{\partial y_1}{\partial u_1} & \dots & \frac{\partial y_1}{\partial u_i} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_i}{\partial u_1} & \dots & \frac{\partial y_i}{\partial u_i} \end{pmatrix} * \begin{pmatrix} u_1 \\ \vdots \\ u_i \end{pmatrix} \quad (4.4)$$

Então, os valores encontrados com a retropropagação são:

$$\hat{U} = J^{-1} * \hat{Y}$$

$$U_r = J^{-1} * Y_r \quad (4.5)$$

Assim, o erro global ($\hat{e}(k)$) é calculado de acordo com a expressão:

$$\hat{e}(k) = U_r(k) - \hat{U}(k) \quad (4.6)$$

com:

$$\begin{aligned} U_r(k) &= [u_r(1), \dots, u_r(i)]^T \\ \hat{U}(k) &= [\hat{u}(1,k), \dots, \hat{u}(i,k)]^T \end{aligned} \quad (4.7)$$

onde:

- $U_r(k)$ é o vetor de ações de controle desejado para malha fechada;
- $\hat{U}(k)$ é o vetor de ações de controle estimado para malha fechada.
- i : refere-se as malhas de controle;
- $u_r(i,k)$: vetor que contém as ações de controle de referências do processo no instante de tempo k ;
- $\hat{u}(i,k)$: vetor que contém as ações de controle previstas a partir do modelo neural do processo.

Este erro global é gerado inúmeras vezes, até ser atingida a tolerância permitida para o erro quadrático da rede neural ou alcançar-se o número máximo de iterações preestabelecidas, para atender as limitações de tempo disponível para processamento das informações, devido à aprendizagem *on-line*. Para garantir a boa representação dinâmica do processo através de redes neurais, adotou-se uma estratégia formada por três redes neurais que atuam paralelamente, conforme descrito no capítulo 3, item 3.8. Considerando que o ajuste da rede neural do controlador está baseado em predições, torna-se fundamental usar-se um modelo que represente com fidelidade o comportamento dinâmico do processo. Por

esta razão, sempre que o erro quadrático, de pelo menos uma das redes neurais do modelo do processo, não atingir a tolerância desejada, o procedimento de aprendizagem das redes neurais do controlador não será executado. Nestas circunstâncias, o modelo do processo a ser usado, será aquele que apresentar menor erro quadrático na representação do vetor de padrões e a rede do controlador a ser usada na geração das ações de controle será aquela que apresentar menor erro quadrático para a situação atual. O modelo dinâmico do processo é utilizado para a geração do erro quadrático da rede neural do controlador.

O controlador baseado em redes neurais representando a dinâmica inversa apresenta a característica de gerar ações de controle enérgicas. Por esta razão, utiliza-se um filtro na saída do controlador amenizando os incrementos nas ações de controle gerados. O filtro a ser utilizado para penalizar as ações de controle apresenta a seguinte forma:

$$u(i, k) = \lambda_i u(i, k - 1) + (1 - \lambda_i) u(i, k) \quad (4.8)$$

4.3.2 Condeg Modificado com Filtro de Kalman

Este algoritmo apresenta uma melhoria no Condeg Modificado pela forma diferenciada de estimar o parâmetro de penalização (λ_i) dado pela equação (4.8), mudando de um valor fixo para um valor auto-ajustável calculado através do Filtro de Kalman. A seguir será apresentada uma introdução sobre o Filtro de Kalman e o algoritmo proposto.

4.3.2.1 Filtro de Kalman

Segundo Junior Bitencourt *et. al* (2007), o filtro de Kalman é um algoritmo recursivo muito eficiente, capaz de estimar as variáveis de estado de sistemas representados por equações de estado lineares. Ele incorpora toda informação obtida a partir de medições ruidosas para estimar as variáveis desejadas. Pressupõe-se que o sistema seja perturbado por ruídos brancos e gaussianos, de forma que os estados possam ser tratados como variáveis aleatórias gaussianas.

Criado por Rudolph E. Kalman em 1960, o filtro de Kalman (FK) foi desenvolvido inicialmente como uma solução recursiva para filtragem linear de dados discretos. Para isto, utiliza equações matemáticas que implementam um estimador preditivo de estados,

buscando corrigir iterativamente a resposta de um determinado sistema através de múltiplas variáveis relacionadas a ele (Kalman, 1960).

Sua origem foi dentro da área de engenharia elétrica relacionada à teoria do controle de sistemas, mas hoje suas áreas de aplicação são muito diversificadas, tais como: processamento de imagem, supervisores de eventos discretos, processamento de sinais, sistemas de inferência, além das áreas econômica, financeira e biomédica (Pinho *et. al*, 2005; Junior Bitencourt *et. al*, 2007).

Na área de controle de processos, o uso do filtro de Kalman já foi utilizado para estimar fator de supressão para um controlador DMC (Ender *et. al*, 2002), fator de penalização de controladores (Ender *et. al*, 2003) e também para ajustar os pesos de redes neurais que atuam como controlador (Scheffer, 2006).

A estrutura original do filtro de Kalman está baseada no modelo de espaço de estados (MEE), desenvolvido, inicialmente, para sistemas lineares. Este modelo busca definir a relação entrada-saída de um sistema linear indiretamente, por meio de um conjunto de variáveis internas x_t , denominadas estados. Os estados são influenciados por seus próprios valores passados x_{t-1} e pelas entradas do sistema, u_t , e por sua vez influenciam as saídas do sistema z_t .

Segundo Harvey (1989), o modelo propriamente dito é definido por duas equações: equação de processo (equação 4.9) e equação de medida (equação 4.10), também conhecida por equação de observação.

$$x_t = Ax_{t-1} + Bu_t + w_{t-1}, \quad x_t \in \mathfrak{R} \quad (4.9)$$

$$z_t = Hx_t + Bu_t + v_t, \quad z_t \in \mathfrak{R}^n \quad (4.10)$$

A equação do processo estima o estado atual x_t também chamado de estado posterior (*a posteriori* – no tempo t dado que se conhece a medição ou observação), através da soma de três parcelas matemáticas. A primeira é composta por uma matriz A , que relaciona o estado atual do processo com o estado anterior (x_{t-1}). A segunda também possui

uma matriz B que relaciona o estado a *posteriori* com os pontos de monitoramento de entrada e controle do processo, representados por um vetor u_t . A última parcela (w_{t-1}) representa um ruído de processo do tipo branco com distribuição de probabilidade normal, que simula uma alteração gradativa de seu estado ao longo do tempo.

A equação de medida tem como objetivo associar o estado de entrada à saída do sistema através de um histórico de medidas. Para isto, é criado um novo valor de medida z_t , através da soma do produto da matriz de correlação H , das medidas de entrada pelo estado atual, com o ruído de medida v_t , que possui as mesmas características do ruído de processo, porém, que independe do estado do mesmo.

De acordo com Aiube (2005), a abstração deste modelo permite a elaboração de um algoritmo computacional capaz de estimar os valores ótimos do vetor de estado para os quais a saída vai ser ótima. Desta forma, é possível gerar uma seqüência de valores de estado por unidade de tempo, prevendo estados futuros com a utilização de estados atuais, permitindo a implementação de sistemas com atualizações em tempo real, de acordo com a Figura 4.3, que descreve o algoritmo de Filtro Kalman.

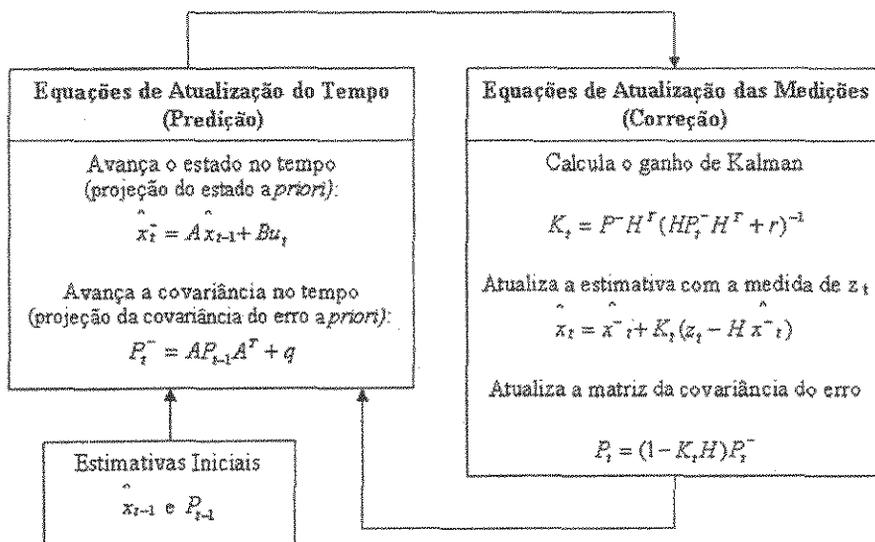


Figura 4.3 Algoritmo do Filtro de Kalman

O algoritmo está basicamente dividido em duas etapas, uma de atualização do tempo e outra de atualização de medida. As equações de atualização do tempo são responsáveis pelo avanço das variáveis de estado e das covariâncias no tempo para se obter, desta forma, as estimativas anteriores (a priori) para o próximo instante. As equações de atualização das medições são responsáveis pela retroalimentação, ou seja, incorporam uma nova informação da variável observável nas estimativas anteriores para obter um ganho (ou melhoria) na estimação posterior. As equações de atualização do tempo são denominadas equações de previsão. As equações de atualização das medições são denominadas equações de correção. Onde: q e r representam as matrizes de covariância dos ruídos w e v , respectivamente e P_t^- (a priori) e P_t (a posteriori) são as matrizes covariância do erro, obtidas a partir dos erros relativos e_t^- (a priori) e e_t (a posteriori), conforme as equações a seguir:

$$e_t^- = x_t - \hat{x}_t^- \quad (4.11)$$

$$e_t = x_t - \hat{x}_t \quad (4.12)$$

$$P_t^- = E[e_t^- - e_t^{-T}] \quad (4.13)$$

$$P_t = E[e_t - e_t^T] \quad (4.14)$$

Aiube (2005) descreve que o primeiro passo do algoritmo é estimar um valor de medida inicial que representará o estado anterior \hat{x}_t^- e sua respectiva estimativa P_{t-1} . Com estes valores determinados, se torna possível executar a etapa de atualização de tempo, responsável por calcular a projeção do estado a priori, aplicando a conhecida equação de processo do modelo de espaço de estados e a projeção do erro de covariância. A etapa de atualização de medida, que é responsável por calcular o ganho de Kalman, estimar o estado a posteriori e atualizar a covariância do erro (Harvey, 1989).

Para um melhor aprofundamento no tema podem ser consultados diversos trabalhos e livros, entre eles: Lewis (1986), Harvey (1989), Papoulis (1991), Durbin e Koopman (2002) e Scheffer (2006).

4.3.2.2 Algoritmo Proposto para Estimar o Parâmetro de Penalização

Para estimar o parâmetro de penalização (λ_i) dado pela equação (4.8), foi proposto o Condeg Modificado com filtro de Kalman. Com a utilização do filtro os valores dos parâmetros (λ_i) são calculados ao longo do tempo, permitindo um melhor ajuste no controlador, pois o parâmetro de penalização influencia diretamente na ação de controle, dando ao controlador a flexibilidade de ter ações mais ou menos enérgicas, de acordo com o valor estimado, que está relacionado diretamente com o processo (Duarte, 2004b).

Conforme descrito no item anterior, para utilizar o filtro de Kalman são necessárias às equações de processo e de medida. As equações (4.15 e 4.16) apresentam a dinâmica do parâmetro de penalização (λ_i).

$$\lambda_{i,t} = \lambda_{i,t-1} + w_{i,t-1} \quad (4.15)$$

$$z_{i,t} = H\lambda_{i,t} + v_{i,t} \quad (4.16)$$

Como já discutido, a equação de observação é baseada no tempo passado, e o parâmetro é estimado de forma retropropagativa, as equações utilizadas para o caso fermentativo consideram o ganho relativo conforme a equação (4.17):

$$\begin{aligned} \lambda_{k+1} &= \lambda_k + (w_k) \\ \text{abs}(y_k^{\text{setpoint}} - y_k) &= K \times \text{abs}(u_k - u_{k-1}) \lambda_k (+v_k) \end{aligned} \quad (4.17)$$

Considerando o caso de estudo com 3 variáveis manipuladas (MIMO 3x3), têm-se as equações (4.18) e (4.19)

$$\begin{aligned} \lambda_{1,k+1} &= \lambda_{1,k} (+ w_k) \\ \lambda_{2,k+1} &= \lambda_{2,k} (+ w_k) \\ \lambda_{3,k+1} &= \lambda_{3,k} (+ w_k) \end{aligned} \tag{4.18}$$

$$\begin{bmatrix} \text{abs}(y_{1,k}^{\text{setpo int}} - y_{1,k}) \\ \text{abs}(y_{2,k}^{\text{setpo int}} - y_{2,k}) \\ \text{abs}(y_{3,k}^{\text{setpo int}} - y_{3,k}) \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} \text{abs}(u_{1,k} - u_{1,k-1}) & 0 & 0 \\ 0 & \text{abs}(u_{2,k} - u_{2,k-1}) & 0 \\ 0 & 0 & \text{abs}(u_{3,k} - u_{3,k-1}) \end{bmatrix} \begin{bmatrix} \lambda_{1,k} \\ \lambda_{2,k} \\ \lambda_{3,k} \end{bmatrix} \tag{4.19}$$

Como são possíveis diferentes formas para determinar o sistema de identificação, foram propostos quatro algoritmos, conforme a Tabela 4.1. A diferença entre os algoritmos está na forma de se calcular o ganho:

Tabela 4.1 Algoritmos Propostos para o Filtro de Kalman

Nome do Algoritmo	Forma de Cálculo do Ganho	Equação
Condeg Kalman 1	Ganho constante (K_{ij})	4.20
Condeg Kalman 2	Ganho constante com fator de velocidade (K_{ij})	4.21
Condeg Kalman 3	Ganho variável (J_{ij})	4.23
Condeg Kalman 4	Ganho variável com fator de velocidade (J_{ij})	4.24

- **Condeg Kalman 1** – Este algoritmo segue a equação (4.20), e considera o ganho constante.

$$\begin{bmatrix} \text{abs}(y_{1,k}^{\text{setpo int}} - y_{1,k}) \\ \text{abs}(y_{2,k}^{\text{setpo int}} - y_{2,k}) \\ \text{abs}(y_{3,k}^{\text{setpo int}} - y_{3,k}) \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} \text{abs}(u_{1,k} - u_{1,k-1}) & 0 & 0 \\ 0 & \text{abs}(u_{2,k} - u_{2,k-1}) & 0 \\ 0 & 0 & \text{abs}(u_{3,k} - u_{3,k-1}) \end{bmatrix} \begin{bmatrix} \lambda_{1,k} \\ \lambda_{2,k} \\ \lambda_{3,k} \end{bmatrix} \tag{4.20}$$

- **Condeg Kalman 2** – O algoritmo de Kalman 2 considera o ganho constante, similar ao Kalman 1, mas associado a um fator de velocidade (θ), conforme a equação (4.21).

$$\theta \begin{bmatrix} \text{abs}(y_{1,k}^{\text{setpo int}} - y_{1,k}) \\ \text{abs}(y_{2,k}^{\text{setpo int}} - y_{2,k}) \\ \text{abs}(y_{3,k}^{\text{setpo int}} - y_{3,k}) \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} \text{abs}(u_{1,k} - u_{1,k-1}) & 0 & 0 \\ 0 & \text{abs}(u_{2,k} - u_{2,k-1}) & 0 \\ 0 & 0 & \text{abs}(u_{3,k} - u_{3,k-1}) \end{bmatrix} \begin{bmatrix} \lambda_{1,k} \\ \lambda_{2,k} \\ \lambda_{3,k} \end{bmatrix} \quad (4.21)$$

Para o cálculo do ganho constante, foram realizadas perturbações nas variáveis de entrada (Fo, R_{Flash} e R_{Microrg}) e observadas as variáveis de saída (S, P e X), considerando o processo em malha aberta. Os valores para K_{ii}, são fixos e obtidos, conforme, a equação 4.22:

$$\begin{bmatrix} \frac{y_1}{u_1} & \frac{y_1}{u_2} & \frac{y_1}{u_3} \\ \frac{y_2}{u_1} & \frac{y_2}{u_2} & \frac{y_2}{u_3} \\ \frac{y_3}{u_1} & \frac{y_3}{u_2} & \frac{y_3}{u_3} \end{bmatrix} \quad (4.22)$$

- **Condeg Kalman 3** – Este algoritmo se diferencia dos anteriores apenas na forma do cálculo do ganho, pois, considera o ganho (J_{ii}) variável, através de coeficientes Jacobianos, conforme a equação (4.23).

$$\begin{bmatrix} \text{abs}(y_{1,k}^{\text{setpo int}} - y_{1,k}) \\ \text{abs}(y_{2,k}^{\text{setpo int}} - y_{2,k}) \\ \text{abs}(y_{3,k}^{\text{setpo int}} - y_{3,k}) \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \begin{bmatrix} \text{abs}(u_{1,k} - u_{1,k-1}) & 0 & 0 \\ 0 & \text{abs}(u_{2,k} - u_{2,k-1}) & 0 \\ 0 & 0 & \text{abs}(u_{3,k} - u_{3,k-1}) \end{bmatrix} \begin{bmatrix} \lambda_{1,k} \\ \lambda_{2,k} \\ \lambda_{3,k} \end{bmatrix} \quad (4.23)$$

- **Condeg Kalman 4** – Similar ao algoritmo Kalman 3, este algoritmo também considera o cálculo do ganho (J_{ii}) através de coeficientes Jacobianos, mas está associado a o fator de velocidade (θ), conforme a equação (4.24).

$$\theta \begin{bmatrix} \text{abs}(y_{1,k}^{\text{setpo int}} - y_{1,k}) \\ \text{abs}(y_{2,k}^{\text{setpo int}} - y_{2,k}) \\ \text{abs}(y_{3,k}^{\text{setpo int}} - y_{3,k}) \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \begin{bmatrix} \text{abs}(u_{1,k} - u_{1,k-1}) & 0 & 0 \\ 0 & \text{abs}(u_{2,k} - u_{2,k-1}) & 0 \\ 0 & 0 & \text{abs}(u_{3,k} - u_{3,k-1}) \end{bmatrix} \begin{bmatrix} \lambda_{1,k} \\ \lambda_{2,k} \\ \lambda_{3,k} \end{bmatrix} \quad (4.24)$$

Para determinação do ganho variável (J_{ij}) o cálculo foi realizado conforme o Jacobiano, que relaciona os valores das entradas e das saídas, descrito na equação 4.25:

$$\begin{bmatrix} \frac{\partial y_1}{\partial u_1} & \frac{\partial y_1}{\partial u_2} & \frac{\partial y_1}{\partial u_3} \\ \frac{\partial y_2}{\partial u_1} & \frac{\partial y_2}{\partial u_2} & \frac{\partial y_2}{\partial u_3} \\ \frac{\partial y_3}{\partial u_1} & \frac{\partial y_3}{\partial u_2} & \frac{\partial y_3}{\partial u_3} \end{bmatrix} \quad (4.25)$$

Quando se utiliza o fator velocidade (θ), mesmo sendo necessário estimar um parâmetro, a utilização do filtro de Kalman ainda reduz o número de parâmetros a serem determinados, e permite uma adaptação do parâmetro de penalização (λ_i) ao longo do tempo.

4 CONCLUSÕES

Neste capítulo foi discutida a importância do controle de processos, foram apresentados dois tipos de controle adaptativo e preditivo, que serão estudados no desenvolvimento desta tese. No controle preditivo foram discutidos os algoritmos Controle por matriz dinâmica (DMC) e Controle Preditivo Baseado em Modelo MPC Neural, que utiliza uma rede neural com aprendizagem *on-line* para predição do modelo. Para o controle adaptativo foi proposto um controle neural direto baseado no erro global, denominado Condeg Modificado, que utiliza redes neurais para atuarem como controlador e na rotina de otimização do controlador representam a dinâmica do processo. Todas as redes utilizadas possuem aprendizagem *on-line*.

Uma introdução sobre o filtro de Kalman foi apresentada para o entendimento do algoritmo Condeg Modificado com filtro de Kalman. Neste algoritmo, o filtro de Kalman foi utilizado para penalizar as ações de controle, tornando este parâmetro auto-ajustável.

Para avaliar diferentes configurações para o cálculo do ganho do filtro de Kalman, foram propostos quatro algoritmos diferentes, onde foram avaliados ganhos fixos, variáveis e associados a um fator de velocidade, que permite uma variação mais suave do parâmetro de penalização. Mesmo com a inclusão de um novo parâmetro (fator velocidade), fica reduzido o número de parâmetros estimados para um processo MIMO (3x3), como é o caso do processo estudado.

RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados os resultados para todos os algoritmos de controle propostos, para o processo extrativo de fermentação alcoólica, conforme descrito no capítulo 4.

5.1 RESULTADO PARA O CONTROLE PREDITIVO

No controle preditivo a resposta gerada pelo modelo é mais rápida que a gerada pelo processo real, desta forma, o controlador tem seus parâmetros ajustados antes da saída do processo real, através da predição do modelo.

Como discutido no capítulo 4, os controladores DMC e MPC Neural são dois tipos de controladores preditivos. A diferença entre os dois é o modelo utilizado: o DMC possui um modelo de convolução (linear) e o MPC Neural utiliza redes neurais artificiais para gerar a resposta predita (modelo não-linear). Além da utilização de modelos não-lineares é interessante que esses modelos tenham uma atualização ao longo do tempo. Isto pode ser alcançado através do uso de redes neurais com aprendizagem *on-line* para representar o modelo.

Para determinar os valores dos parâmetros usados no projeto dos controladores DMC e MPC Neural já discutido no Capítulo 4 foram realizadas diversas simulações, variando-se os parâmetros e observando-se a resposta do processo em malha fechada. Os resultados obtidos foram:

- Para valores do horizonte de modelo (N_m): DMC $N_m = 7$ e MPC Neural $N_m = 8$.
- Para valores do horizonte de predição (N_y): DMC $N_y = 5$ e MPC Neural $N_y = 3$.

- Para valores do horizonte de controle (N_u) dos dois algoritmos DMC e MPC Neural, o valor determinado foi $N_u = 1$.
- O fator de supressão (f) como discutido anteriormente é muito sensível ao horizonte de controle com $N_u = 1$, os valores utilizados serão: DMC = 0,35-50-50. MPC Neural = 0,7-50-50.

Para avaliar o desempenho dos controladores preditivos serão utilizados os problemas do tipo regulador e servo, discutidos no capítulo 2 nos subitens 2.3.3.1 e 2.3.3.2, respectivamente.

No Apêndice B serão apresentados todos os resultados para os algoritmos DMC e MPC Neural para os problemas do tipo servo, regulador e servo-regulador individualmente para melhor visualização dos resultados. Também pode ser visualizada a comparação entre os resultados do algoritmo MPC com e sem a aprendizagem das redes ao longo do tempo.

Para todos os resultados, o MPC Neural com aprendizagem *on-line* das redes apresentou uma performance igual ou superior ao DMC. A Figura 5.1 apresenta o resultado para a variável controlada P para problema do tipo servo em S, comparando a resposta para o algoritmo MPC Neural com redes que possuem apenas aprendizagem com pesos padrões (*off-line*) e redes treinadas ao longo do tempo (*on-line*).

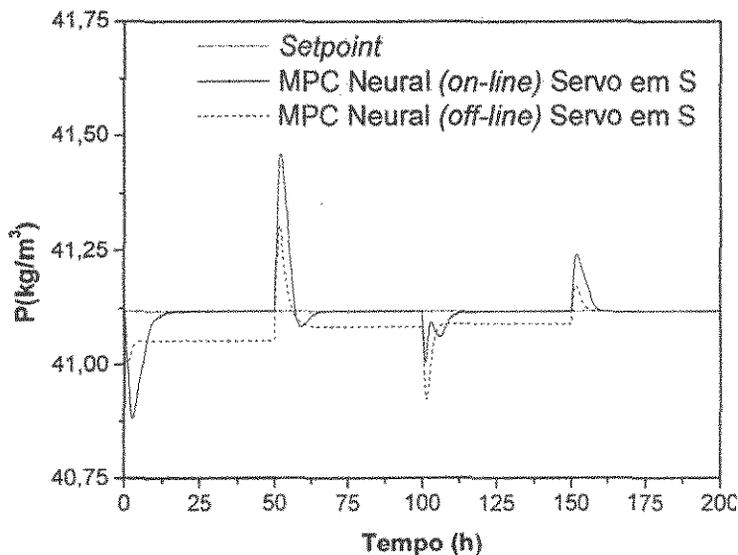


Figura 5.1 MPC Neural com aprendizagem On-line e Off-line

Como se observa na Figura 5.1 os resultados com aprendizagem *on-line* foram melhores, pois não apresentaram a presença de *off-set*. Devido a esta melhor performance como esperado para o MPC Neural com aprendizagem *on-line*, os resultados apresentados para comparação entre os algoritmos MPC Neural e DMC serão sempre utilizando MPC Neural com aprendizagem *on-line*.

Para problemas do tipo servo em S, P e X os resultados entre os dois algoritmos foram similares, por isso será apresentado apenas o resultado para o problema do tipo servo em S, conforme apresentado no item 5.1.1. As Figuras 5.2 e 5.3 apresentam os resultados obtidos para os dois algoritmos das variáveis controladas e das variáveis manipuladas, respectivamente. Os resultados para cada um dos algoritmos para os problemas do tipo servo em S, P e X são apresentados no apêndice B.

Nos subitens 5.1.2 e 5.1.3 serão apresentados os resultados para os dois algoritmos para problemas do tipo regulador e servo-regulador, respectivamente. Com os resultados obtidos para as variáveis controladas a manipuladas. Neste tipo de problemas o algoritmo usando MPC Neural com aprendizagem *on-line* como esperado também apresentou melhores resultados, por isso será utilizado sempre o algoritmo com aprendizagem *on-line*.

Os resultados foram similares para os problemas do tipo servo e regulador para ambos os algoritmos por isso serão apresentados os resultados para as variáveis controladas e manipuladas para o problema do tipo regulador e servo em S. No apêndice B, como discutido anteriormente, são apresentados os outros problemas do tipo servo e regulador, para ambos os algoritmos.

5.1.1 Problema do Tipo Servo

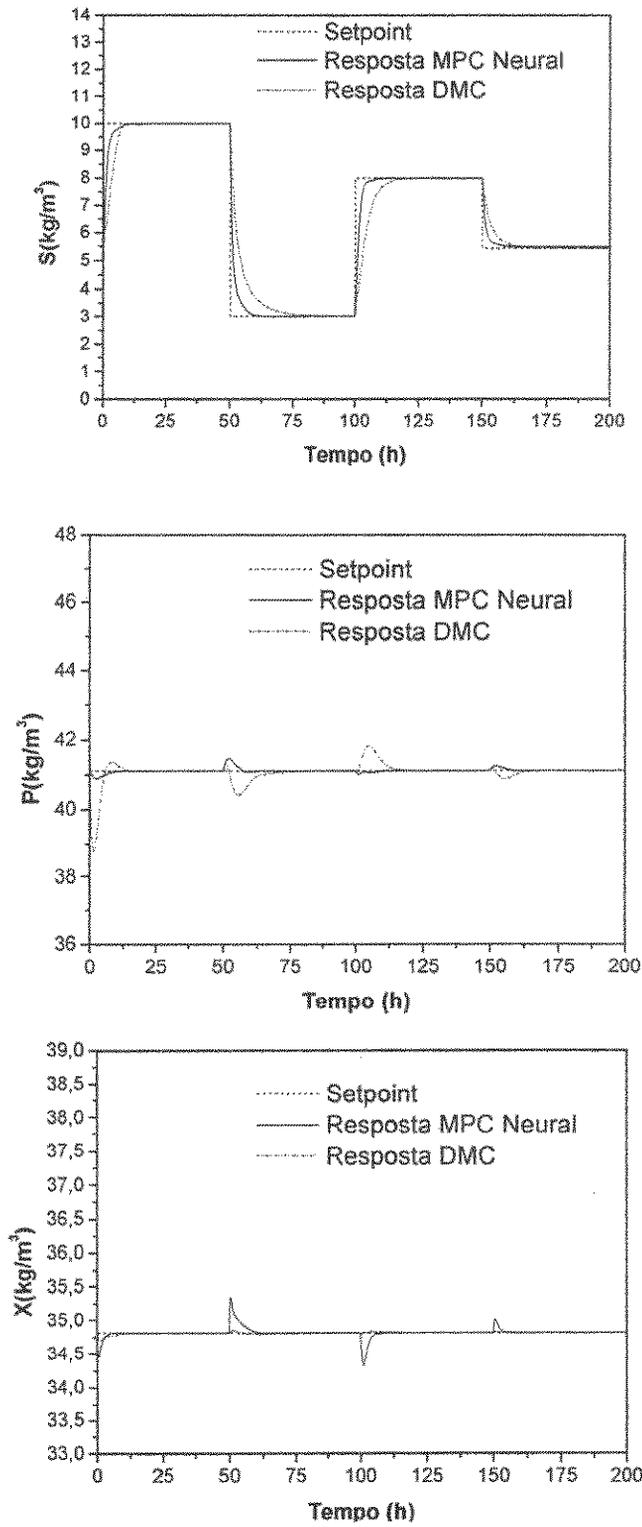


Figura 5.2 Variáveis Controladas para Problema Servo em S

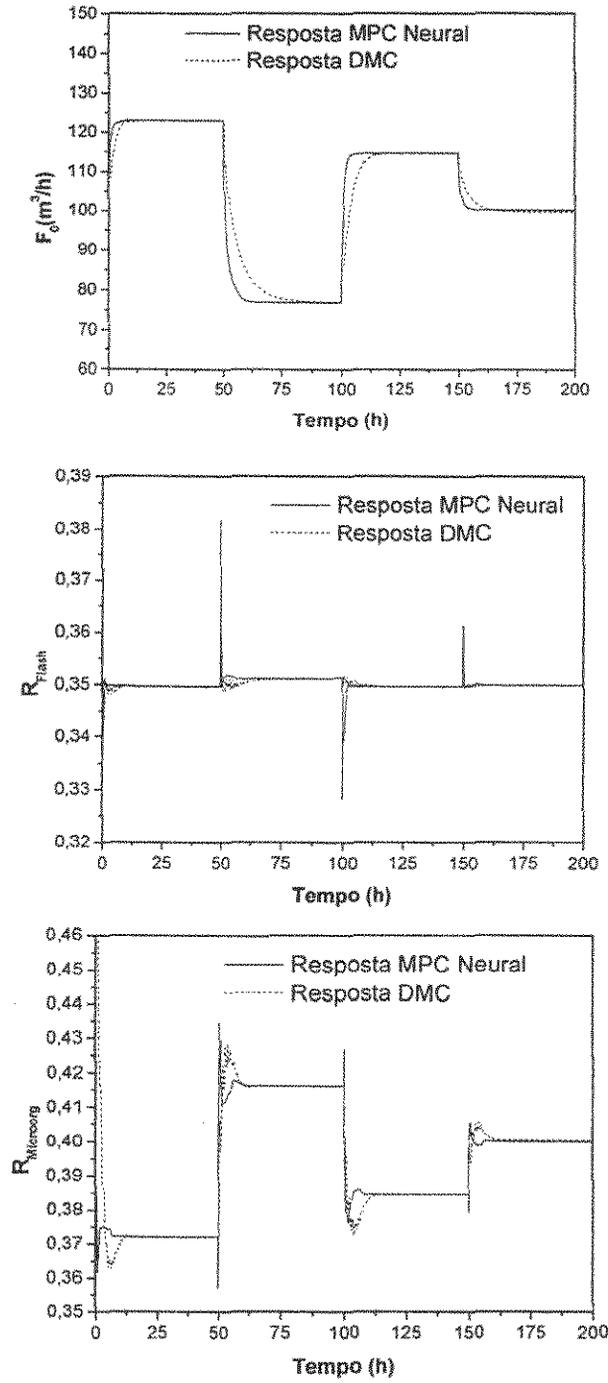


Figura 5.3 Variáveis Manipuladas para Problema Servo em S

5.1.2 Problema do Tipo Regulador

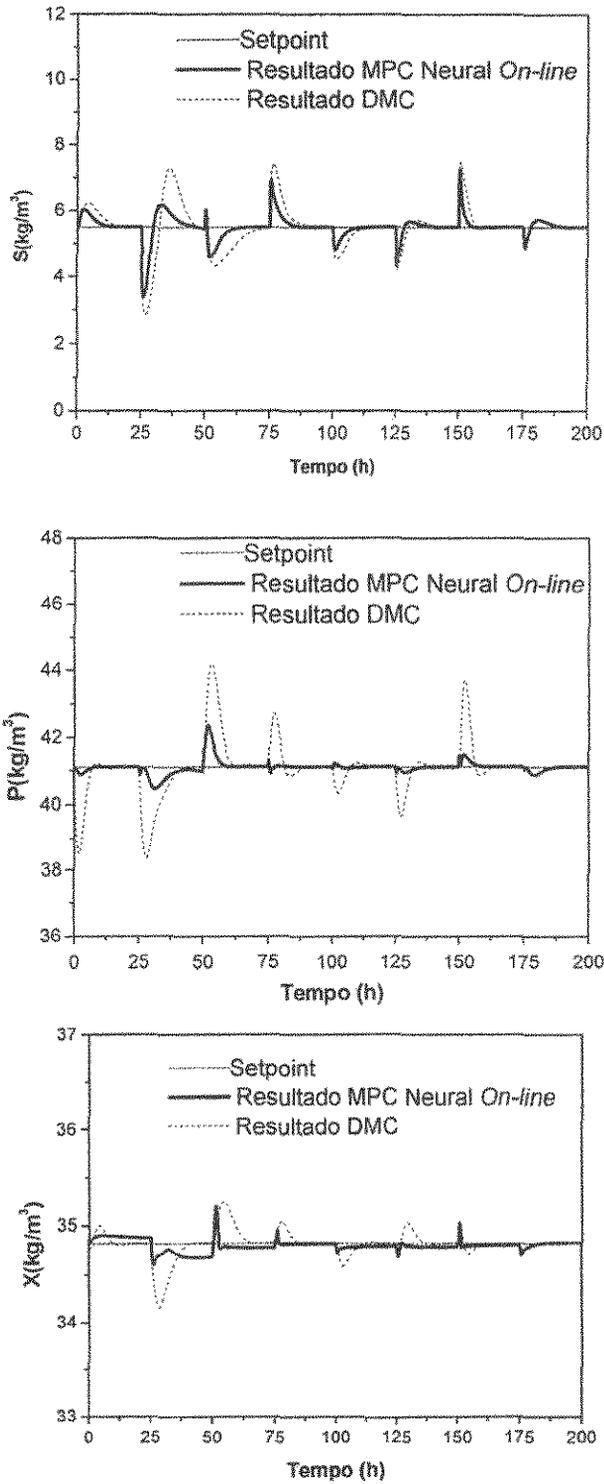


Figura 5.4 Variáveis Controladas para Problema Regulador



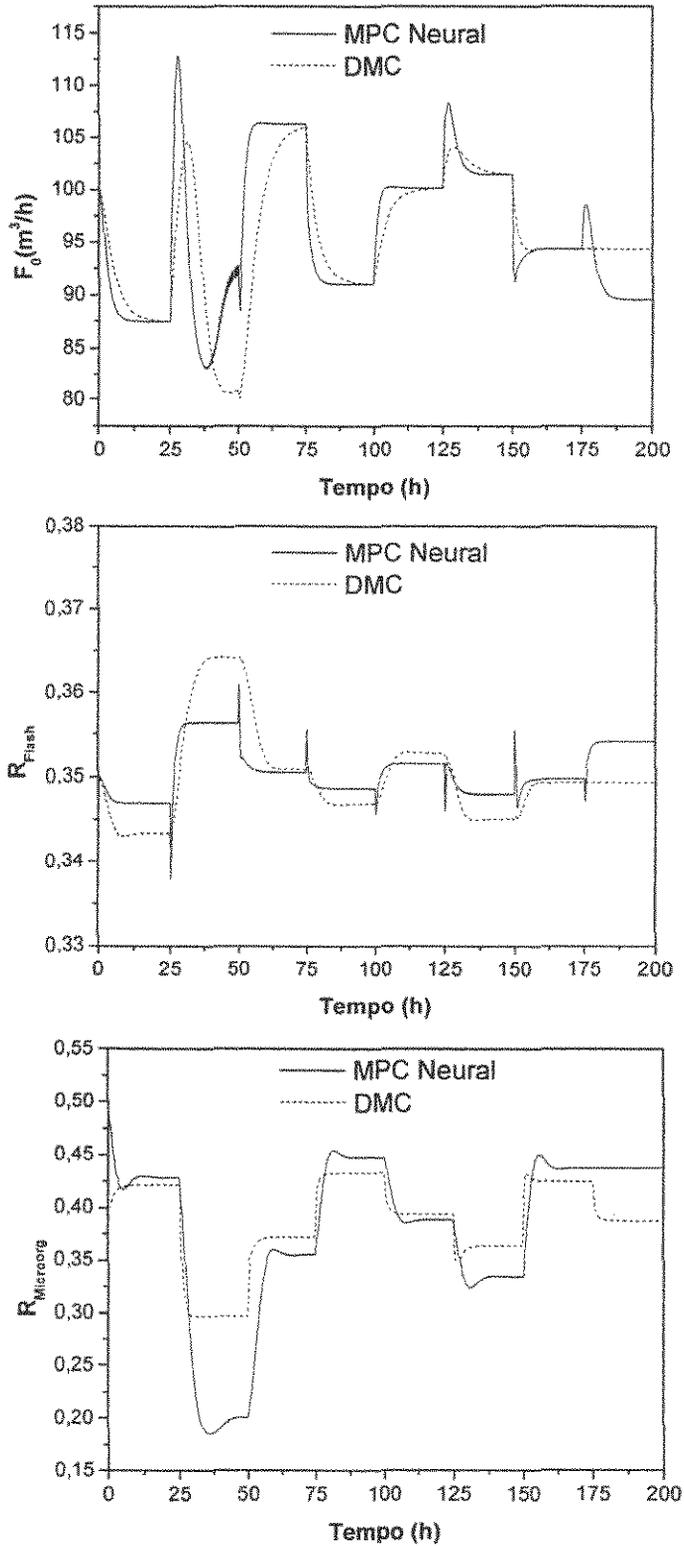


Figura 5.5 Variáveis Manipuladas para Problema Regulador

5.1.3 Problema do Tipo Servo e Regulador

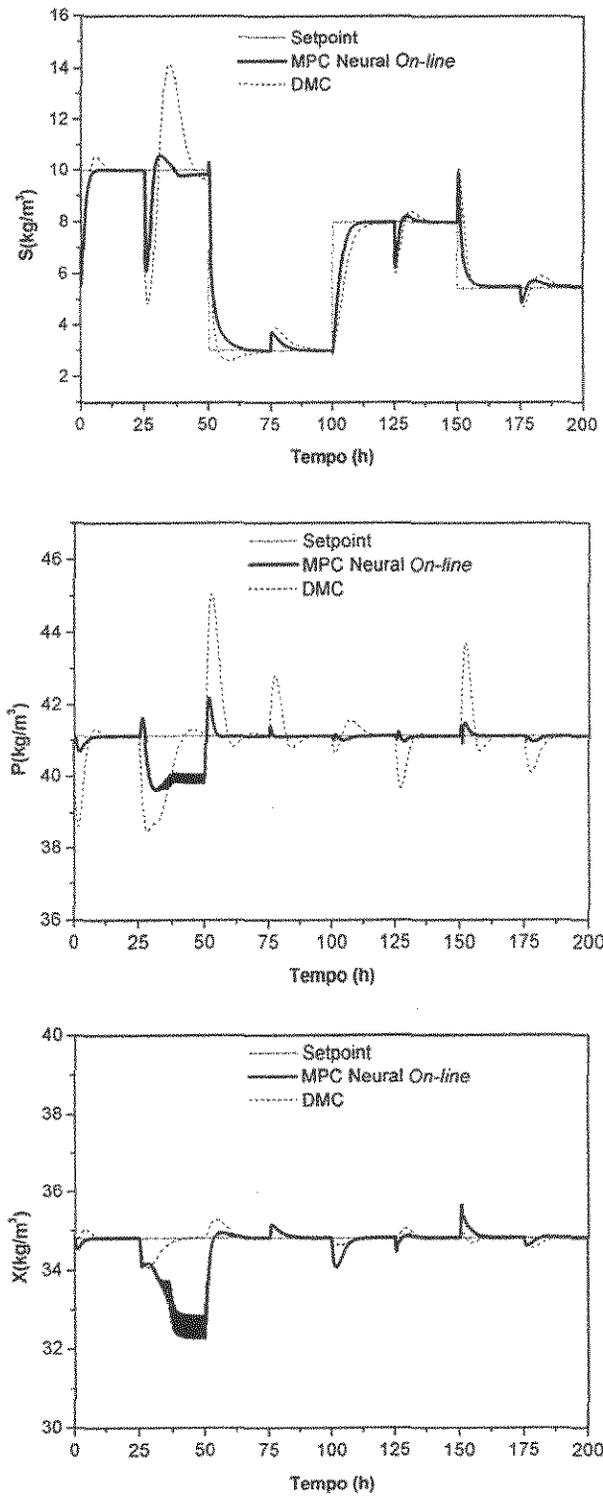


Figura 5.6 Variáveis Controladas para Problema Servo S e Regulador

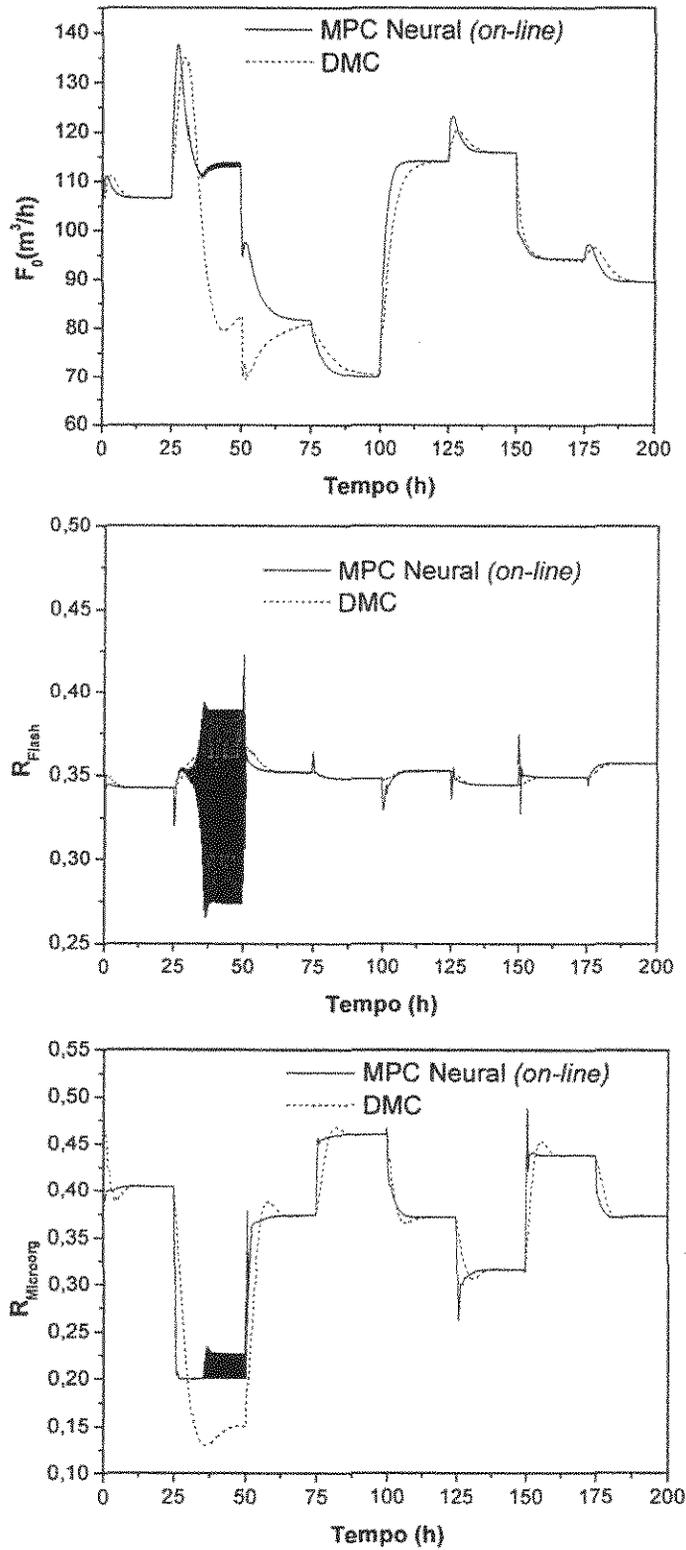


Figura 5.7 Variáveis Manipuladas para Problema Servo S e Regulador

Com os resultados apresentados nas sessões anteriores pode se observar que o algoritmo do tipo MPC Neural com aprendizagem ao longo do tempo (*on-line*) apresentou melhores resultados comparado com o que possuía apenas a aprendizagem com os pesos padrões.

Como esperado, com o processo de aprendizagem *on-line* as redes conseguiam representar melhor o modelo, já que conseguiam “aprender” sua nova dinâmica e conseqüentemente o controlador, que é baseado no modelo, apresentou uma melhor performance. Isto pode ser visualizado principalmente quando o problema é do tipo regulador e a rede que foi treinada apenas com variáveis de entrada (F_0 , R_{Flash} e $R_{Microorg}$) e de saída (S, P e X) conseguiu representar bem o processo mesmo com duas novas variáveis (S_0 e T_0) que não tinham sido apresentadas a rede durante a aprendizagem *off-line*. Como já comentado anteriormente, todos os resultados para esta comparação estão apresentados no Apêndice B.

Melhores desempenhos foram obtidos com o MPC Neural *on-line* para o problema do tipo regulador assim como para problemas do tipo servo, inclusive com a ausência do *off-set* presente no MPC Neural *off-line*, o MPC Neural foi escolhido para comparação com o DMC, que possui um modelo linear, como já discutido anteriormente.

Para os problemas do tipo servo, os resultados obtidos para os dois algoritmos foram satisfatórios para as variáveis controladas, com ambos os algoritmos atingindo o *setpoint*. O que se pode observar é que o MPC Neural apresentou uma resposta mais rápida para a variável controlada S e menor quantidade de picos em P comparada ao DMC. Mas como se observa na Figura 5.3, as variáveis manipuladas apresentaram picos instantâneos para o MPC Neural o que não é uma situação adequada para implementação do controlado. Devido a movimentos bruscos das válvulas.

Para o problema do tipo regulador, o MPC Neural apresentou melhores resultados para todas as variáveis controladas comparado ao DMC. Isso era esperado porque além do modelo não-linear ser o mais aplicado para este tipo de processo a aprendizagem *on-line* das redes que representam o processo conseguiram se adaptar as novas perturbações que ocorreram no sistema. Por isso o DMC apresentou uma maior quantidade de picos,

especialmente para a variável controlada P, que representa a quantidade de produto no fermentador.

Para problemas do tipo servo-regulador os resultados dos algoritmos foram semelhantes e por isso optou-se por se apresentar apenas o problema do tipo regulador e servo em S. Para todos os resultados deste tipo de problema, o MPC Neural apresentou melhores resultados para as variáveis controladas comparado ao DMC, com a presença de picos menores, com exceção dos tempos iniciais entre 25 e 50 horas em que o MPC Neural não atingiu o *setpoint* para as variáveis P e X, situação que também ocorre para o controlador DMC. Isso pode ser observado em todos os problemas do tipo servo e regulador atuando de forma simultânea. A razão para este comportamento deve-se ao fato de que neste período foram consideradas situações extremas onde se tem a temperatura no limite e a quantidade de substrato mínima na alimentação, gerando assim a condições extremas de operação, levando a uma grande perturbação no processo. Isto gera uma diminuição na quantidade de microrganismos e conseqüentemente a quantidade de produto no fermentador. Para as variáveis manipuladas do problema servo-regulador em S (Figura 5.7), foram apresentados resultados com instabilidade e picos, o que representaria movimentos bruscos e repetitivos da válvula, dificultando a implementação do controlador. O controle adaptativo apresenta uma vantagem para estes casos, pois tem um fator de velocidade associado, evitando assim estas mudanças bruscas nas variáveis manipuladas.

5.2 RESULTADO PARA O CONTROLE ADAPTATIVO

A seguir, serão apresentados os resultados para o algoritmo de controle Condeg Modificado, que é um controlador adaptativo baseado em redes neurais artificiais com aprendizagem *on-line* que para o auto-ajuste da penalização da ação de controle, foram utilizados quatro algoritmos do Filtro de Kalman, que possuem diferentes formas para o cálculo do ganho, conforme descrito no capítulo 4.

O algoritmo Kalman 1 possui o ganho constante, o algoritmo Kalman 2 possui o ganho constante associado a um fator de velocidade, o algoritmo Kalman 3 possui o ganho variável e o algoritmo 4 possui o ganho variável associado a um fator de velocidade.

Para os algoritmos Kalman 2 e Kalman 4 que possuem o ganho associado a um fator de velocidade (θ) foram feitas variações nos valores dos mesmos fazendo variações de $\theta = 0,1$; $\theta = 0,5$ e $\theta = 0,9$.

Para o algoritmo Kalman 2 os resultados para o problema do tipo servo em S, regulador e regulador e servo (S, P e X) os resultados para θ igual a 0,5 e 0,9 foram muito similares e $\theta = 0,1$ apresentou os melhores resultados. Para os outros problemas servos em X e P os resultados foram similares para todos os valores do fator de velocidade (θ).

Para o algoritmo de Kalman 4, os resultados para o problema servo em X e regulador e servo (S, P e X) apresentaram, como o algoritmo Kalman 2, os melhores resultados para menores valores de teta. Para os outros problemas servos em S e P e do regulador os resultados foram os mesmos para todos os valores do fator de velocidade.

Nas Figuras 5.8 e 5.9 são apresentados os resultados da variável controlada S para o problema do tipo servo em S, dos algoritmos Kalman 2 e Kalman 4, respectivamente.

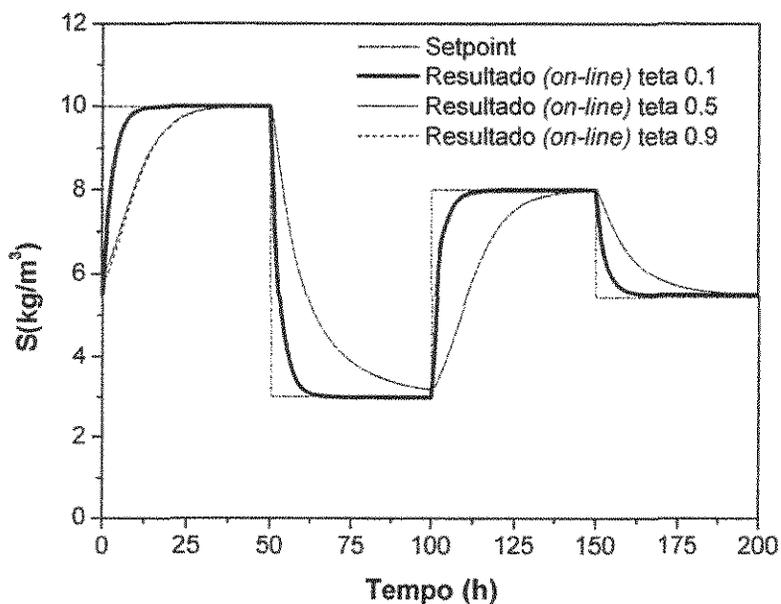


Figura 5.8 Comparação de teta para Algoritmo do Filtro de Kalman 2

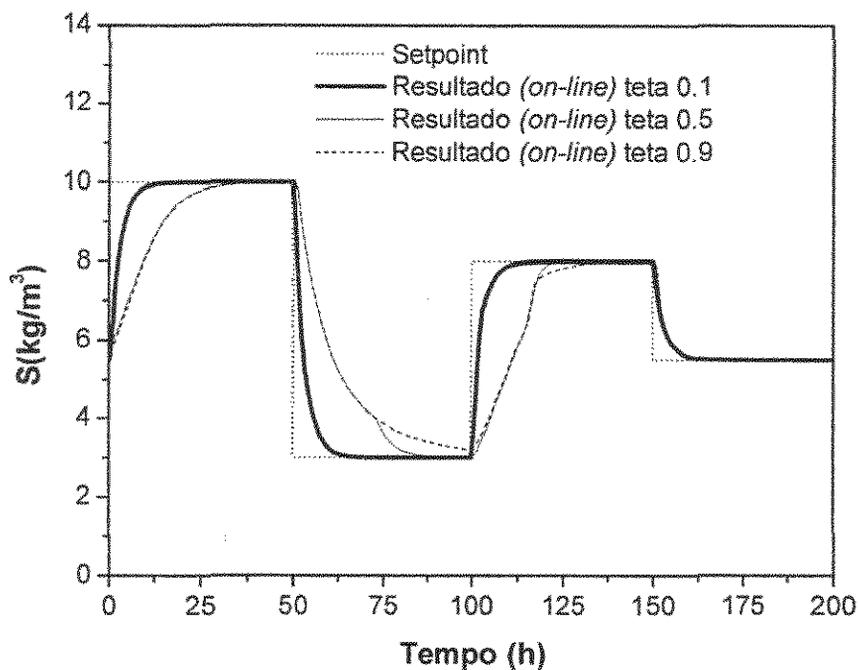


Figura 5.9 Comparação de teta para Algoritmo do Filtro de Kalman 4

Devido a estas considerações, serão apresentados os resultados apenas para os valores de $\theta = 0,1$. No apêndice C serão apresentados os resultados para os algoritmos Kalman 2 e Kalman 4 com todas as variações do fator de velocidade (θ).

Como já discutido anteriormente, no Capítulo 4, mesmo com a introdução de um novo parâmetro, o fator de velocidade, o uso do algoritmo do Filtro de Kalman para ajustar a penalização da ação de controle leva a uma diminuição do número de parâmetros e principalmente um auto-ajuste desta penalização, que torna o controle mais robusto.

Para análise deste tipo de controlador com seus respectivos algoritmos foram também avaliados problemas do tipo servo, regulador e servo regulador, conforme descrito nos subitens 5.2.1, 5.2.2 e 5.2.3, respectivamente.

Para o problema do tipo servo em X os resultados foram similares ao do problema do tipo servo em P; todos os algoritmos apresentaram as respostas muito próximas por isso este não foi apresentado.

5.2.1 Problema do Tipo Servo

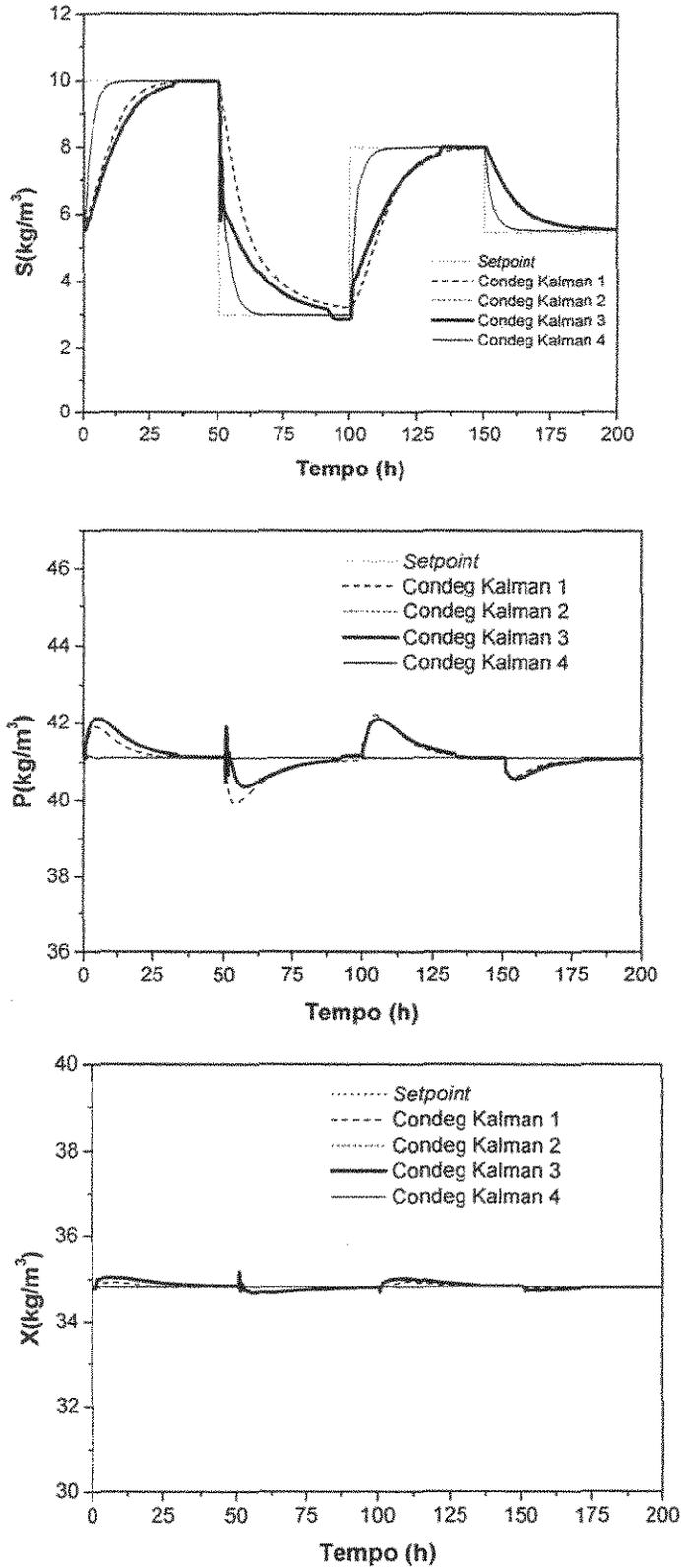


Figura 5.10 Variáveis Controladas para Problema Servo S

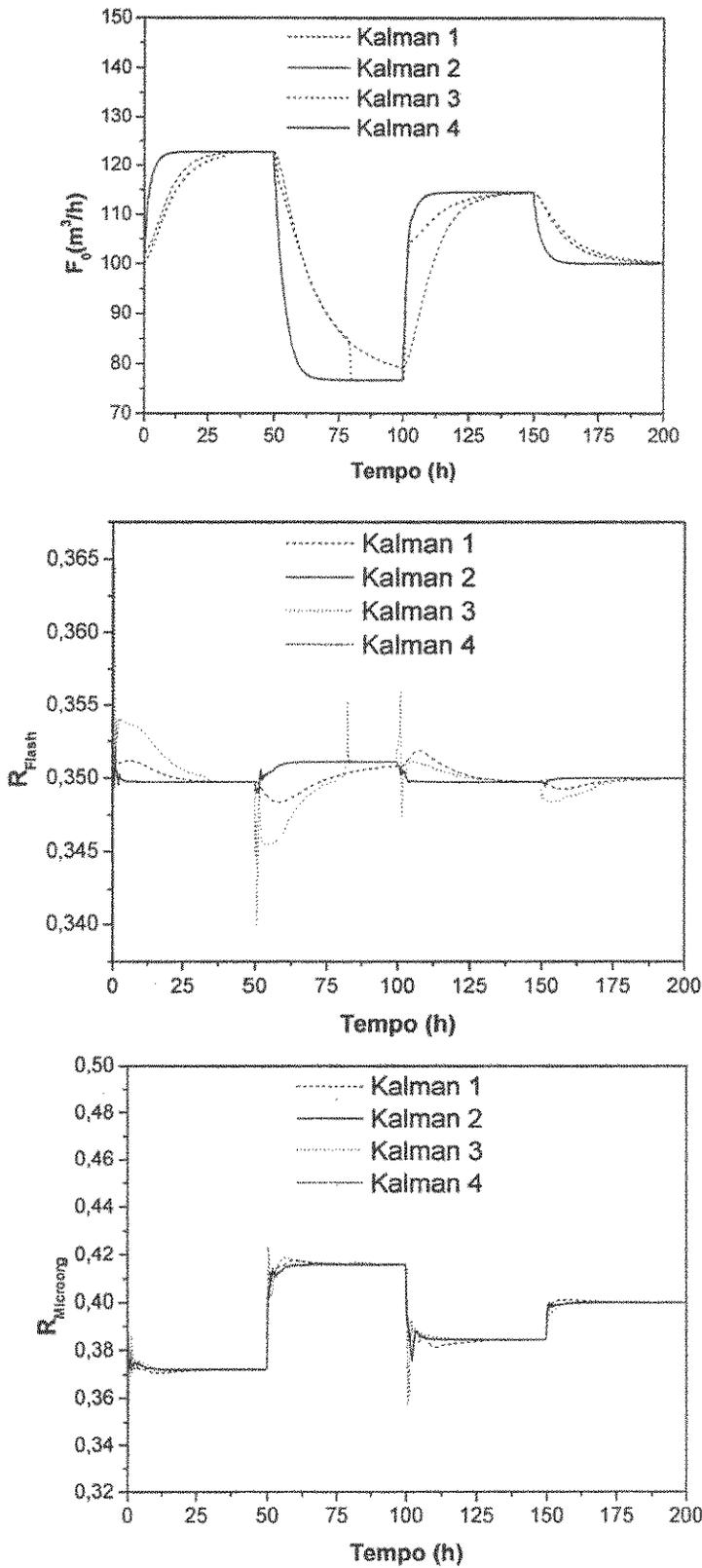


Figura 5.11 Variáveis Manipuladas para Problema Servo S

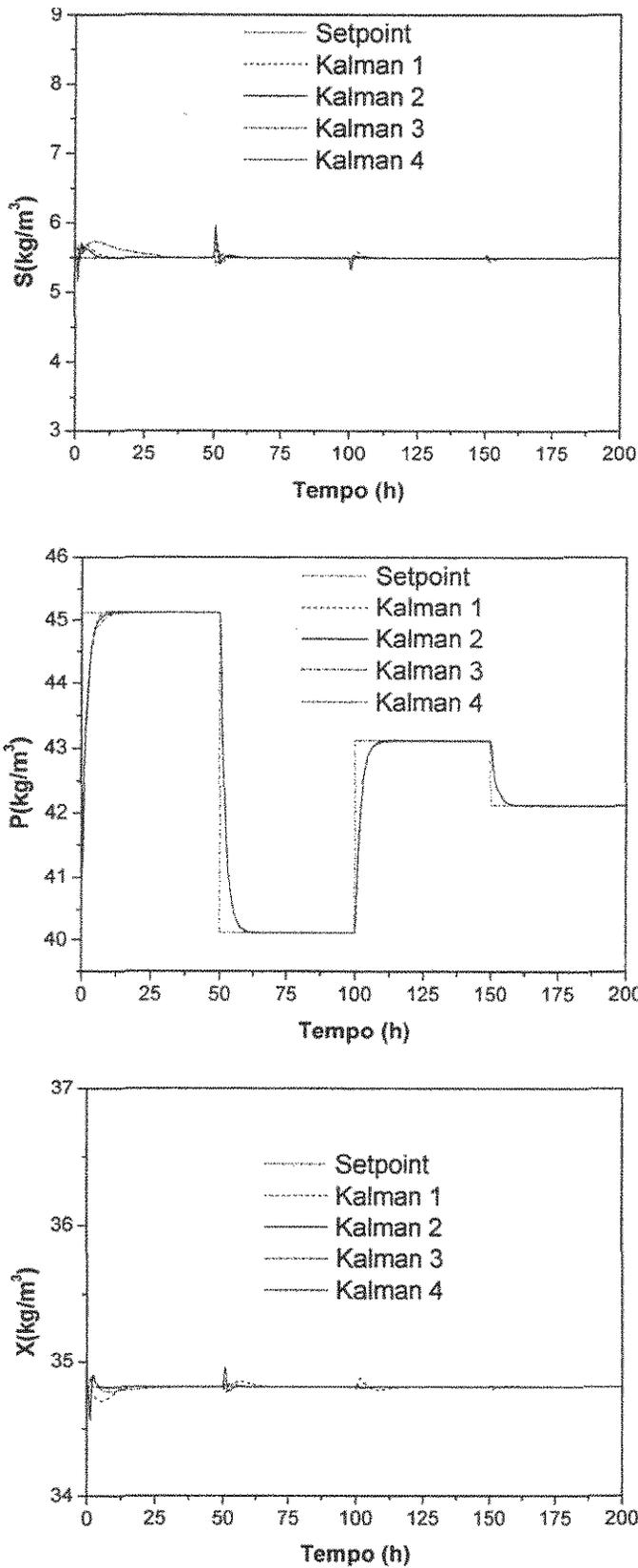


Figura 5.12 Variáveis Controladas para Problema Servo P

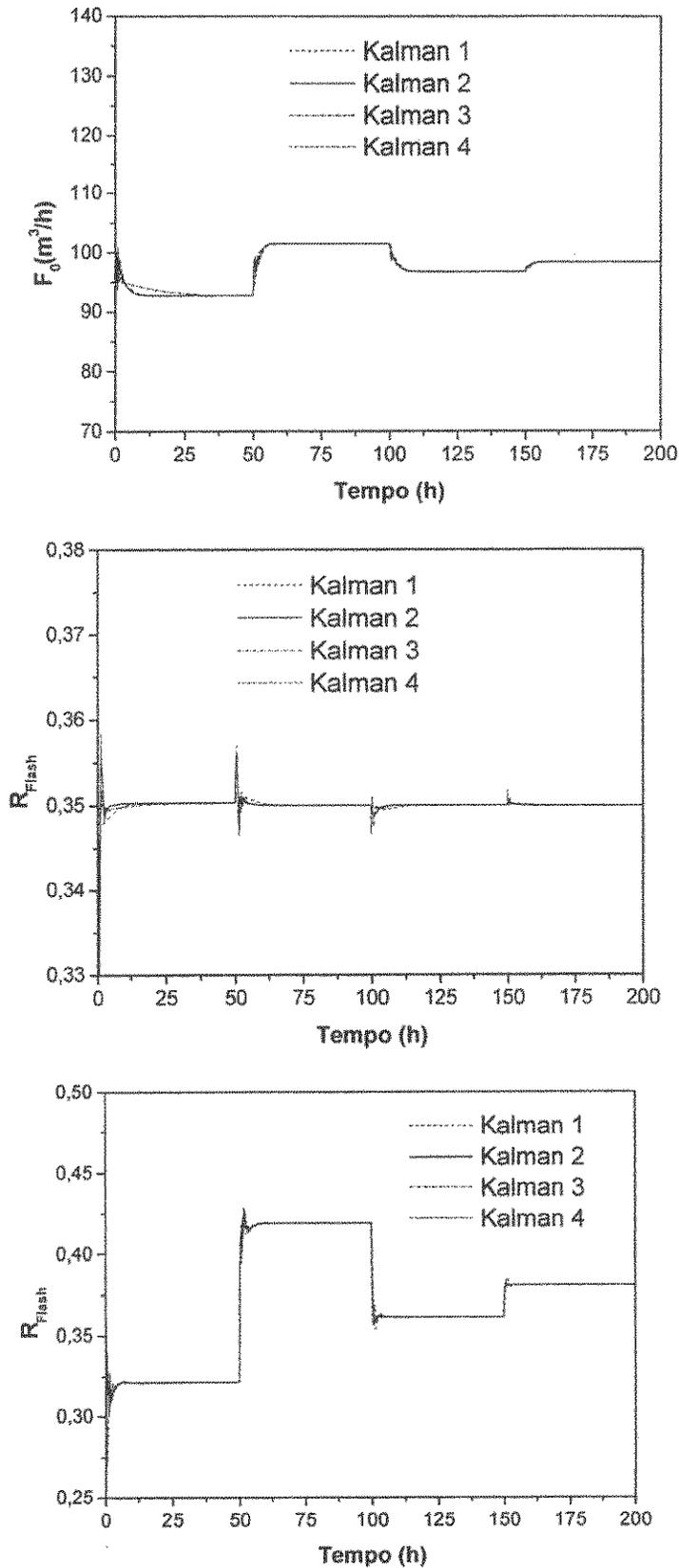


Figura 5.13 Variáveis Manipuladas para Problema Servo P

5.2.2 Problema do Tipo Regulador

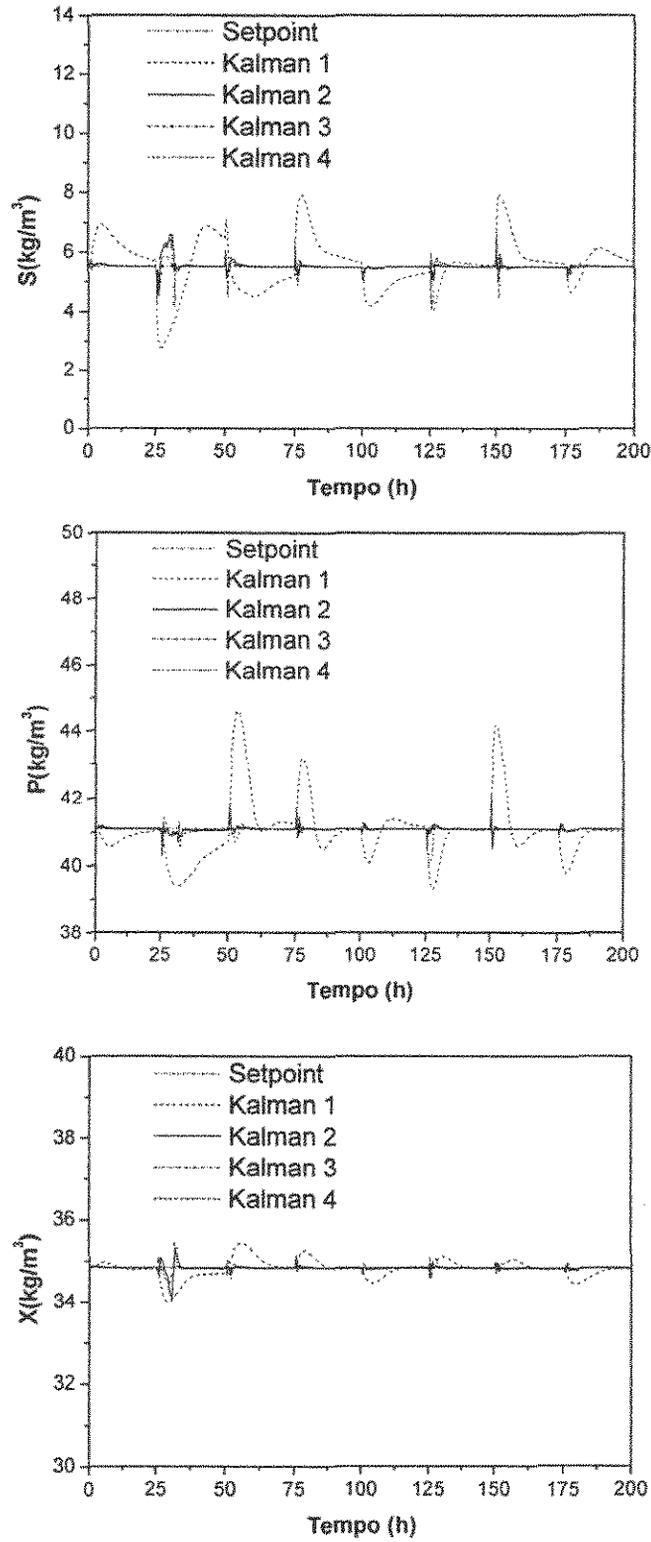


Figura 5.14 Variáveis Controladas para Problema Regulador

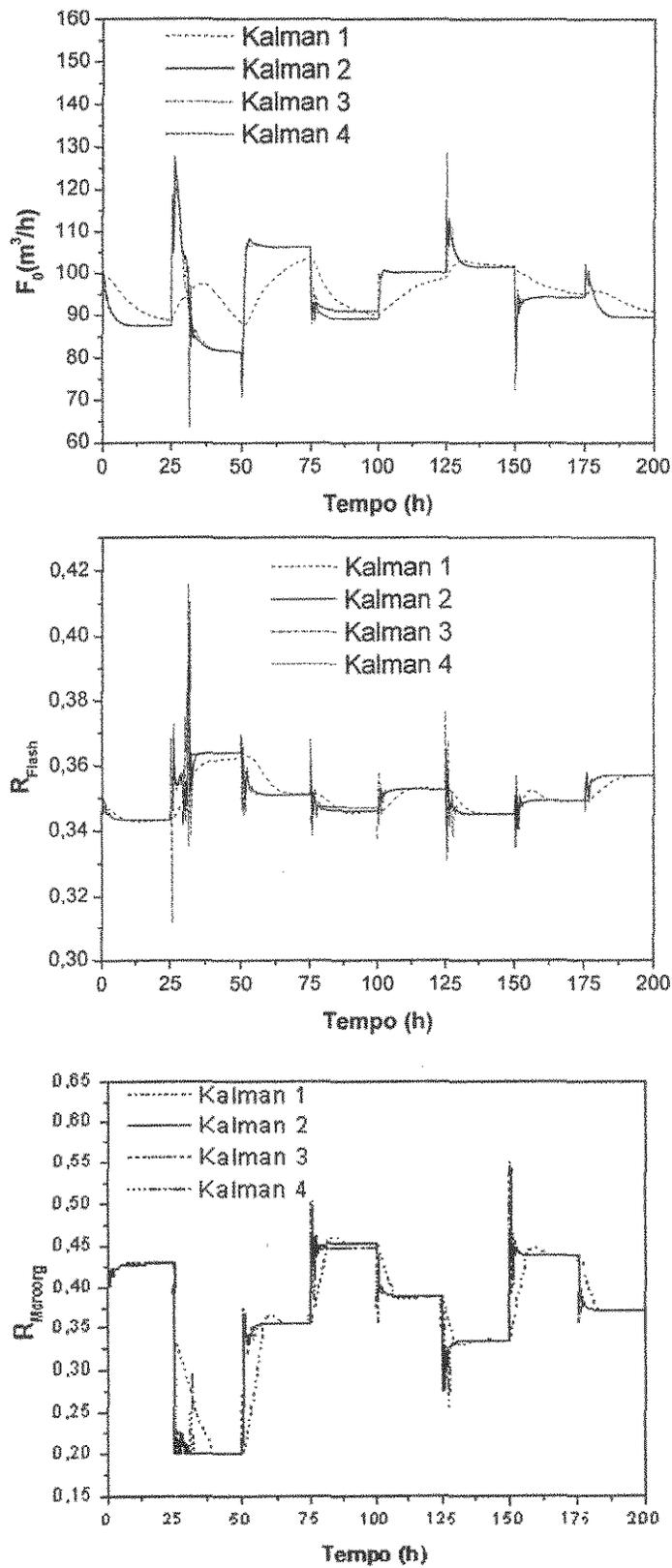


Figura 5.15 Variáveis Manipuladas para Problema Regulator

5.2.3 Problema do Tipo Servo e Regulador

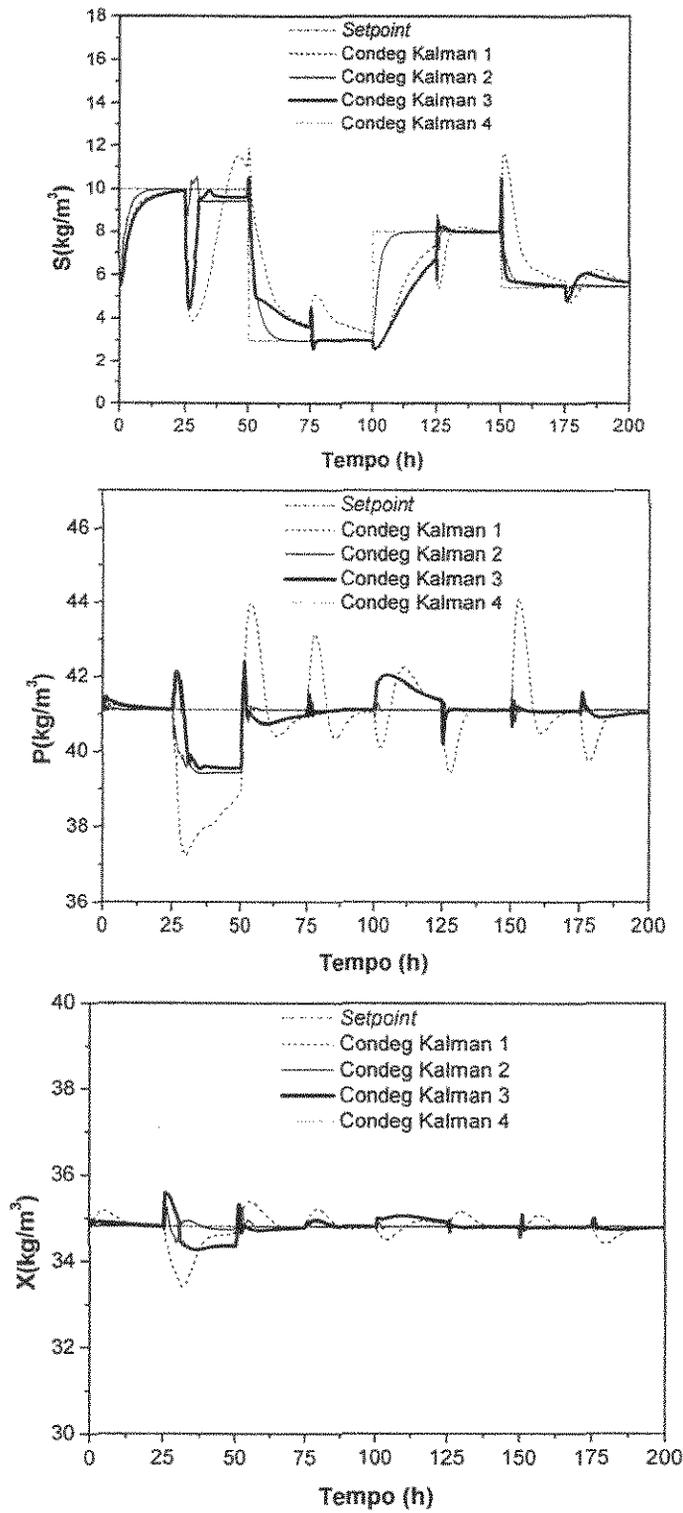


Figura 5.16 Variáveis Controladas para Problema Servo S e Regulador

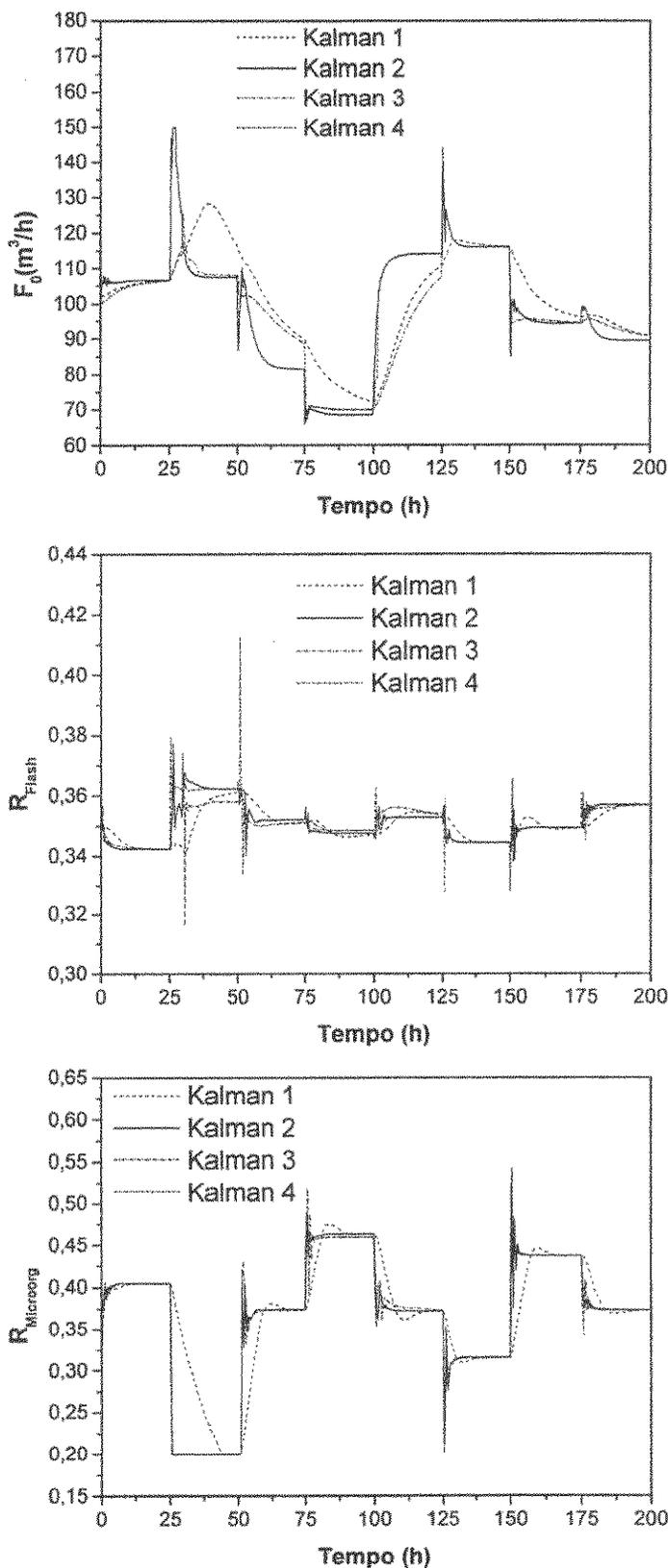


Figura 5.17 Variáveis Manipuladas para Problema Servo S e Regulador

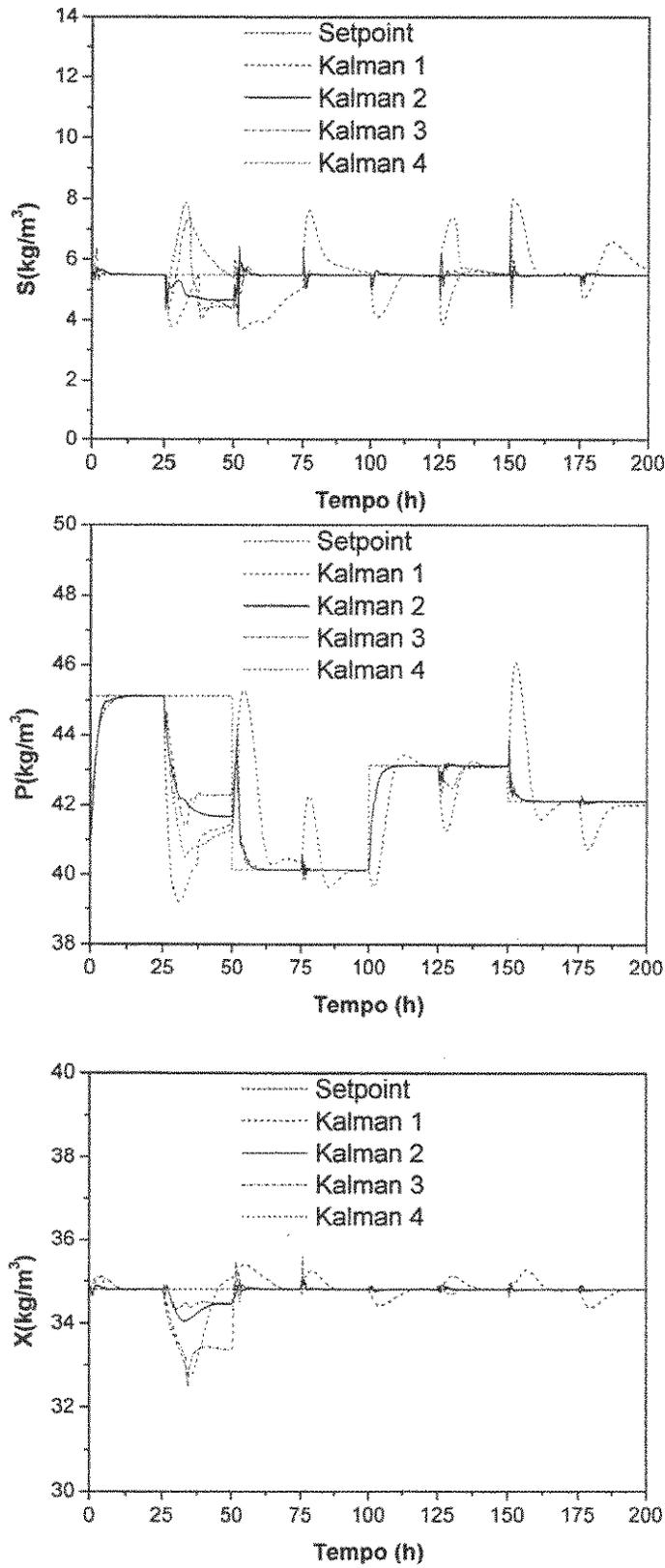


Figura 5.18 Variáveis Controladas para Problema Servo P e Regulador

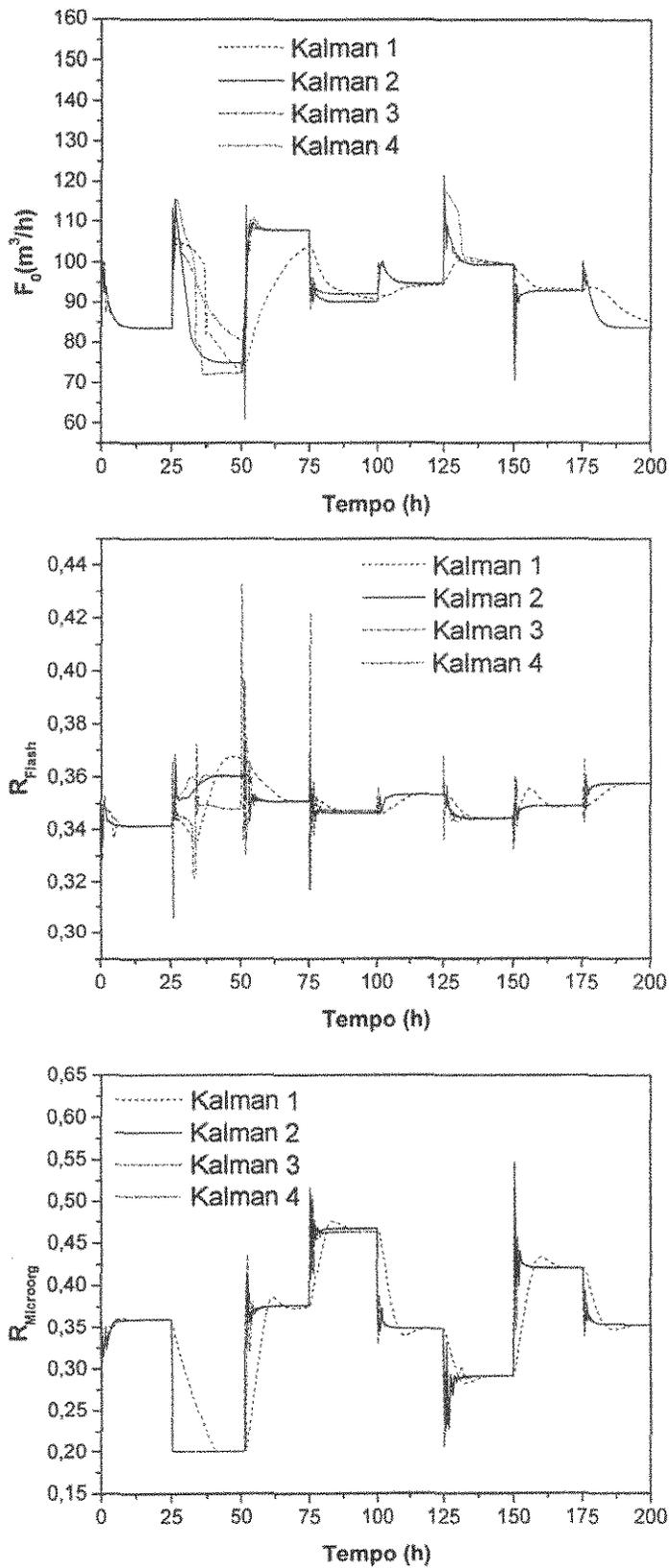


Figura 5.19 Variáveis Manipuladas para Problema Servo P e Regulador

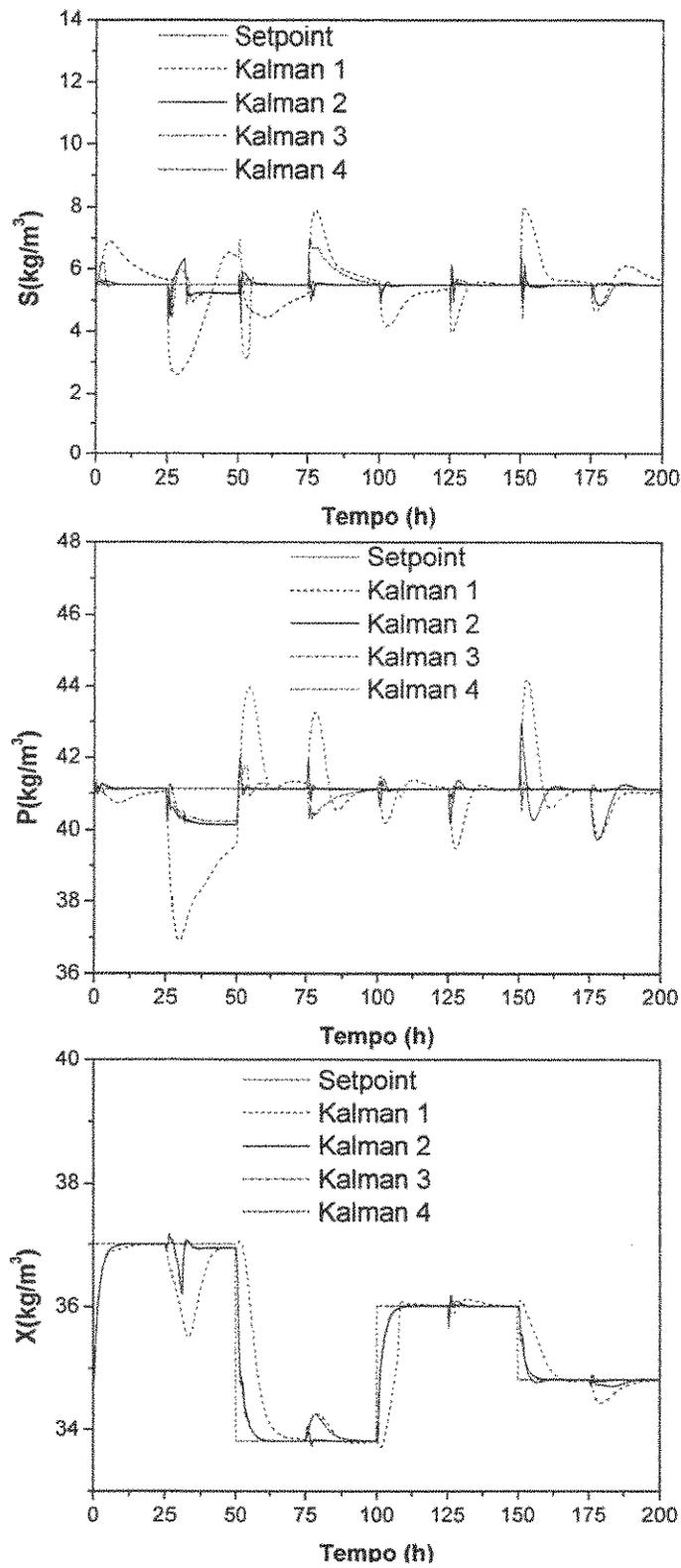


Figura 5.20 Variáveis Controladas para Problema Servo X e Regulador

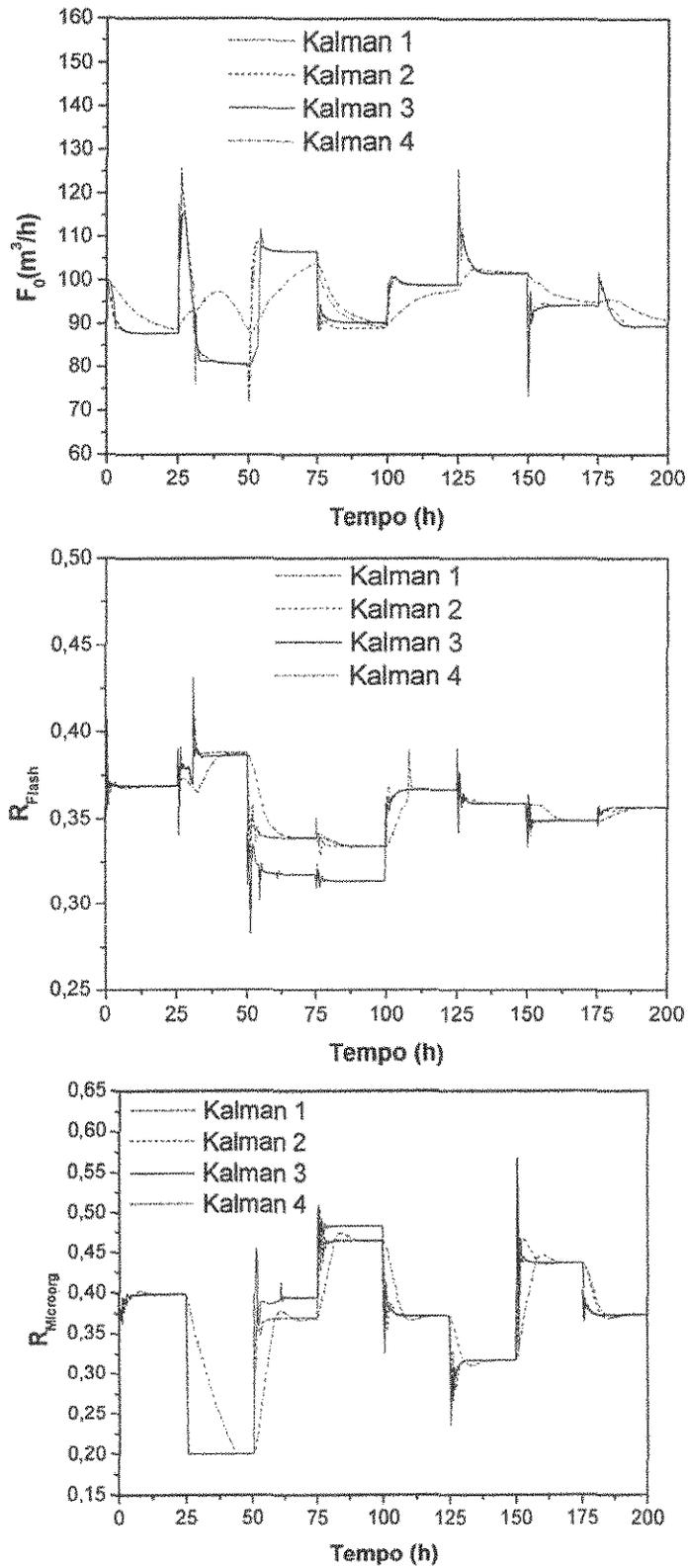


Figura 5.21 Variáveis Manipuladas para Problema Servo X e Regulador

Com a análise da Figura 5.10, que apresenta os resultados para as variáveis controladas para o problema do tipo servo em S, se observa que o algoritmo Kalman 2 e Kalman 4 se sobrepõem para todas as variáveis controladas e que apresentam os melhores resultados quando comparados com os algoritmos Kalman 1 e 3, que não possuem o fator de velocidade associado.

Para o problema do tipo servo em P, todos os algoritmos apresentaram uma boa performance dos controladores com respostas muito próximas, conforme ilustrado nas Figuras 5.12 e 5.13 que apresentam as respostas para as variáveis controladas e manipuladas, respectivamente. Como já discutido anteriormente de forma similar para o problema do tipo servo em X, os algoritmos apresentaram praticamente os mesmos resultados.

Nas Figuras 5.14 e 5.15 pode se observar que os algoritmos Kalman 2, 3 e 4 apresentaram os resultados muito próximos tanto para as variáveis controladas como manipuladas para problema do tipo regulador. Os resultados para os algoritmos Kalman 2, 3 e 4 foram bons e muito próximos para todas as variáveis controladas (S, P e X), para Kalman 1. Neste caso os resultados apresentaram muitos picos, comportamento similar ao controlador preditivo DMC.

Para problema do tipo regulador e servo em S os algoritmos Kalman 2 e 4 que possuem o fator de velocidade associado apresentaram os melhores resultados para todas as variáveis. Os algoritmos que não possuem esse fator de velocidade apresentaram resultados com muitos picos e não atingiram o *setpoint* para algumas perturbações como visualizado na Figura 5.16. Nota-se que o algoritmo Kalman 3, apesar da presença de picos, apresentou uma performance superior ao algoritmo Kalman 1 que possui o ganho constante ao longo do tempo.

Para problema do tipo regulador e servo em P os algoritmos Kalman 2 e 3 apresentaram os melhores resultados e bem semelhantes. O algoritmo Kalman 4 foi superior ao Kalman 1 que apresentou muitos picos para a variável controlada S. Para as variáveis controladas P e X os algoritmos Kalman 2, 3 e 4 apresentaram bons resultados com exceção para o intervalo de tempo 25-50 (h) devido às perturbações impostas ser

muito próximas dos limites de operação inferior para S_o e superior para T_o . O algoritmo Kalman 1 apresentou muitos picos e resultados inferiores, conforme mostra a Figura 5.18.

Para problema do tipo regulador e servo em X os algoritmos Kalman 2, 3 e 4 apresentaram resultados próximos e tiveram uma boa performance. O algoritmo Kalman 1, que tem o ganho constante sem o fator de velocidade, apresentou resultados com a presença de muitos picos e não atingiu o *setpoint* para muitas perturbações, como pode ser visualizado na Figura 5.20, que apresenta os resultados para as variáveis controladas do problema do tipo servo em X e regulador.

De forma similar aos controladores preditivos DMC e MPC Neural, o Condeg Modificado com seus diferentes algoritmos de ajuste da penalização de ação de controle (Kalman 1, 2, 3 e 4) tiveram resultados insatisfatórios das variáveis P para os tempos entre 25 e 50 horas, devido às perturbações impostas serem muito rigorosas. Isso mostra que a importância de controlar também uma dessas variáveis ao longo do tempo. De fato, em situações extremas, onde ocorre a diminuição da quantidade de substrato ao limite mínimo e aumento da temperatura ao limite máximo de operação, o controlador não consegue manter o *setpoint* para essas perturbações associadas a problema do tipo servo.

A Figura 5.22 apresenta a variação dos parâmetros de penalização da ação de controle para as variáveis controladas S, P e X (λ_1 , λ_2 e λ_3) para o algoritmo do Filtro de Kalman 2, que possui associado o fator de velocidade θ , onde foram avaliados diferentes valores de θ , sendo estes $\theta = 0,1$; $\theta = 0,5$ e $\theta = 0,9$ para o problema do tipo servo em P.

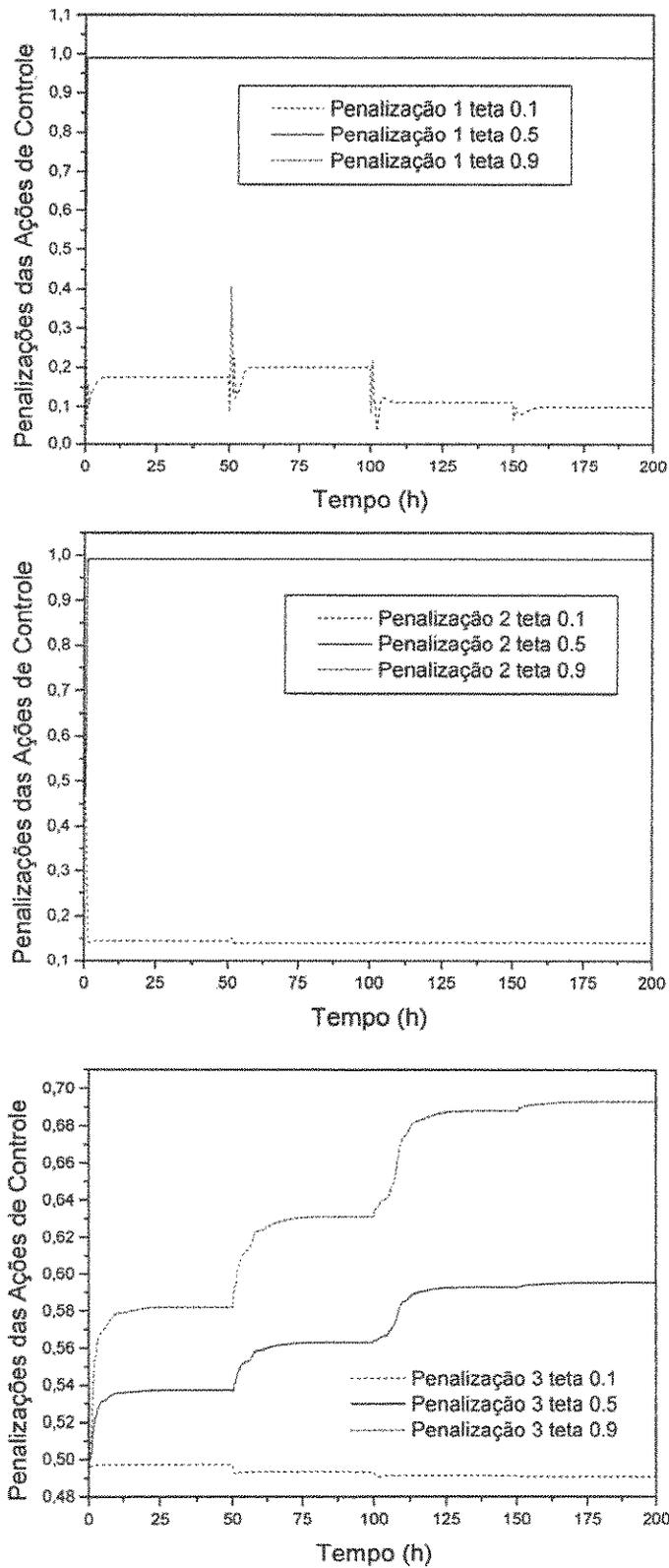


Figura 5.22 Penalizações das Ações de Controle para Diferentes Valores de θ

Através destes resultados, pode-se observar a importância deste parâmetro ser ajustado ao longo do tempo e não ser fixo. Como mostrado na Figura 5.22 os valores da penalização para as três ações de controle foi variando conforme as perturbações eram dadas ao processo. Observa-se também que para valores de teta menores ($\theta = 0,1$), ou seja, uma menor velocidade associada ocorre uma mudança mais suave e numa faixa menor fazendo com que os resultados sejam melhores com esta variação mais lenta, conforme discutido anteriormente. Para valores de $\theta = 0,5$ e $\theta = 0,9$ os resultados foram similares para as penalizações das ações de controle em S e P (λ_1 e λ_2) e para λ_3 os valores foram todos diferenciados conforme visualizado na Figura 5.22.

5.3 RESULTADO COMPARATIVO DOS ALGORITMOS PROPOSTOS

Neste item serão avaliados os algoritmos do tipo preditivo DMC, MPC Neural e Condeg Modificado com Filtro de Kalman que possui em seu algoritmo o ganho variável e associado a um fator de velocidade (Kalman 4), sendo considerada aprendizagem *on-line* para as redes e o fator de velocidade $\theta = 0,1$. As Figuras 5.23, 5.24 e 5.25 apresentam os resultados para a variável controlada S, para os problemas do tipo servo em S, regulador e regulador e servo em S, respectivamente.

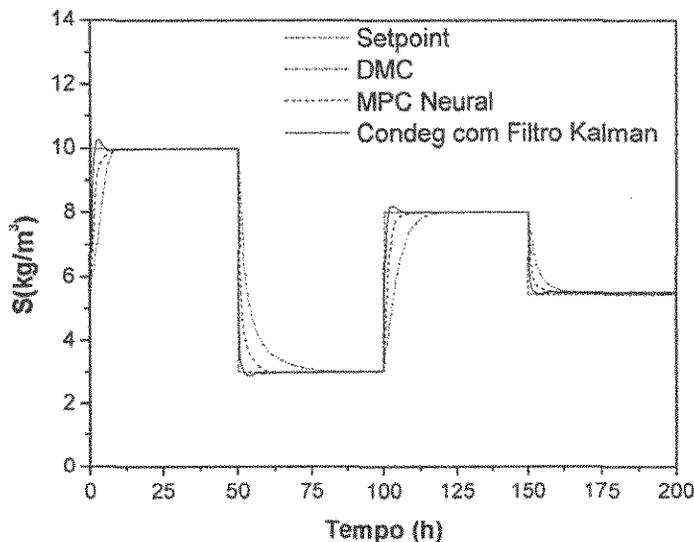


Figura 5.23 Resultado Comparativo para Problema Servo em P

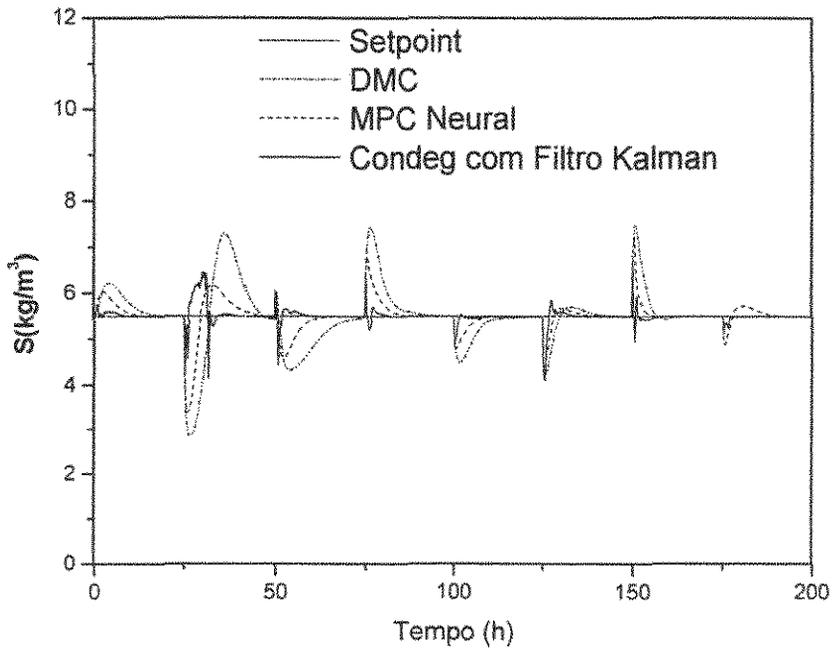


Figura 5.24 Resultado Comparativo para Problema Regulador

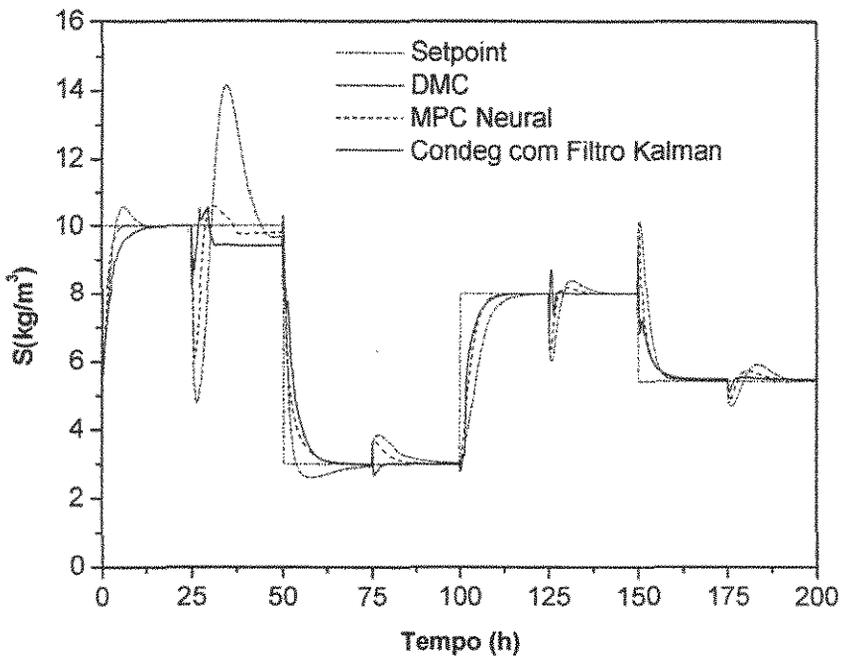


Figura 5.25 Resultado Comparativo para Problema Regulador e Servo em P

Comparando os algoritmos se pode visualizar que para todos os problemas o controlador que apresentou uma resposta mais lenta e com maior quantidade de picos foi o DMC. O controlador MPC Neural apresentou uma resposta intermediária entre os outros

dois algoritmos e que o algoritmo proposto Condeg com filtro de Kalman apresentou uma melhor performance, especialmente para problemas do tipo regulador e servo e reguladores aplicados de forma simultânea.

5 CONCLUSÕES

O capítulo 5 apresentou os resultados para cada algoritmo de forma individual e também um comparativo entre eles. Pode-se observar que os algoritmos que possuíram redes com aprendizagem *on-line* tiveram resultados superiores e sem a presença de *off-set* quando comparados com redes sem a aprendizagem *on-line*.

Com o comparativo entre os algoritmos pode-se observar que o algoritmo proposto Condeg Modificado com Filtro de Kalman possui uma ótima performance para todos os problemas estudados.

CONCLUSÕES E SUGESTÕES

Este capítulo apresenta as conclusões obtidas e sugestões para trabalhos futuros diante do que foi exposto ao longo dos outros capítulos, evidenciando a eficiência do uso de um algoritmo que permita ao controlador se adaptar ao longo do tempo através de uma rotina de adaptação *on-line* de seus parâmetros.

6.1 CONCLUSÕES

Este trabalho apresentou o desenvolvimento e avaliação de algoritmos de controle aplicados a um processo extrativo de fermentação alcóolica contínua. Este é um processo com uma dinâmica complexa inerente ao metabolismo dos microrganismos. Foram estudados problemas do tipo servo, regulador e servo e regulador aplicados de forma simultânea, considerando sempre problemas do tipo MIMO 3x3. Todos os algoritmos foram implementados em Linguagem Fortran 90.

Para o estudo sobre algoritmos de controle do tipo preditivo, onde foram considerados dois algoritmos: controle por matriz dinâmica (DMC) e o controle baseado em modelo não-linear, onde o modelo não-linear é representado por uma rede neural artificial (MPC Neural). Para o MPC Neural foram comparados dois algoritmos, considerando a aprendizagem das redes ao longo do tempo (*on-line*) através de uma estratégia de aprendizagem que utiliza três redes paralelas e considerando redes treinadas apenas de maneira *off-line*.

Com os resultados obtidos pode-se verificar que os algoritmos que utilizaram as redes com aprendizagem *on-line* apresentaram uma melhor performance e ausência de *offset*, gerado em alguns problemas, comparados aos algoritmos de controle que utilizaram redes com aprendizagem *off-line*. Isto é especialmente observado para os casos de problemas do tipo reguladores, pois as redes com treinamento *off-line*, foram treinadas apenas com dados das variáveis manipuladas e controladas. Assim as variáveis do tipo carga não eram conhecidas pelas redes.

No estudo de redes neurais artificiais pode se observar a grande capacidade e seu potencial na representação e generalização de processos. As redes estudadas mostraram eficiência tanto no uso como modelos diretos (controlador neural) e indiretos (dinâmica do processo), por isso a grande aplicação na área de desenvolvimento de algoritmos de controle.

Também foi proposto um algoritmo de controle adaptativo, denominado Condeg Modificado com filtro de Kalman, baseado em redes neurais artificiais, também com aprendizagem *on-line*, onde foram utilizadas redes para atuarem como controlador e também para representarem a dinâmica do processo na etapa de otimização do controlador.

Além disto, para evitar variações bruscas das ações de controle foi adicionado um filtro de primeira ordem que determinava as penalizações das ações de controle. Para ajustar este parâmetro de penalização ao longo do tempo foi utilizado um filtro de Kalman e analisados quatro algoritmos diferenciados para o cálculo do ganho

Quando se utiliza o fator velocidade (θ) mesmo sendo necessário estimar um parâmetro a utilização do filtro de Kalman ainda reduz o número de parâmetros a serem determinados e permite uma adaptação do parâmetro de penalização (λ_i) ao longo do tempo.

Com os resultados obtidos se pode observar que o auto-ajuste deste parâmetro de penalização permite uma melhor performance do algoritmo proposto e que o uso do fator de velocidade se torna indispensável em alguns problemas do tipo servo e regulador que amenizam a ação de controle e tornam os resultados mais eficientes. Especialmente para problemas do tipo servo, o fator de velocidade associado com valor de 0,1 apresentou melhores resultados comparados a valores de 0,5 e 0,9 para este fator.

Comparando os resultados para problemas do tipo servo todos os algoritmos estudados, DMC, MPC Neural e Condeg apresentaram uma boa performance, com exceção do Condeg Kalman 1 e 3, que não possuíam o fator de velocidade associado, sendo que os algoritmos Condeg Kalman 2 e 4 com valores do fator de velocidade 0.1, apresentaram uma ótima performance.

Também foi observado para problemas do tipo preditivo oscilações e picos nas variáveis manipuladas, o que dificulta a implementação do controle devido a variações bruscas nas válvulas, o que dificilmente ocorre no controle adaptativo, que nestes casos apresenta um melhor desempenho.

Para problemas do tipo regulador o algoritmo que apresentou melhores resultados foi o Condeg com todas as variações de Kalman (1, 2, 3 e 4). O algoritmo MPC Neural com aprendizagem *on-line* apresentou resultados intermediários entre o Condeg e o DMC que apresentou os piores resultados, com a presença de muitos picos e instabilidade nas variáveis manipuladas.

Para problemas do tipo servo e regulador novamente o controlador DMC apresentou resultados inferiores aos outros dois algoritmos, sendo que para o Condeg e MPC Neural para problemas do tipo regulador e servo em S os resultados foram similares, e para problemas dos tipos reguladores e servos em P e X, o algoritmo Condeg apresentou melhores resultados.

Para todos os algoritmos avaliados os resultados para o intervalo de tempo entre 25 e 50 horas foram feitas perturbações nas variáveis de carga (S_o e T_o) próximas do limite de operação, mínimo de substrato e máxima temperatura, o que tornou inviável o controle do processo para todos os algoritmos de controle, quando o problema era do tipo servo e regulador em P. Isto mostra que é necessário manter o controle de uma dessas variáveis quando se pretende manter uma alta produção de etanol.

O algoritmo Condeg com filtro de Kalman com fator de velocidade associado tanto com ganho constante como variável apresentaram resultados semelhantes e tiveram os resultados iguais ou superiores aos outros algoritmos.

Isto demonstra que o algoritmo proposto apresentou-se robusto para o processo estudado e permitiu que o controlador tivesse seus parâmetros atualizados ao longo do tempo, ou seja, permite um controle efetivo, pois consegue acompanhar a dinâmica do processo.

6.2 SUGESTÕES PARA TRABALHOS FUTUROS

Finalmente, para seqüência deste trabalho sugere-se como continuidade alguns trabalhos a serem explorados, tais como:

- Estudo avançado da arquitetura das redes neurais artificiais usadas, para determinar número de passados e neurônios da camada intermediária;
- Estudo e implementação de modelos híbridos nos algoritmos de controle;
- Implementação experimental do algoritmo Condeg com filtro de Kalman.

REFERÊNCIAS BIBLIOGRÁFICAS

AGÊNCIA-UDOP Presidente Lula enaltece o combustível. Disponível em: www.jornalcana.com.br, Acesso em maio de 2004.

AGÊNCIA – BRASIL Governo planeja destinar R\$500 milhões para estocagem do álcool. Disponível em <http://www.agenciabrasil.gov.br>. Acesso em 2004.

AIUBE, F., A., L., Modelagem dos preços futuros de commodities: abordagem pelo filtro de partículas, Rio de Janeiro. PUC-Rio Tese, 2005.

AL-ALAWI, A. AL-ALAWI, S. M. ISLAM, S. M. Predictive control of an integrated PV-diesel water and power supply system using an artificial neural network. *Renewable Energy*, v. 32, p. 1426-1439, 2007.

ALVES, J. G. L. F. Estudo da influência da temperatura na cinética de crescimento anaeróbico de *Saccharomyces cerevisiae*. Campinas. Dissertação (Mestre em engenharia de alimentos), Universidade Estadual de Campinas, 1996.

ANDRIETA, S. R. e MAUGERI FILHO, F. *Advances in Bioprocess Engineering*, Kluwer Academic Publishers. Optimum Design of a Continuous Fermentation Unit of an Industrial Plant for Alcohol Production, p 47-52, 1994.

ANDRIETTA, S. R. Modelagem, simulação e controle de fermentação alcoólica em escala industrial. Tese de Doutorado: Faculdade de Engenharia de Alimentos, Universidade Estadual de Campinas, 1994.

ANDRIETA, S. R. e STUPIELLO, J. P. Simulação e modelagem para processos de fermentação alcoólica (I) batelada alimentada, STAB, Açúcar, álcool e subprodutos. V. 8, p. 36-40, 1990a.

ANDRIETA, S. R. e STUPIELLO, J.P.. Simulação e modelagem para processos de fermentação alcoólica (II) contínua, STAB, Açúcar, álcool e subprodutos. V. 9, p. 45-51, 1990b.

ASSIS, A. JOSÉ de. Identificação e Controle de Processos Não-Lineares Utilizando Redes Neurais Artificiais. Campinas: Universidade Estadual de Campinas. Tese (Doutorado), 2004.

ATALA, D. I. P. Montagem, Instrumentação, Controle e Desenvolvimento Experimental de um Processo de Fermentação Alcoólica Extrativo. Campinas - SP., 2004. Tese (Doutor em Engenharia de Alimentos). Faculdade de Engenharia de Alimentos, Universidade Estadual de Campinas.

ATALA, D. I. P. COSTA, A. C., MACIEL FILHO, R. MAUGERI, F. Kinetics of Ethanol Fermentation with High Biomass Concentration Considering the Effect of Temperature. *Applied Biochemistry and Biotechnology*, v. 91, p. 353-365, 2001.

BALLINI, R., ZUBEN, F. J. VON. Book reviews: Application of neural networks to adaptive control of nonlinear systems. *Automatica*, v. 36, p. 1929-1933, 2000.

BAUGHMAN, D. R. LIU, Y. A. *Neural Networks in Bioprocessing and Chemical Engineering*, London: Academic Press, 1995.

BAKSHI, B. R. STEPHANOPOULOS, G. Wave-net: A Multiresolution, Hierarchical Neural Network with Localized Learning. *AIChE Journal*, v. 39-1, 1993.

BENNE, M. GRONDING-PEREZ, B., CHABRIAT, J.P, HERVÉ, P. Artificial neural networks for modeling and predictive control an industrial evaporation process. *Journal of Food Eng.* v. 46, p. 227-334, 2000.

BEQUETTI, B. W. Non linear control of chemical processes: a review. *Industrial Chem. Eng. Research.* V.30, p. 1391-1431, 1991.

BOSKOVIC, J. D. e NARRENDRA, K. S. Comparison of linear, nonlinear and neural network based adaptive controllers for a class of feed-back fermentation processes. *Automatica.* V. 31, n.6, p. 817-840, 1995.

BRAGA, A. de Pádua, LUDEMIR, T. B., CARVALHO A.C. P. L. *Redes Neurais Artificiais – Teoria e Aplicações.* Rio de Janeiro: LTC, 2000.

BRAAKE, H. A. B., CAN, E. J. L., SCHERPEN, J. M. A., VERBRUGGEN, H. B. Control nonlinear chemical processes using neural models and feedback linearization. *Computers and Chem. Eng.* v.22, n.7-8, p. 1113-1127,1998.

BRODERICK, G., BUDMAN, H. M., KNAPP, T. D. Adaptive Control of a CSTR with a Neural Network Model. *Journal of Process Control*, v.11, p.53-63, October, 1999.

BULSARI, A. B. *Neural Networks for Chemical Engineers*. Amsterdam: Elsevier, 1995.

CARNICELLI M. Álcool volta á tona com motor bi-combustível. Disponível em : http://carsale.uol.com.br/opapoecarro/mercado/mercado_190304.shtml. Acesso em março 2004.

CHEN, F. C. Backpropagation Neural Networks for Nonlinear Self- Tuning Adaptive Control. *IEEE Control Systems Magazine*, v.10-3, p.44-48, 1990.

CHEN, S., BILLINGS, S. A. Neural Networks for Nonlinear Dynamic System Modeling and Identification. *International Journal Control*, v.56-2, p. 319-346, 1992.

CLARKE, D. W., MOTHTADI, C. e TUFFS, P. S. Generalized predictive control. 1. The basic algorithm, *Automatica*, v. 23, p. 137-148, 1987.

CONSTACE PAUL. A era do etanol? *Revista do Banco Interamericano de Desenvolvimento, BIDAMÉRICA*. Publicado em 16 de abril de 2007. Disponível em: <http://www.iadb.org/idbamerica/index.cfm?thisid=4259>, acesso em Março de 2007.

COSTA, A. C., ATALA, D. I. P., MACIEL FILHO, R. MAUGERI FILHO, F. Factorial Design and simulation for the Optimization and Determination of Control Structures for na Extractive Alcoholic Fermentation. *Process Biochemistry*, v. 37-2, p. 125-137, 2001.

COSTA, A. C., MELEIRO, L. A. C., MACIEL FILHO, R. Non linear Predictive Control of na Extractive Alcoholic Fermentation Process. *Process Biochemistry*, v. 38-5, p. 743-750, 2002.

CUNHA, R. D. F. Financiamento da estocagem do álcool deve ser prorrogado Ministério da Agricultura Pecuária e Abastecimento, Brasília, 28 de janeiro: Indústria, 2004.

CUTLER, C. R. DMC of imbalanced Susténs, ISA Trans., v. 21, n.1 p.1, 1982.

CUTLER, C. R. e RAMAKER, B. L. Dynamic matrix control – A computer control algorithm, AICHE 86th National Meeting, Texas, 1979.

CYBENKO, G. Approximation by Superposition of a Sigmoidal Function. Mathematics of Control, Signals and Systems. v. 2, p. 303-314, 1989.

DARSONO S. e LABADIE, J. W. Neural optimal control algorithm for real time regulation of in-line storage in combined sewer systems. Environmental Modelling e Software. P. 1-13, 2006.

DE SOUZA JÚNIOR, M. B. D. Redes Neurais Multicamadas Aplicadas a Modelagem e Controle de Processos Químicos. Rio de Janeiro: Universidade Estadual do Rio de Janeiro, 1993. Tese (Doutorado).

DUARTE, Elis Regina. Estratégias de Controle Não Linear Baseada em Redes Neurais Artificiais com Aprendizagem “On-Line”. Campinas: Universidade Estadual de Campinas, 2004. Dissertação de Mestrado.

DUARTE, Elis Regina, ENDER, Laércio, MACIEL FILHO, Rubens. A Methodology for On-Line Learning of Neural Networks in Control Strategy Development. Proceedings of 16th International Congress of Chemical and Process Engineering, Pragua, 2004a.

DUARTE, E. R., MACIEL FILHO, R., ENDER, L. Parameter Adjustment Using Kalman Filter of Advanced Control Strategy Applied to Fermentation Process In: 16th INTERNATIONAL CONGRESS OF CHEMICAL AND PROCESS ENGINEERING, Prague, 2004b.

DUARTE, Elis Regina, ENDER, Laércio, MACIEL FILHO, Rubens. Evaluation of Alternative Configurations for On-line Learning in Control Strategies Development. Proceedings of the Biannual Brazilian Symposium on Artificial Neural Networks (SBRN), Brasil, 2004c.

DUARTE, Elis. Regina, MACIEL FILHO, Rubens., ENDER, Laércio. A Non-Linear Control Strategy of a Fixed Bed Catalytic Reactor for Bio-Ethanol Oxidation In: 28th Symposium on Biotechnology for Fuels and Chemicals, Nashville, 2006.

DUARTE, Elis Regina, ENDER, Laércio, ATALA, Daniel I. P., MACIEL FILHO, Rubens. Control of an Extractive Fermentation Process to Bio-Ethanol Production. Proceedings of 28th Symposium on Biotechnology for Fuels and Chemicals, Denver, 2007.

DURBIN, J. e KOOPMAN, S. J. Time Series Analysis by State Space Methods. Oxford University Press, 2002.

EBERHART, R., DOBBINS, R. Neural Networks PC Tools – A Practical Guide. Academic Press, San Diego, 1990.

ENCARNAÇÃO JÚNIOR, G. Álcool Revisitado. Economia e Energia. v.34, 2002.

ENDER, L., MACIEL FILHO, R.. A Robust Procedure for On-line Learning of Neural Networks in Control Strategy Development. Proceedings of 3rd European Congress of Chemical Engineering. Germany, 2001a.

ENDER, L., MACIEL FILHO, R. A Robust Procedure for On-line Learning of Neural Networks in Control Strategy Development. Part II. Proceedings of 6th World Congress of Chemical Engineering. Melbourne, Austrália, 2001b.

ENDER, L. Redes Neurais Aplicadas em Estratégias de Controle não Linear. Campinas: Universidade Estadual de Campinas, 2002. Tese (Doutorado).

ENDER, L.; SCHEFFER, R.; MACIEL FILHO, R.. Computer design of a new predictive adaptive controller coupling neural networks and Kalman filter applied to

siso and mimo control. In: Johan Grievink; Jan van Schijndel. (Org.). Computer-Aided Chemical Engineering - European Symposium on Computer Aided Process Engineering-12. Amsterdam: Elsevier, 2002, v. 10, p. 481-486.

ENDER, L.; SCHEFFER, R.; MACIEL FILHO, R. Computer Design of a new advanced Control Parameter Adjustment using Kalman Filter. In: Sauro Pierucci. (Org.). Chemical Engineering Transactions - 6th Italian Conference on Chemical and Process Engineering. 3 ed. Pisa: AIDIC - Associazione Italiana Di Ingegneria Chimica, 2003, v. 3, p. 1197-1202.

FURFERI, R. CARFAGNI, M. DAOU, M. Artificial neural network software for real-time estimation of olive oil qualitative parameters during continuous extraction. Computers and Electronics in Agriculture, v. 55, p. 115-131, 2007.

GADKAR, K. G. MEHRA, S. GOMES, J. On-line adaptation of neural networks for bioprocess control. Computers of Chem. Eng., v. 29, p.1047-1057, 2005

GIARRATANO, J. C., CULBERT, C. and SAVELY, R. T., The State of the Art for Current and Future Expert Systems Tools, ISA Transactions, 29, 1, 17-25, 1990.

GOZDZIEJEWSKI JR. R., ENDER, L. Desenvolvimento de Software Didático em Redes Neurais Artificiais. Anais do Congresso Brasileiro de Engenharia Química, Porto Alegre, 1984.

GLOBO NOTÍCIAS. Disponível em: <http://g1.globo.com/Noticias/Negocios/0,,MUL32493-5600-1008.00.html>. Acesso em Maio de 2007.

HANG, C. C., GE, S. S., ZHANG, T. Design and Performance Analysis of a Direct Adaptive Controller for Nonlinear Systems. Automatica, v.35, p. 1809-1817, March, 1999.

HARVEY, A. C. Forecasting, structural time series models and the Kalman Filter. Cambridge: Cambridge University Press, 1989.

HEBB, D. O. The organization of the Behavior: A Neuropsychological Theory,. New York: Wiley, 1949.

HECHT-NIELSEN R., Theory of the Backpropagation Neural Network, IEEE int. Conf. on Neural Network, v. 1, p.593-605, 1989.

HECHT-NIELSEN R., Neurocomputing, Addison-Wesley Publishing Company, Reading, MA, 1990.

HENRIQUE, H. M., LIMA, E. L. , SEBORG, D. E. Model Structure Determination in Neural Network Models, Chemical Engineering Science, v. 55-22 p. 5457-5469, 2000.

HENSON, M. A. Nonlinear model predictive control: current status and future directions. Computers and Chemical Eng. v.23, p. 187-202, 1998.

HENSON, M. A. e SEBORG, D. E. Nonlinear Process Control, Prentice Hall, 1997.

HOPFIELD, J.J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities, Proceedings of the National Academy of Sciences of the USA, v.79, p. 2554-2558, 1982.

HORNIK, K., STINCHCOMBE, M., WHITE, H. Multi-layered Feedforward Neural Networks are Universal Approximations. Neural Networks, v.2, p. 359-366, 1990.

HOSKINS, J. C. and HIMMELBLAU, D. M., Artificial Neural Network Models of Knowledge Representation in Chemical Engineering, Comp. Chemical Engineering., 12, 9/10, 881-890, 1988.

HUNG, L., CHUNG, H. Decoupled control using neural network base sliding mode controller for nonlinear systems. Expert Systems with Applications, v. 32, p.1168-1182, 2007.

HUNT, K. J., SBARBARO, D., ZBIKOWSKI, R., GAWATHROP, P. J. Neural Networks in Control Systems: A survey, Automatica, v. 28, n.6, p. 1083-1112, 1992.

HUSSAIN M. A. Review of the applications of neural networks in chemical process control – simulation and online implementation. *Artificial Intelligence in Engineering*, v.13, p.55-68. June, 1999.

IANET – Inteligência Artificial. Versão 3.8. <http://www.lokall.com/ianet/rdneuro.htm>.
Capturado em 14/08/2003.

JILSON, K. R., YDSTIE, B. E. Process networks with decentralized inventory and flow control. *Journal of Process Control*, v. 17, p. 399-413, 2007.

JUNIOR BITENCOURT, H. TORRES, L. A. B., AGUIERRE, L. A. O filtro de kalman para sistemas não-lineares. Disponível em www.cph.eng.ufmg.br/docscph/mateevento28.pdf. Acesso em Maio de 2007.

KADI EL HANY. Modeling the mathematical behavior of fiber-reinforced polymeric composite materials using artificial neural networks – A review. *Composite Structures*, v. 73, p. 1-23, 2006.

KAUFMANN P., ANDERSSON G. G. Development of a generalized neural network. *Chemo metrics and Intelligent Laboratory Systems*, v. 50, p. 101-105, 2000.

KALMAN, R.E. *Journal of basic engineering, Transactions of ASME, series D*, v. 82, n. 1, pp 35-45, “A new approach to linear filtering and prediction problems”, 1960.

KOVÁCS, Z. L. *Redes Neurais Artificiais – Fundamentos e Aplicações*. São Paulo: Edição Acadêmica, 1996.

KOLMOGOROV, A. N. On the Representation of Continuous Functions. *URSS: Dokl. Acad. Nauk*, 1957.

LEE, M., PARK,S. A new scheme combining neural feedforward control with model-predictive control. *Aiche Journal*, v. 38, n.2, p. 193-200, 1992.

LEWIS, F.L. *Optimal Estimation*, John Wiley & Sons, New York, NY, 1986.

LIEW, A. C., WANG, F., CHANG, C. S. Simulated Hardware Design of Artificial Neural Networks for Adaptive Plant Control. *Electric Power Systems Research*, v.37, p. 231-240, March, 1996.

LOESCH, C., SARI, S. T. *Redes Neurais: Fundamentos e Modelos*. Blumenau: Universidade Regional de Blumenau, 1996.

MACEDO, ISAIAS. Greenhouse gas emissions and energy balances in bio-ethanol production and utilization in Brazil (1996). *Biomass and Bioenergy*, v. 14-1, p. 77-81, 1997.

MARCHETTI, J. L., MELLICHAMP, D. A, SEBORG, D. E. Predictive control based on discrete convolution models. *Ind. Eng. Chem. Proc.Des. Dev.*, v. 22, p. 488-495, 1983.

MAURATH, P. R., LAUB, A. J., SEBORG, D. E., MELLICHAMP, D. A. Predictive controller design by principal components analysis. *Ind. Eng. Chem. Res. V. 27*, p. 1204- 1212, 1988.

McCULLOCH, W.S. e W. PITTS. A Lógica Calculus of the Ideas Immanent in Neurons Activity. *Bulletin of Mathematical Biophysics*, v.5, p. 115, 1943.

MINSKY, M. L., PAPERT, S. A. *Perceptrons*. MIT Press, Cambridge, MA, 1969.

MELEIRO, L. A. C., *Projeto e Aplicação de Controladores Baseados em Modelos Lineares, Neurais e Nebulosos*. São Paulo: Universidade Estadual de Campinas, 2002. Tese (Doutorado).

MORARI, M., LEE, J. H. Model predictive control: past, present and future. *Computers and Chemical Eng.* v.23, 667-682, 1999.

MOREIRA, J. R. e GOLDEMBERG, J. O programa de álcool. 2002). Disponível em: Em: <http://www.mct.gov.br/clima/comunicold/alcohol.html>, acesso em Maio de 2002.

De acordo com Meleiro (2002) o modelo utilizado pelo DMC geralmente é obtido a partir da aplicação de um degrau em cada variável de entrada do processo, seguido da correspondente medição das respostas das variáveis de saída. Estas respostas ao degrau são utilizadas para identificar o modelo de convolução do processo. Através do modelo de resposta ao degrau, podem-se prever as mudanças futuras das saídas do processo como uma combinação linear das variações nas entradas.

Em cada instante de amostragem, o valor da resposta ao degrau da variável de saída é medido e chamado de coeficiente de resposta ao degrau a_i . A diferença entre dois coeficientes de resposta ao degrau sucessivo é chamado de coeficiente de resposta ao impulso e é dado pela equação (1.1) (Newell e Lee, 1989):

$$h_i = a_i - a_{i-1} \quad (1.1)$$

O valor predito da variável de saída \hat{y}_k , no instante de amostragem k , pode ser expresso de acordo com a equação (1.2):

$$\hat{y}_k = \sum_{i=1}^{Nm} h_i u_{k-i} \quad (1.2)$$

O algoritmo DMC para sistemas mono-variáveis (*Single input single output -SISO*) utiliza o modelo descrito pela equação (1.2). Reescrevendo-a para o instante de amostragem $k+1$, tem-se a equação (1.3):

$$\hat{y}_{k+1} = \sum_{i=1}^{Nm} h_i u_{k+1-i} \quad (1.3)$$

Subtraindo a equação (1.2) da equação (1.3), obtém-se:

$$\hat{y}_{k+1} - \hat{y}_k = \sum_{i=1}^{Nm} h_i (u_{k+1-i} - u_{k-i}) \quad (1.1)$$

Definindo $\Delta u_k = u_k - u_{k-1}$, a equação (1.1) pode ser reescrita:

$$\hat{y}_{k+1} = \hat{y}_k + \sum_{i=1}^{Nm} h_i \Delta u_{k+1-i} \quad (1.5)$$

O modelo de convolução representado na equação (1.5) pode ser estendido para considerar um horizonte de predição de N_y instantes futuros:

$$\hat{y}_{k+j} = \hat{y}_{k+j-1} + \sum_{i=1}^{Nm} h_i \Delta u_{k+1-i} \quad \text{para } j=1, \dots, N_y \quad (1.6)$$

Para corrigir os erros de predição do modelo dado pela equação (1.6) o algoritmo utiliza uma estratégia de realimentação, onde o valor predito da saída do processo no instante de amostragem k é comparado como valor medido desta mesma variável. O desvio calculado entre os dois valores é utilizado para corrigir as futuras predições.

Por exemplo, no instante de amostragem $k+1$, tem-se:

$$y^c_{k+1} = \hat{y}_{k+1} + (y_k - \hat{y}_k) \quad (1.7)$$

Onde y_k é o valor medido da variável de saída no instante de amostragem k e y^c_{k+1} é o valor predito da mesma variável para o instante de amostragem $k+1$.

Generalizando para os N_y instantes futuros, tem se a equação (1.8):

$$y^c_{k+j} = \hat{y}_{k+j} + (y^c_{k+j-1} - \hat{y}_{k+j-1}) \quad \text{para } j=1, \dots, N_y \quad (1.8)$$

Para o instante $k+j$ com $j=1$, verifica-se que o termo y^c_{k+j-1} da equação (1.8) corresponde ao termo y_k da equação (1.7), ou seja, o valor medido da saída. Para $j=2$, a equação (1.8) fica:

$$y^c_{k+2} = \hat{y}_{k+2} + (y^c_{k+1} - \hat{y}_{k+1})$$

$$\begin{aligned}
 &= \hat{y}_{k+2} + (\hat{y}_{k+1} + (y_k - \hat{y}_k)) - \hat{y}_{k+1} \\
 &= \hat{y}_{k+2} + (y_k - \hat{y}_k)
 \end{aligned} \tag{1.9}$$

Para o instante $k+j$ com $j=N_y$, obtém-se a equação (1.10):

$$y^c_{k+N_y} = \hat{y}_{k+N_y} + (u_k - \hat{u}_k) \tag{1.10}$$

Pode-se observar que as previsões para os instantes futuros de $k+1$ a $k+N_y$, feitas no instante atual k , são corrigidas como mesmo valor, dado pela diferença entre o valor medido e o valor predito da saída do processo no instante de amostragem atual k . Combinando as equações (1.6) e (1.8), chegando a equação (1.11):

$$y^c_{k+j} = y^c_{k+j-1} + \sum_{i=1}^{Nm} h_i \Delta u_{k+j-1} \quad \text{para } j=1, \dots, N_y \tag{1.11}$$

Para o instante $k+j$ com $j=1$, a equação (1.11) fica:

$$\begin{aligned}
 y^c_{k+j} &= y^c_k + \sum_{i=1}^{Nm} h_i \Delta u_{k+1-i} \\
 &= y_k + \sum_{i=1}^{Nm} h_i \Delta u_{k+1-i} \\
 &= y_k + h_1 \Delta u_k + h_2 \Delta u_{k-1} + \dots + h_{N_m} \Delta u_{k+1-N_m}
 \end{aligned} \tag{1.12}$$

Uma vez que os valores de Δu_{k-1} , Δu_{k-2} , ..., Δu_{k-N_m} são conhecidos, pode-se agrupá-los em um único termo conforme a equação (1.13):

$$S_i = \sum_{i=2}^{Nm} h_i \Delta u_{k+1-i} \tag{1.13}$$

que, por sua vez, pode ser utilizado para reescrever a equação (1.12):

$$y^c_{k+1} = y_k + h_1 \Delta u_k + S_1 \quad (1.11)$$

Repetindo-se o processo para o instante $k+j$ com $j=2$, obtém-se a equação (1.15):

$$y^c_{k+2} = y^c_{k+1} + h_1 \Delta u_{k+1} + h_2 \Delta u_k + \dots + h_{N_m} \Delta u_{k+2-N_m} \quad (1.15)$$

Agrupando novamente os valores conhecidos em um único termo:

$$y^c_{k+2} = y^c_{k+1} + h_1 \Delta u_{k+1} + h_2 \Delta u_k + S_2 \quad (1.16)$$

Onde $S_2 = \sum_{i=3}^{N_m} h_i \Delta u_{k+2-i}$, substituindo a equação (1.11) na equação (1.16), obtém-se:

$$y^c_{k+2} = y_k + (h_1 + h_2) \Delta u_k + h_1 \Delta u_{k+1} + S_1 + S_2 \quad (1.17)$$

Definindo-se:

$$a_i = \sum_{j=1}^i h_j \quad \text{para } i=1, \dots, N_y \quad (1.18)$$

$$S_m = \sum_{i=m+1}^{N_m} h_i \Delta u_{k+m-i} \quad \text{para } m=1, \dots, N_y \quad (1.19)$$

$$P_i = \sum_{m=1}^i S_m \quad \text{para } i=1, \dots, N_y \quad (1.20)$$

Pode-se escrever para o instante $k+j$:

$$y^c_{k+j} = y_k + a_j \Delta u_k + a_{j-1} \Delta u_{k+1} + \dots + a_1 \Delta u_{k+j-1} + P_j \quad (1.21)$$

A equação (1.21) pode ser reescrita na forma matricial, conforme a equação (1.22):

$$\begin{bmatrix} y^c_{k+1} \\ y^c_{k+2} \\ \vdots \\ y^c_{k+N_y-1} \\ y^c_{k+N_y} \end{bmatrix} = \begin{bmatrix} a_1 & 0 & \cdots & 0 & 0 \\ a_2 & a_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{N_y-1} & a_{N_y-2} & \cdots & a_1 & 0 \\ a_{N_y} & a_{N_y-1} & \cdots & a_2 & a_1 \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+N_y-2} \\ \Delta u_{k+N_y-1} \end{bmatrix} + \begin{bmatrix} y_k + P_1 \\ y_k + P_2 \\ \vdots \\ y_k + P_{N_y-1} \\ y_k + P_{N_y} \end{bmatrix} \quad (1.22)$$

Compactando, temos a equação (1.23):

$$(y^c)_{N_y \times 1} = (M)_{N_y \times N_y} (\Delta u)_{N_y \times 1} + (y^P)_{N_y \times 1} \quad (1.23)$$

Onde $M_{(N_y \times N_y)}$ é a matriz triangular e $y^P_{(N_y \times 1)}$ é o vetor de predições dados pela equação (1.22). Tais predições são feitas sem considerar ações de controle nos intervalos de amostragem k a $k+N_y$, ou seja, este vetor corresponde às predições feitas para o horizonte de predição levando-se em conta somente as ações de controle realizadas no passado (Meleiro, 2002):

Se definirmos a trajetória desejada y^d , igual ao valor desejado (y_{sp}), tem-se:

$$\begin{bmatrix} y^d_{k+1} \\ y^d_{k+2} \\ \vdots \\ y^d_{k+N_y-1} \\ y^d_{k+N_y} \end{bmatrix} = \begin{bmatrix} y_{spk} \\ y_{spk} \\ \vdots \\ y_{spk} \\ y_{spk} \end{bmatrix} \quad (1.21)$$

Subtraindo a equação (1.22) da equação (1.21) e definindo o erro ($e_k = y_{spk} - y_k$):

$$E = -M\Delta u + E' \quad (1.25)$$

onde

$$E = \begin{bmatrix} y^d_{k+1} - y^c_{k+1} \\ y^d_{k+2} - y^c_{k+2} \\ \vdots \\ y^d_{k+N_y-1} - y^c_{k+N_y-1} \\ y^d_{k+N_y} - y^c_{k+N_y} \end{bmatrix} \quad e \quad E' = \begin{bmatrix} e_k - P_1 \\ e_k - P_2 \\ \vdots \\ e_k - P_{N_y-1} \\ e_k - P_{N_y} \end{bmatrix} \quad (1.26)$$

Para que a saída predita seja igual a desejada $E=0$, e, conseqüentemente:

$$\Delta u = M^{-1} + E' \quad (1.27)$$

A solução fornecida pela equação (1.27) é única, sendo obtida a partir de um sistema determinado. Entretanto, esta solução é baseada na imposição de que a variável controlada seja igual à trajetória, desejada ao longo do horizonte de controle (N_u), o que muitas vezes é fisicamente impossível. O algoritmo DMC consiste em obter um sistema indeterminado, reduzindo arbitrariamente a dimensão do vetor Δu de acordo com o tamanho do horizonte de controle. Para isso, considera-se que $\Delta u_{k+j} = 0$ para $j \geq N_u$, sendo que $N_u \geq N_y$ (Pinto, 1990).

Com base nessas considerações, pode-se reescrever a equação (1.27):

$$\begin{bmatrix} y^c_{k+1} \\ y^c_{k+2} \\ \vdots \\ y^c_{k+N_y-1} \\ y^c_{k+N_y} \end{bmatrix} = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{N_y-1} & a_{N_y-2} & \cdots & a_{N_y-N_u} \\ a_{N_y} & a_{N_y-1} & \cdots & a_{N_y-N_u+1} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+N_u-2} \\ \Delta u_{k+N_u-1} \end{bmatrix} + \begin{bmatrix} y_k + P_1 \\ y_k + P_2 \\ \vdots \\ y_k + P_{N_y-1} \\ y_k + P_{N_y} \end{bmatrix} \quad (1.28)$$

Ou na forma compacta:

$$(y^c)_{N_y \times 1} = (A)_{N_y \times N_u} (\Delta u)_{N_u \times 1} + (y^p)_{N_y \times 1} \quad (1.29)$$

Na equação (1.29) a matriz $A_{(N_y \times N_u)}$, chamada de matriz dinâmica, relaciona as entradas com as saídas do processo e através dela se pode calcular analiticamente (através

da solução de um problema de mínimos quadrados) o vetor ótimo de ações de controle (Qin e Badgwell, 1997). Observa-se que a matriz dinâmica A é formada pelas primeiras N_u colunas da matriz M .

O sistema que não apresenta uma solução única deve ser resolvido segundo um critério de otimização. O algoritmo DMC calcula o vetor Δu para minimizar o seguinte critério de desempenho:

$$J = E^T E \quad (1.30)$$

cuja solução é dada por:

$$\Delta u = (A^T A)^{-1} A E^T \quad (1.31)$$

A equação (1.31) fornece uma solução que representa um problema de otimização sem restrição o que pode provocar ações de controle muito bruscas. Porém, isto pode ser evitado pela inclusão de um índice de desempenho modificado que incorpora restrições aos movimentos da variável manipulada, segundo a equação (1.32):

$$J = (E^T E) + (\Lambda \Delta u)^T (\Lambda \Delta u) \quad (1.32)$$

onde $\Lambda (N_u \times N_u)$ é uma matriz diagonal (matriz de ponderação) que permite introduzir penalizações nas ações de controle. Os elementos da matriz de ponderação são dados pela equação (1.33):

$$\lambda_{ij} = f \text{ se } i=j \quad e \quad \lambda_{ij} = 0 \text{ se } i \neq j \quad (1.33)$$

onde f é o fator de ponderação ou também chamado de fator de supressão.

A lei de controle resultante da função objetivo descrita pela equação (1.32) é dada por (Cutler e Ramaker, 1979), como mostrado na equação (1.31):

$$\Delta u = (A^T A + \Lambda^T \Lambda)^{-1} A^T E^T \quad (1.31)$$

$$\hat{y}_{k+1} = \hat{y}_k + \sum_{i=1}^{Nm} H_i u_{k+1-i} \quad (1.38)$$

O modelo de convolução dado pela equação (1.38) pode ser estendido para N_y instantes futuros, de acordo com a equação (1.39):

$$\hat{y}_{k+j} = \hat{y}_{k+j-1} + \sum_{i=1}^{Nm} H_i u_{k+j-i} \quad \text{para } j=1, \dots, N_y \quad (1.39)$$

Corrigindo as previsões de forma análoga ao procedimento descrito pela equação (1.8):

$$y^c_{k+j} = \hat{y}_{k+j} + (y^c_{k+j-1} - \hat{y}_{k+j-1}) \quad \text{para } j=1, \dots, N_y \quad (1.10)$$

onde, no instante $k+1$, y^c_{k+j-1} é igual ao valor medido y_k . Substituindo a equação (1.36) na equação (1.37), obtém-se:

$$y^c_{k+j} = y^c_{k+j-1} + \sum_{i=1}^{Nm} H_i \Delta u_{k+j-i} \quad \text{para } j=1, \dots, N_y \quad (1.11)$$

Repetindo o procedimento descrito para os sistemas SISO, obtém-se a forma MIMO da equação (1.28):

$$\begin{bmatrix} y^c_{k+1} \\ y^c_{k+2} \\ \vdots \\ y^c_{k+N_y-1} \\ y^c_{k+N_y} \end{bmatrix} = \begin{bmatrix} A_1 & 0 & \dots & 0 \\ A_2 & A_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_y-1} & A_{N_y-2} & \dots & A_{N_y-N_u} \\ A_{N_y} & A_{N_y-1} & \dots & A_{N_y-N_u+1} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+N_y-2} \\ \Delta u_{k+N_y-1} \end{bmatrix} + \begin{bmatrix} y_k + P_1 \\ y_k + P_2 \\ \vdots \\ y_k + P_{N_y-1} \\ y_k + P_{N_y} \end{bmatrix} \quad (1.12)$$

onde:

$$A_i = \sum_{j=1}^i H_j \quad \text{para } i=1, \dots, N_y \quad (1.13)$$

$$S_m = \sum_{i=m+1}^{N_y} H_i \Delta u_{k+m-i} \quad \text{para } m=1, \dots, N_y \quad (1.11)$$

$$P_i = \sum_{m=1}^i S_m \quad \text{para } i=1, \dots, N_y \quad (1.15)$$

Reescrevendo a equação (1.12) na forma compactar:

$$(y^c)_{N_y \times 1} = M_{N_y \times N_u} (\Delta u)_{N_u \times 1} + (y^p)_{N_y \times 1} \quad (1.16)$$

$$E = -M \Delta u + E' \quad (1.17)$$

contudo, M e Δu são definidos na equação (1.16) e E e E' são dados por:

$$E = \begin{bmatrix} y^d_{k+1} - y^c_{k+1} \\ y^d_{k+2} - y^c_{k+2} \\ \vdots \\ y^d_{k+N_y-1} - y^c_{k+N_y-1} \\ y^d_{k+N_y} - y^c_{k+N_y} \end{bmatrix} \quad \text{e} \quad E' = \begin{bmatrix} e_k - P_1 \\ e_k - P_2 \\ \vdots \\ e_k - P_{N_y-1} \\ e_k - P_{N_y} \end{bmatrix} \quad (1.18)$$

onde $e_k = y_{spk} - y_k$.

De acordo com Meleiro (2002) como no caso SISO, o algoritmo DMC calcula o vetor Δu para minimizar o seguinte critério de desempenho:

$$J = (E^T E) + (\Lambda \Delta u)^T (\Lambda \Delta u) \quad (1.19)$$

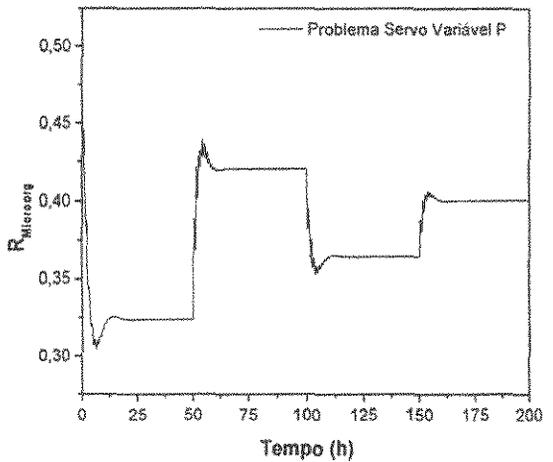
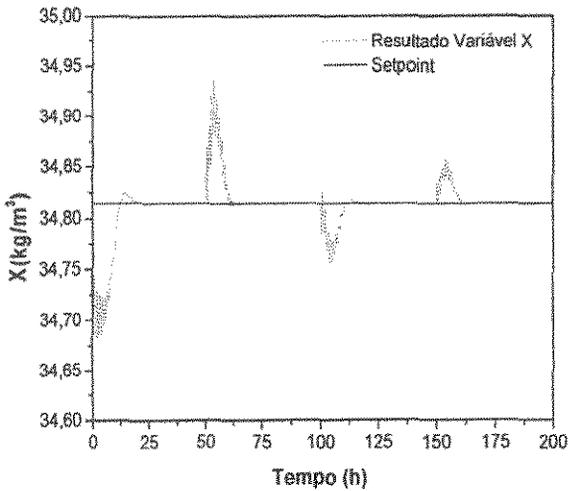
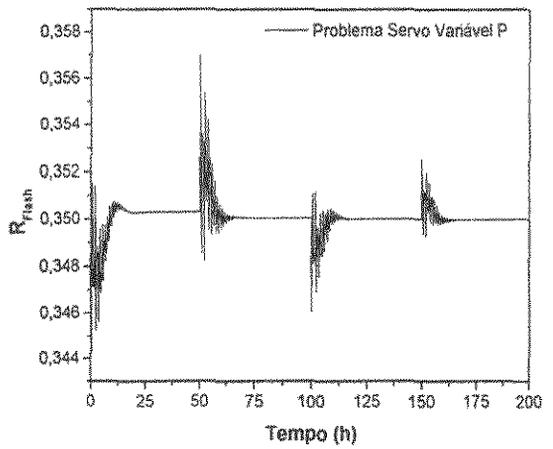
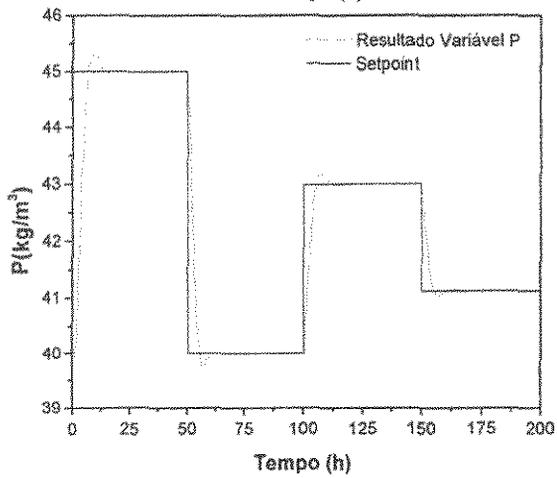
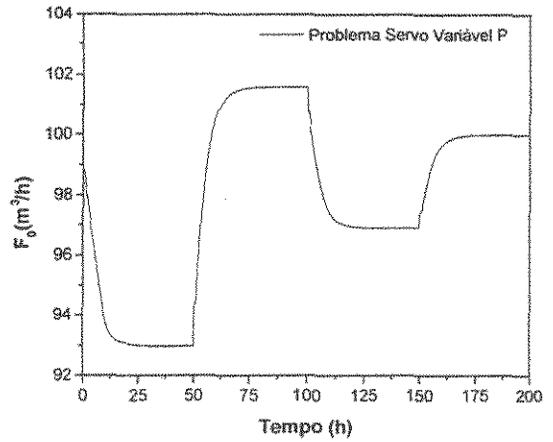
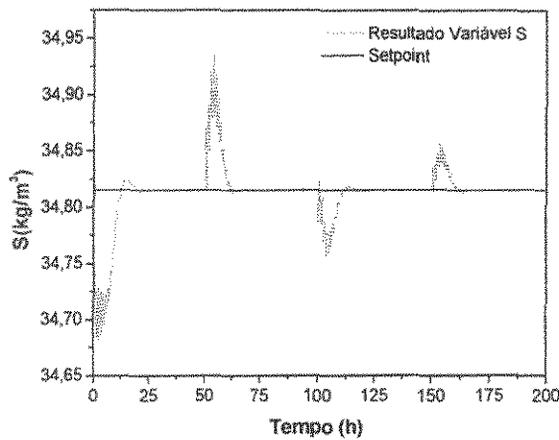
cuja lei de controle resultante da minimização da função objetivo descrita pela equação (1.50) é:

$$\Delta u = (A^T A + \Lambda^T \Lambda)^{-1} A^T E' \quad (1.50)$$

Os valores das variáveis manipuladas no instante k são calculados da seguinte forma:

$$u_k = u_{k-1} + \Delta u_k \quad (1.51)$$

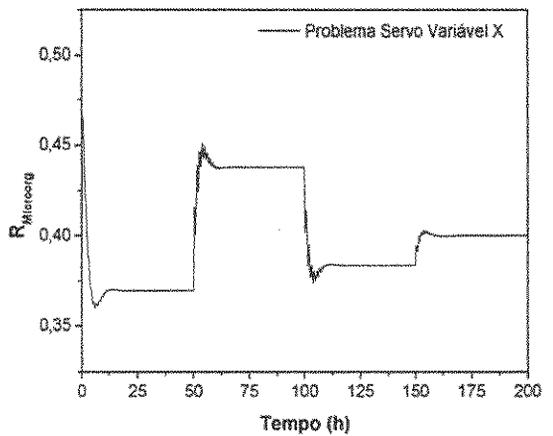
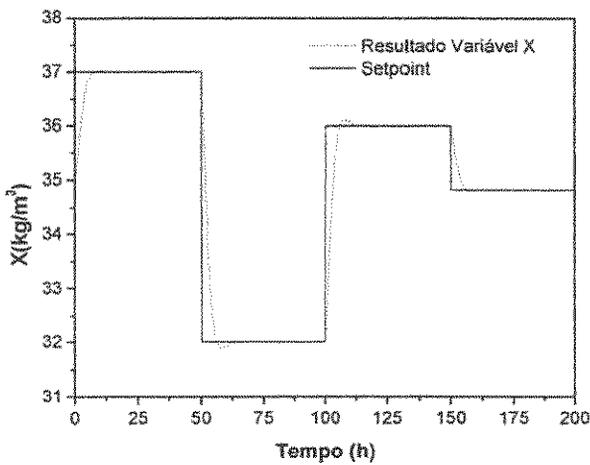
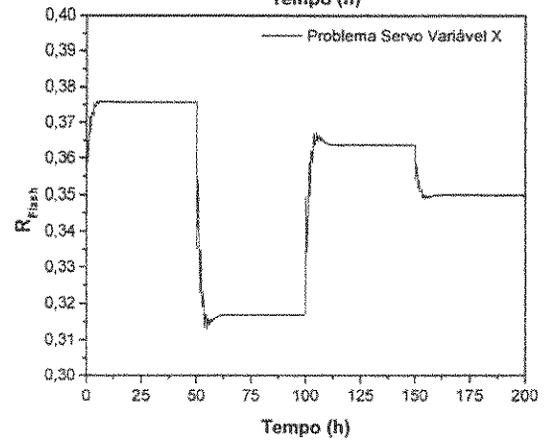
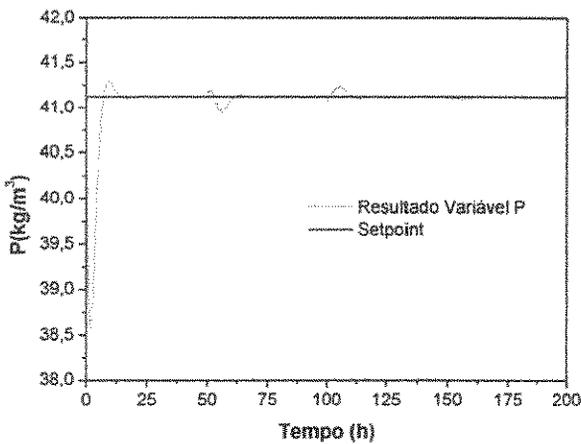
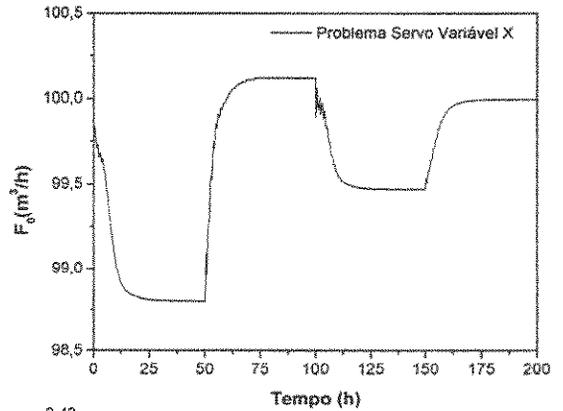
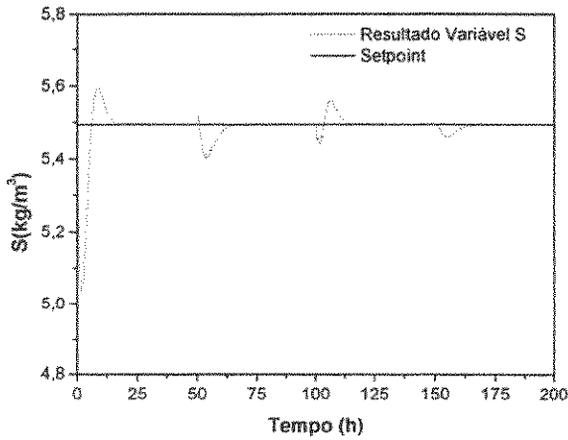
Problema do Tipo Servo (DMC) variável P



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

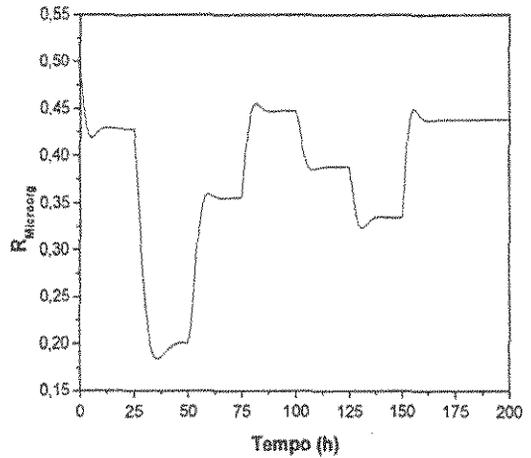
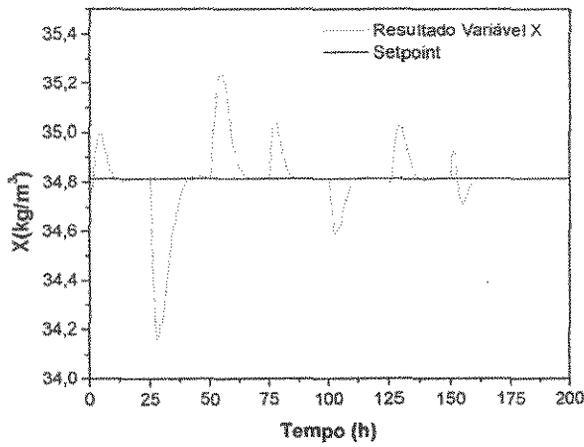
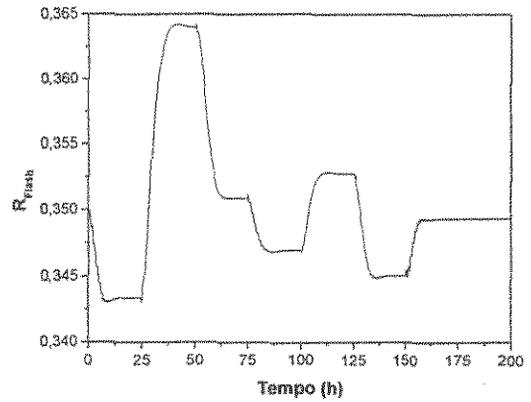
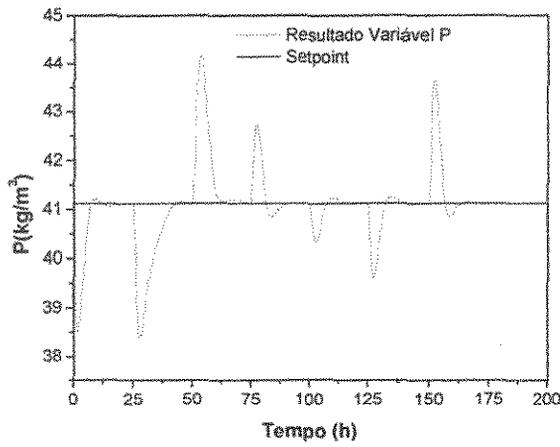
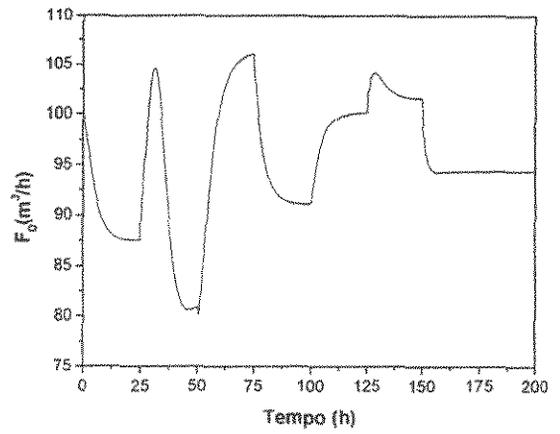
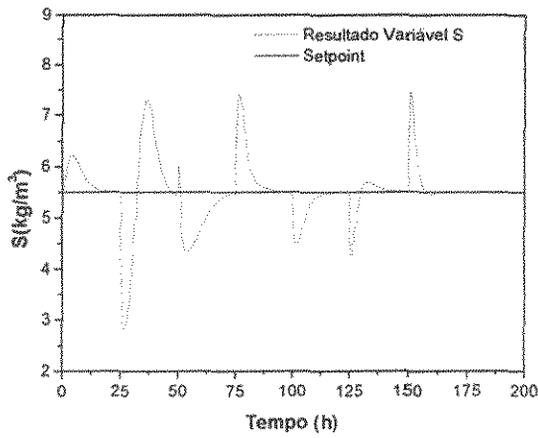
Problema do Tipo Servo (DMC) variável X



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

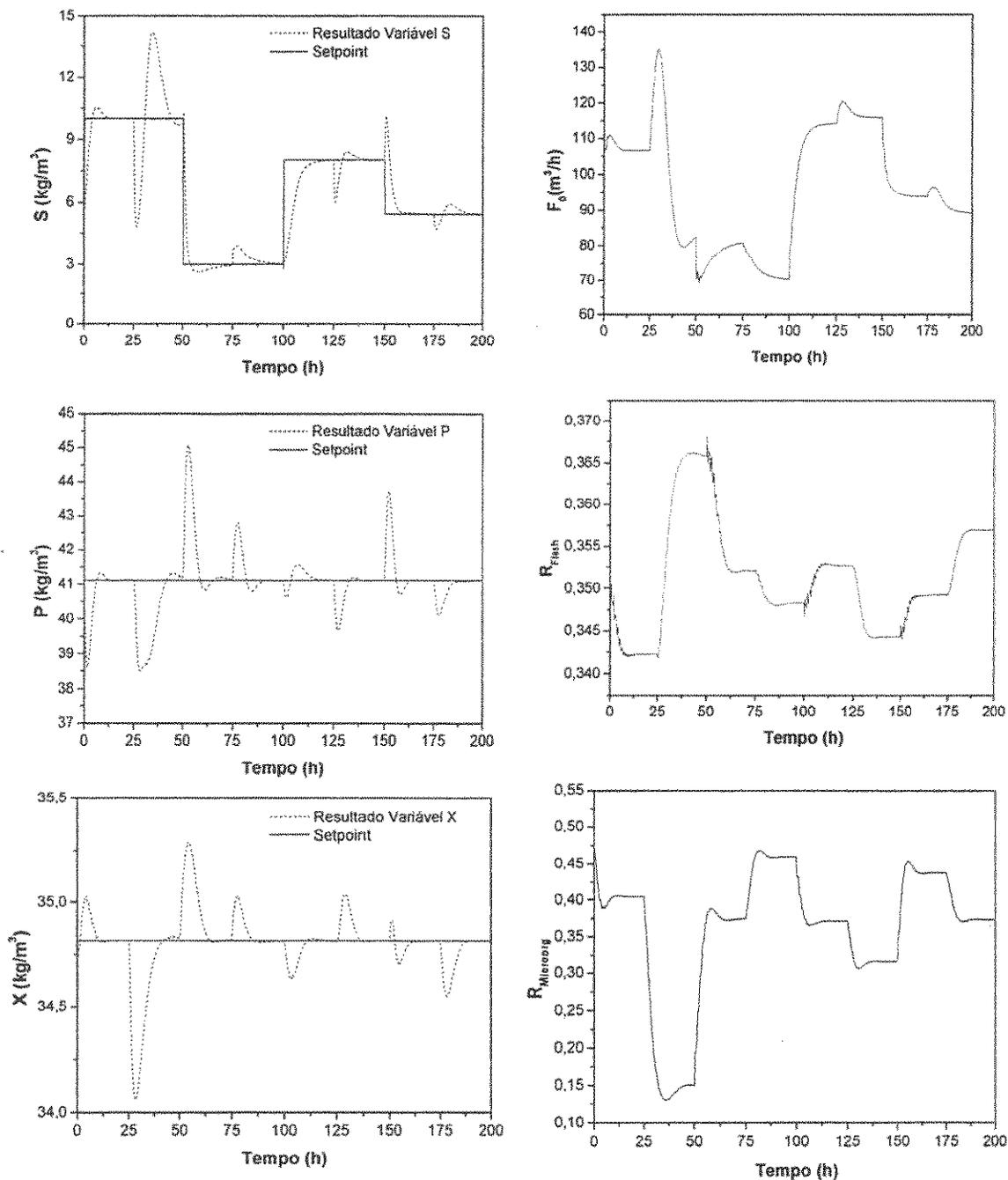
Problema do Tipo Regulador (DMC)



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

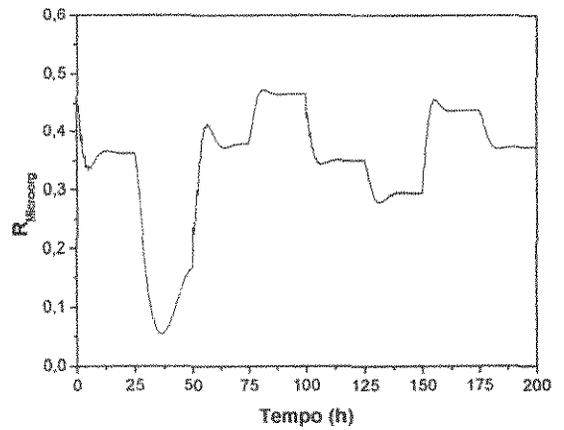
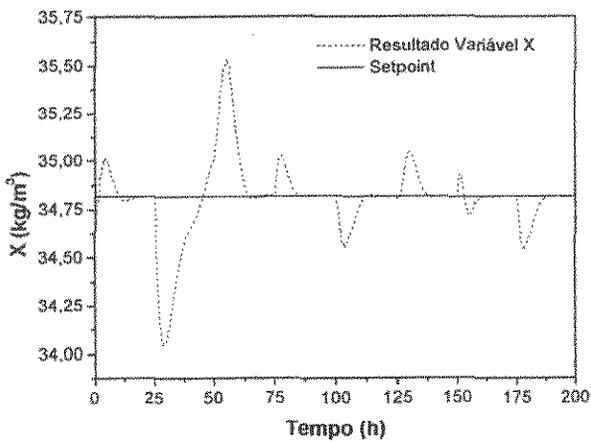
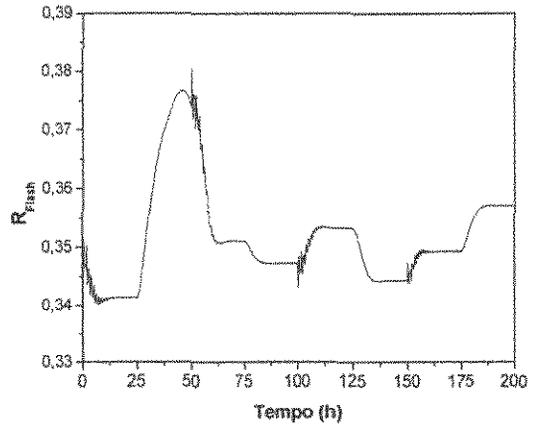
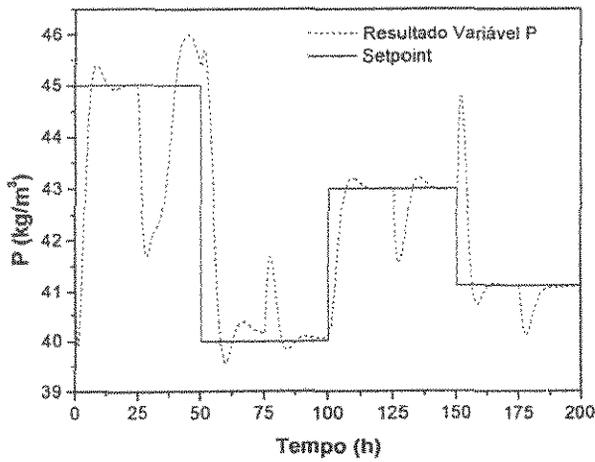
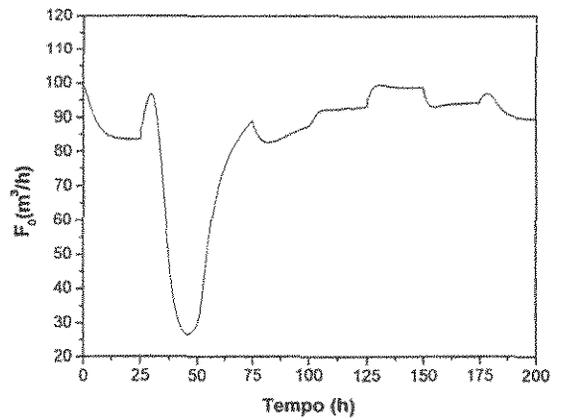
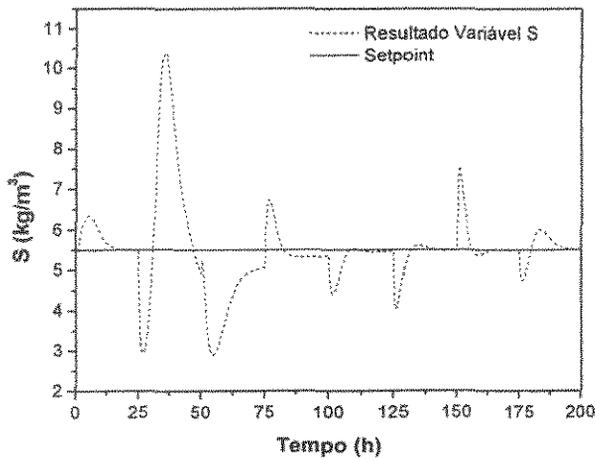
Problema do Tipo Servo e Regulador (DMC) variável S



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

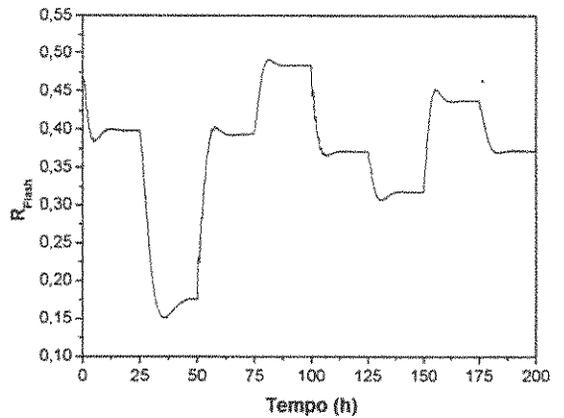
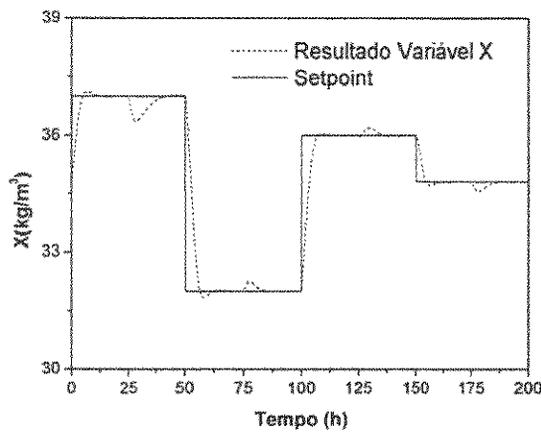
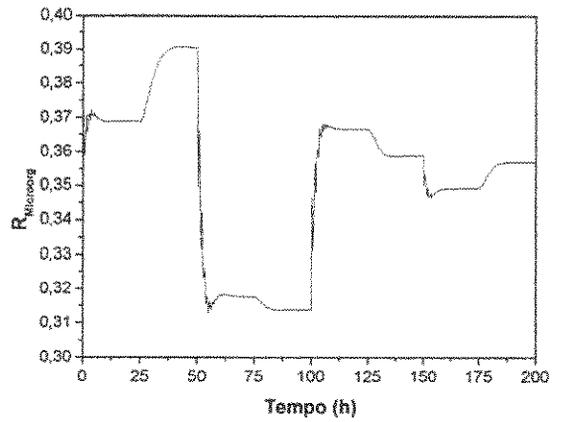
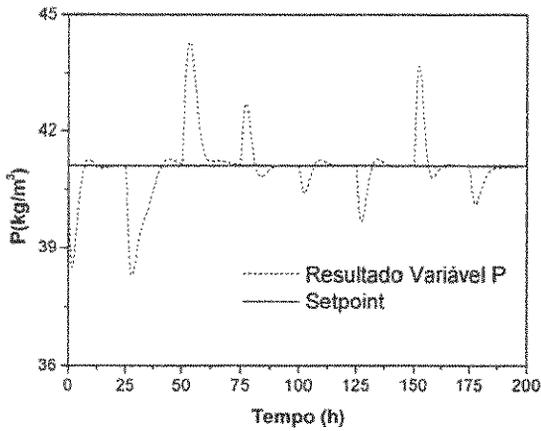
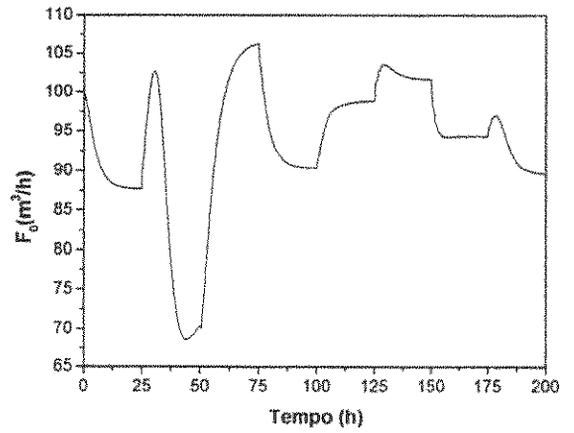
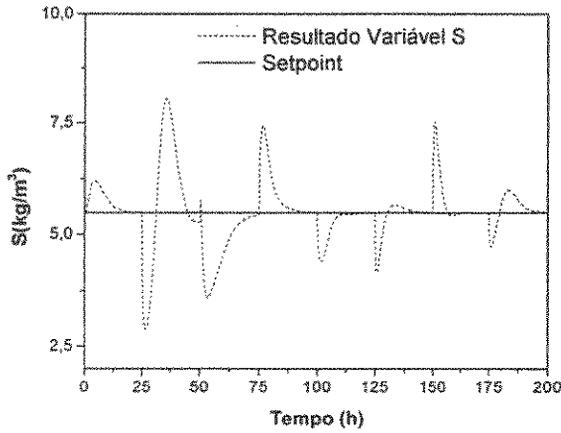
Problema do Tipo Servo e Regulador (DMC) variável P



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

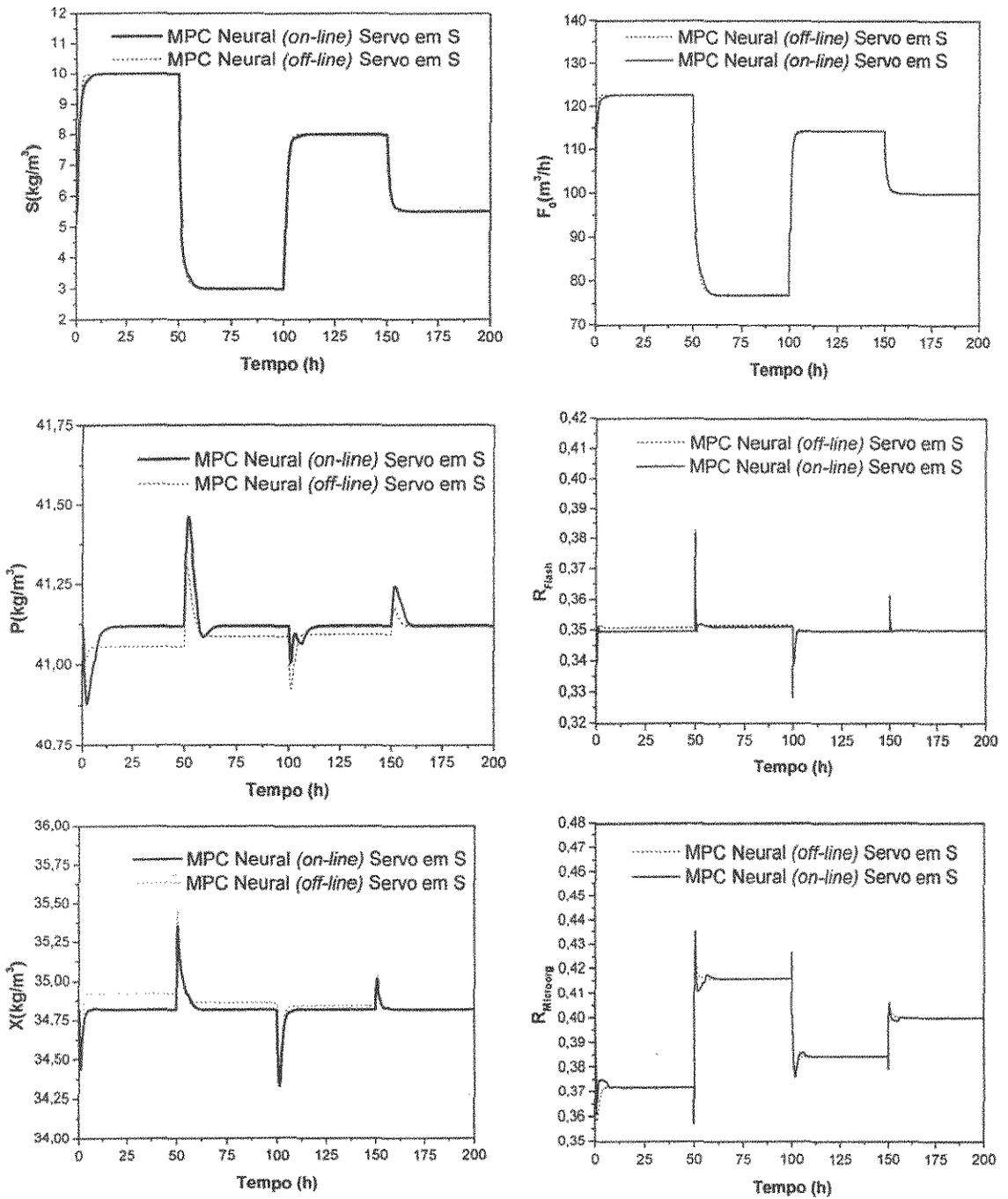
Problema do Tipo Servo e Regulador (DMC) variável X



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

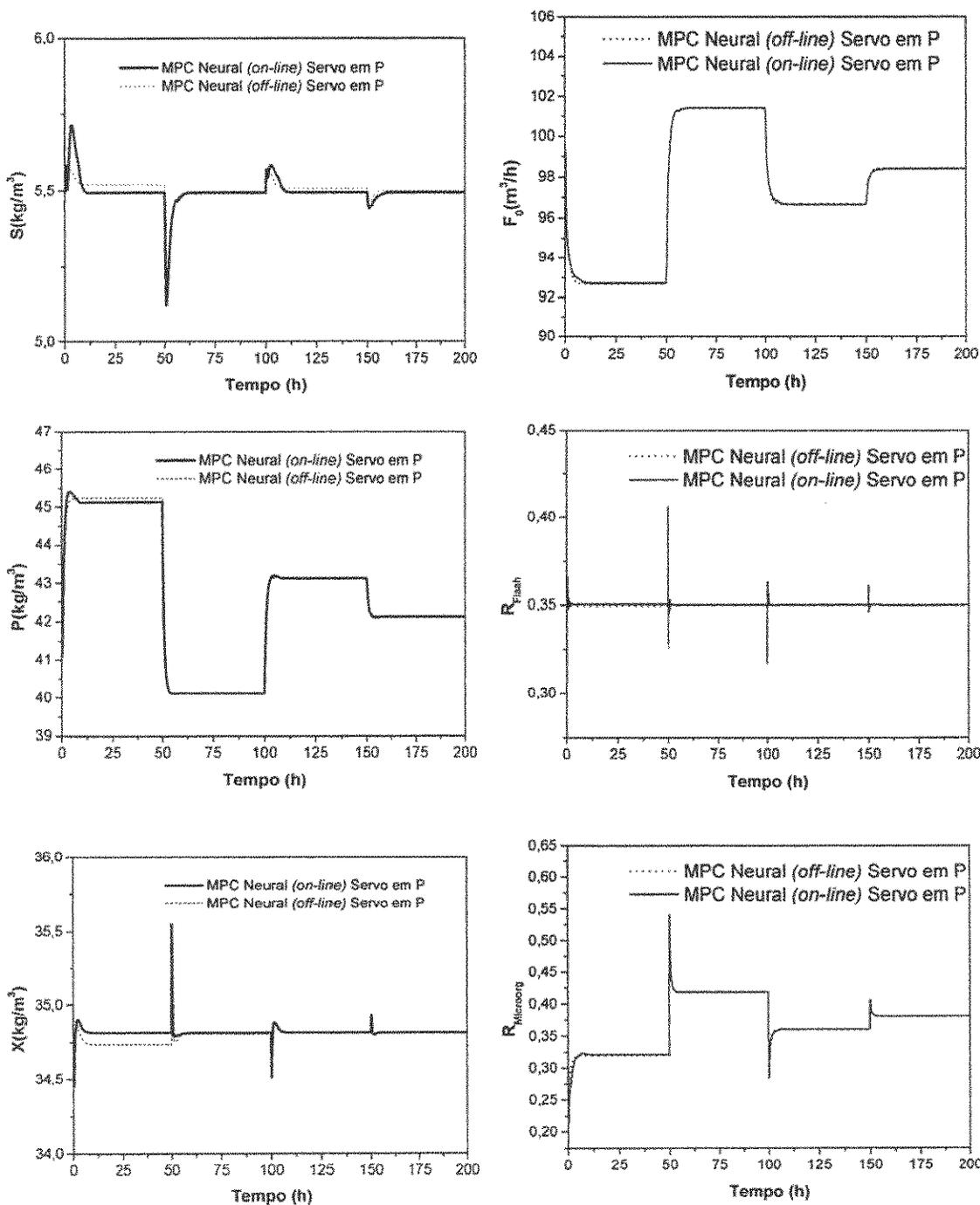
Problema do Tipo Servo (MPC Neural) variável S



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

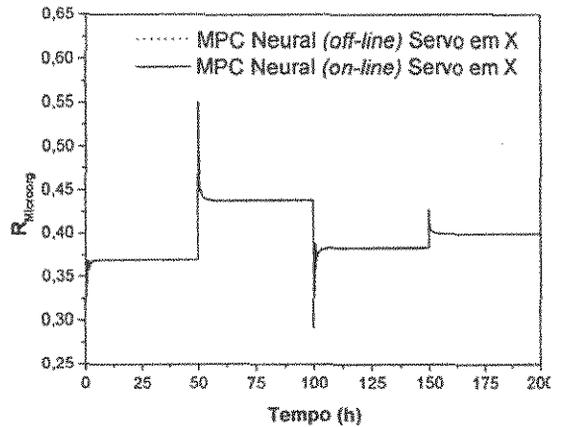
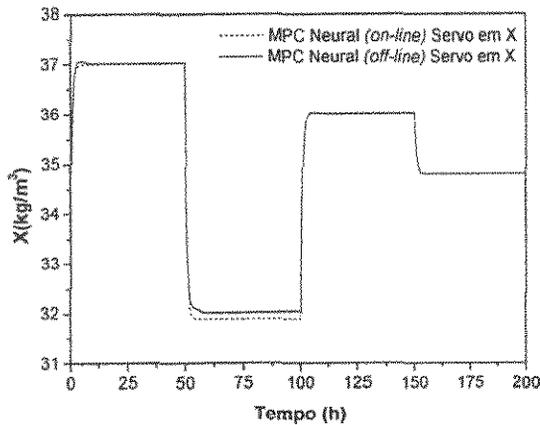
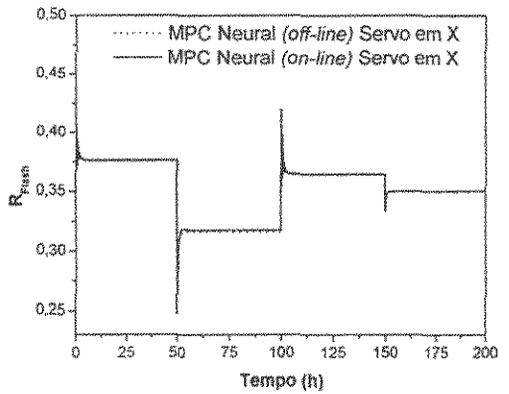
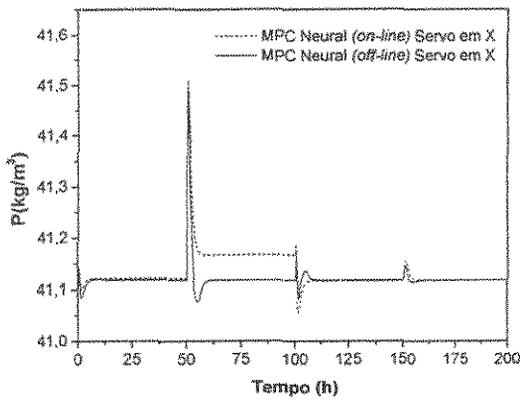
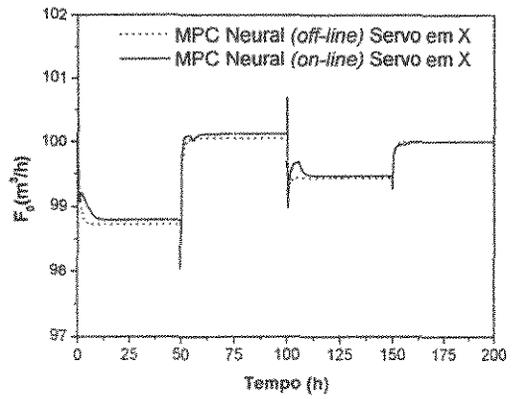
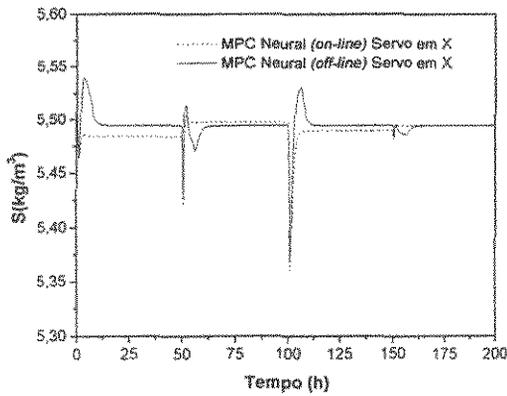
Problema do Tipo Servo (MPC Neural) variável P



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

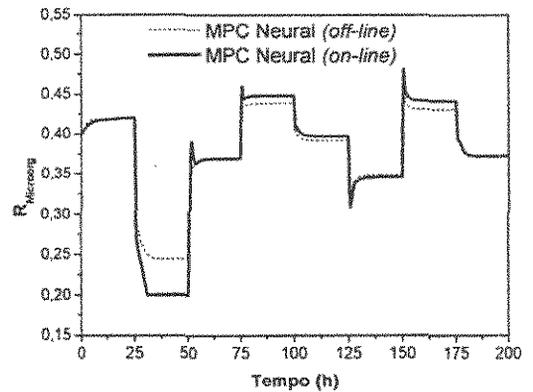
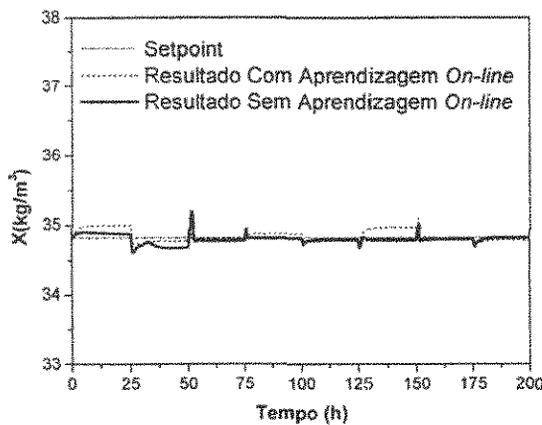
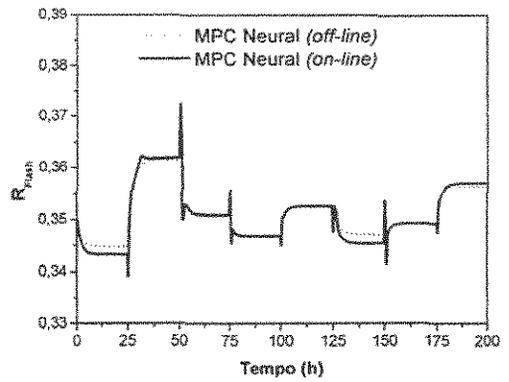
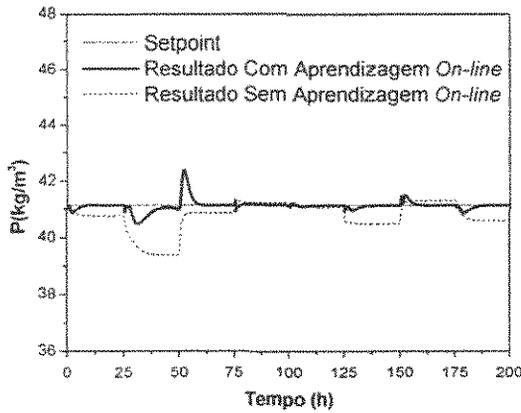
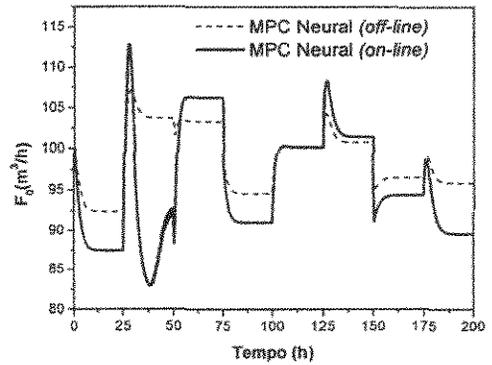
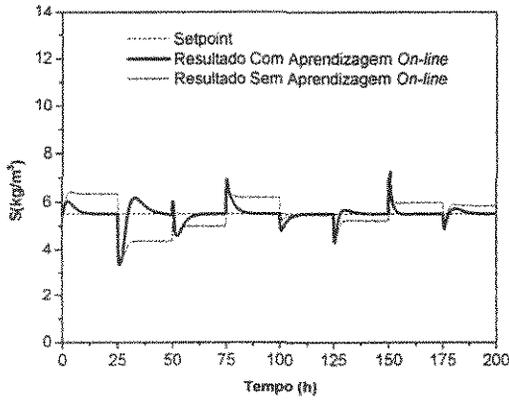
Problema do Tipo Servo (MPC Neural) variável X



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

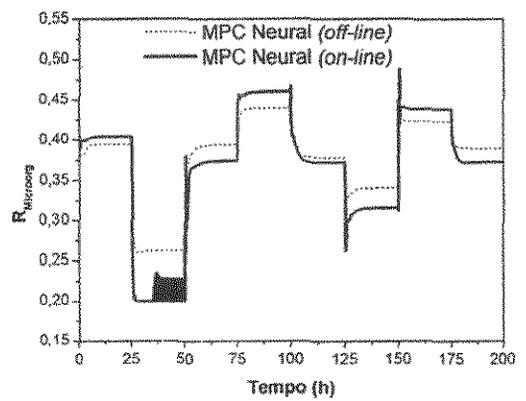
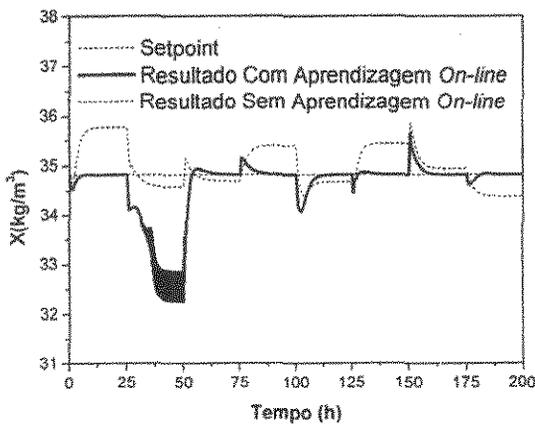
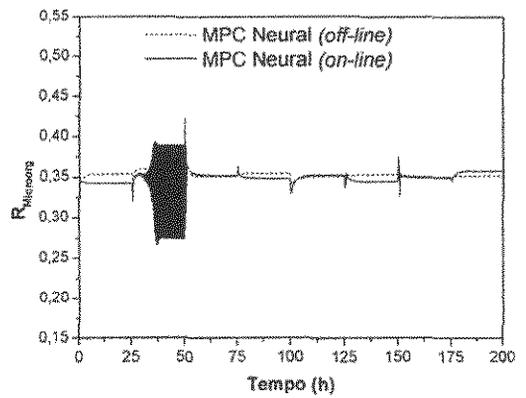
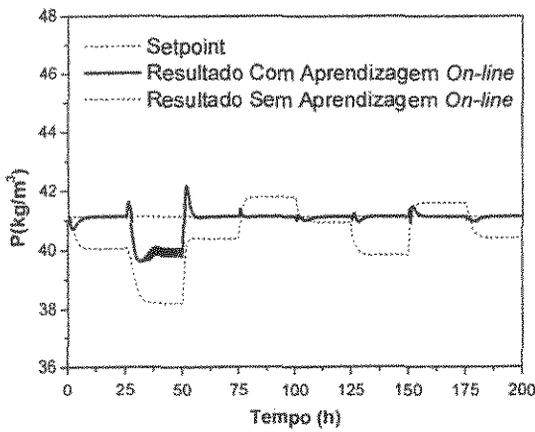
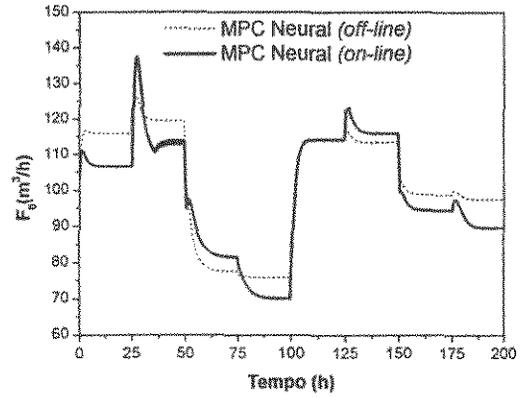
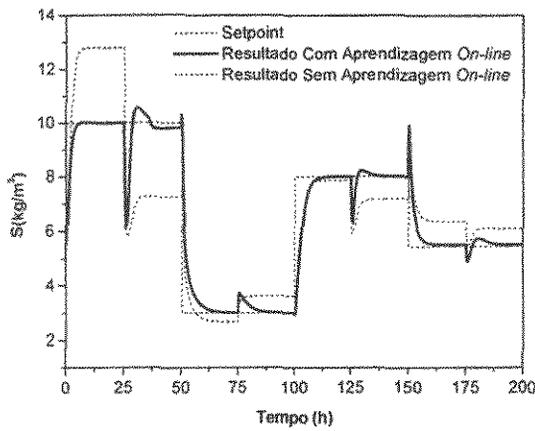
Problema do Tipo Regulador (MPC Neural)



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

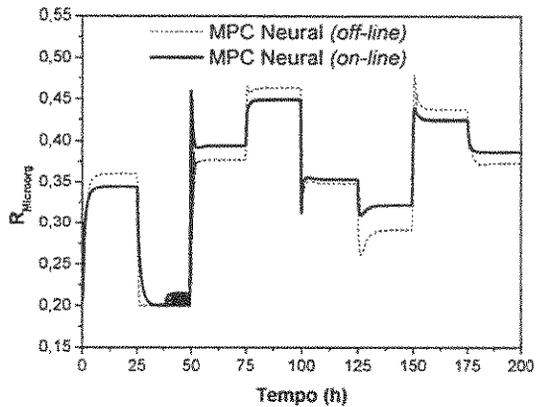
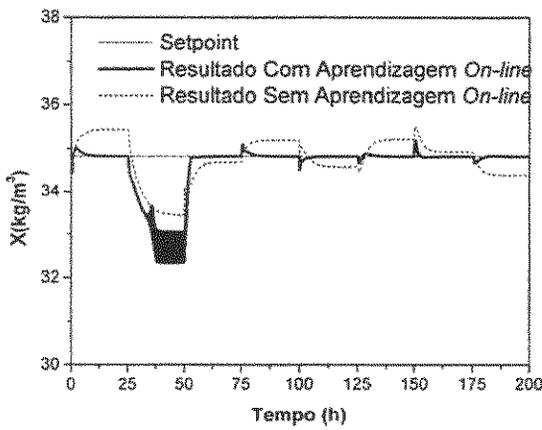
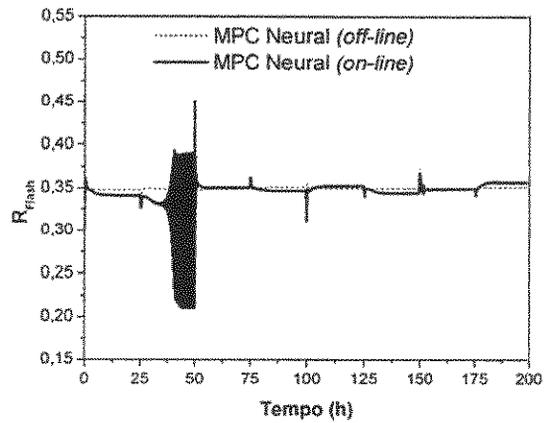
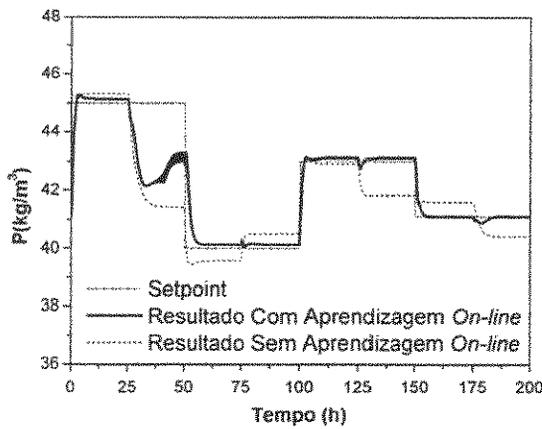
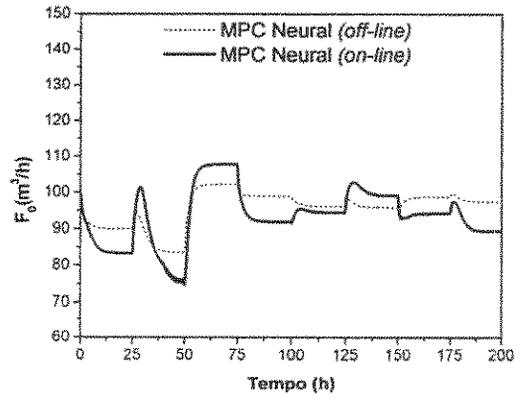
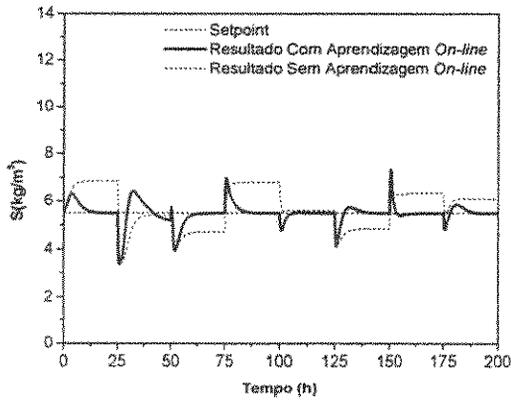
Problema do Tipo Servo e Regulador (MPC Neural) variável S



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

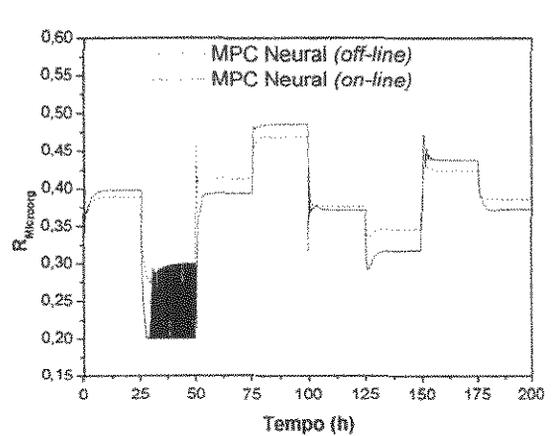
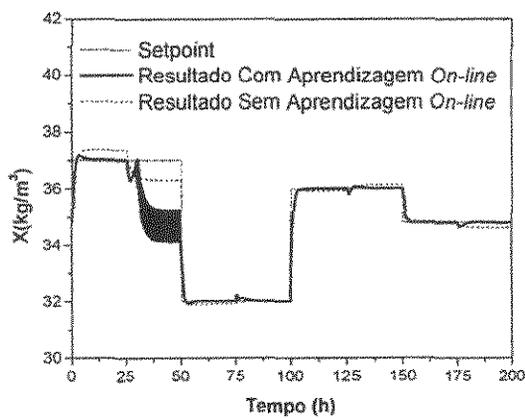
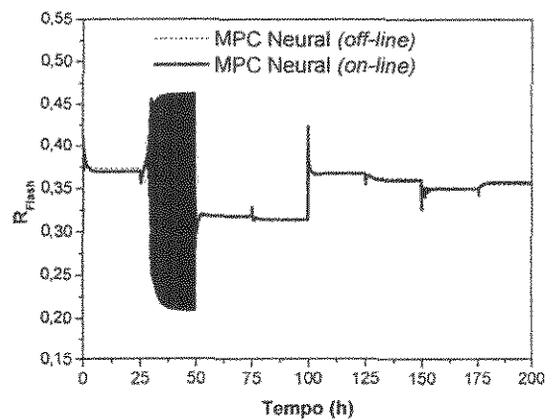
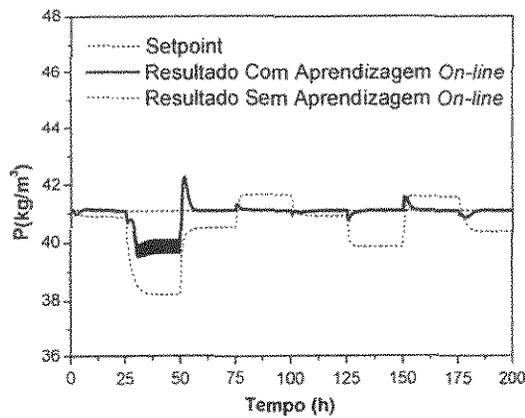
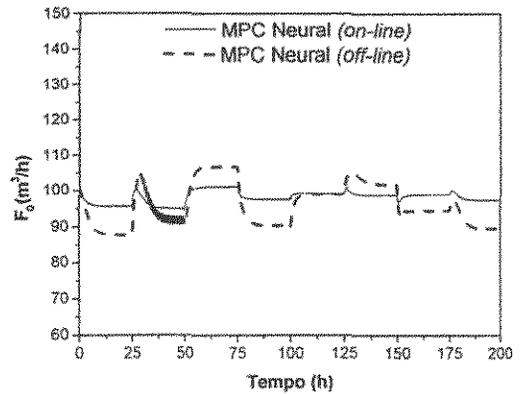
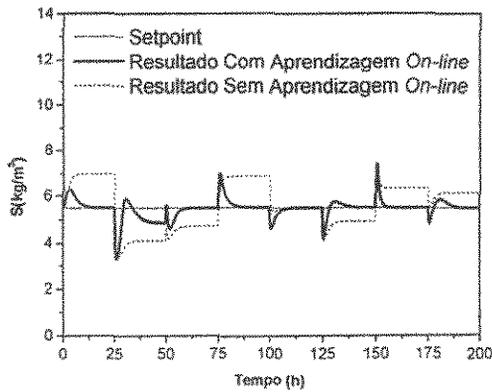
Problema do Tipo Servo e Regulador (MPC Neural) variável P



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

Problema do Tipo Servo e Regulador (MPC Neural) variável X

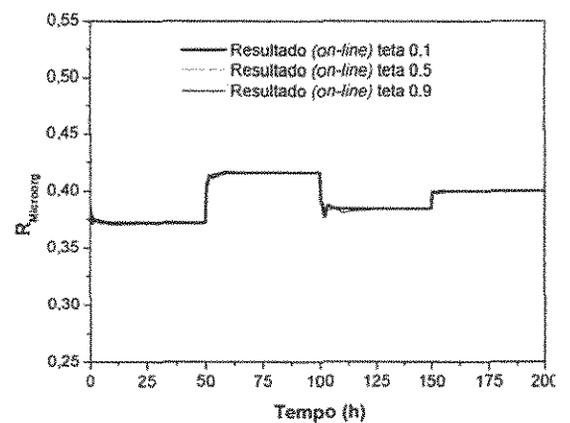
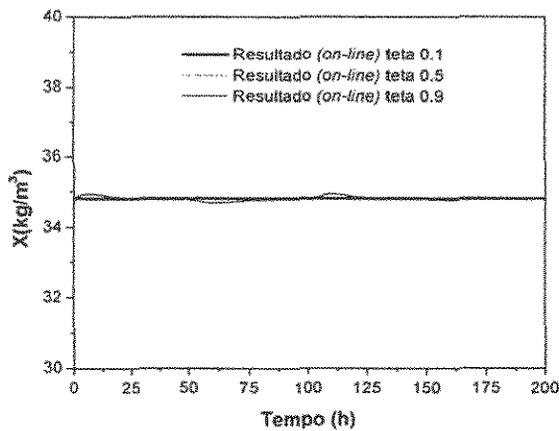
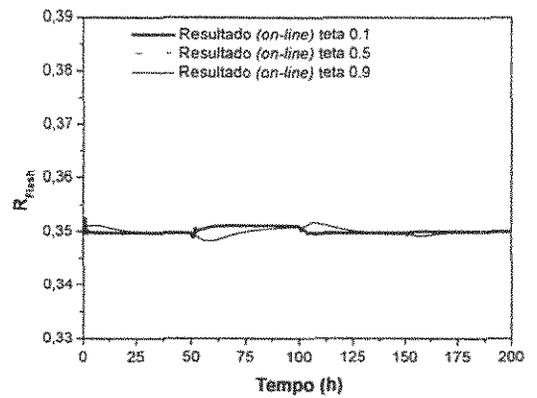
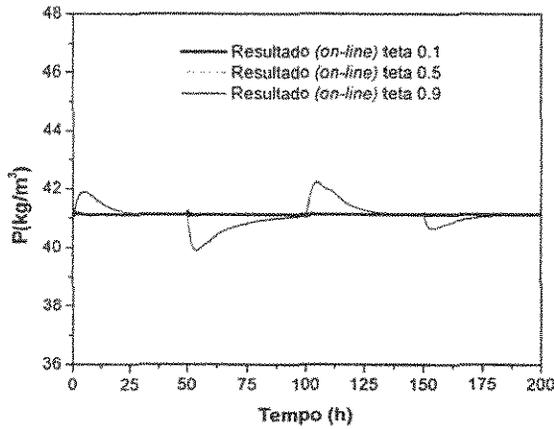
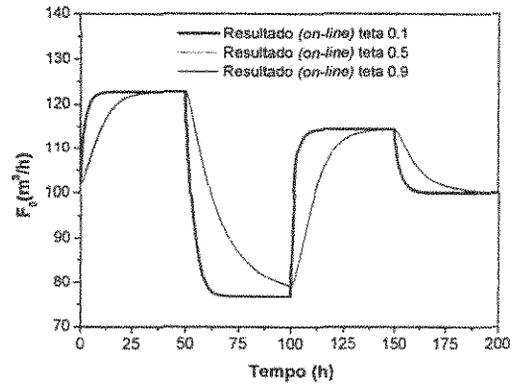
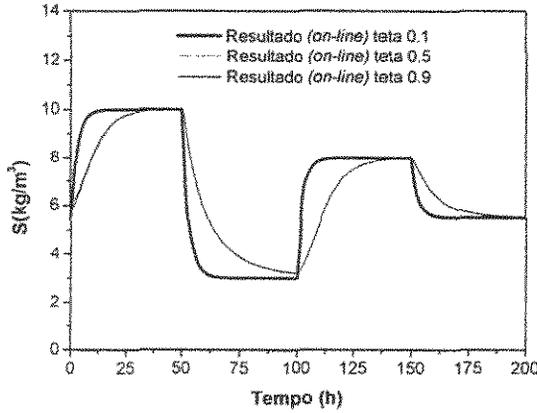


Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

A seguir serão apresentados os resultados para o algoritmo Kalman 2 (ganho constante) e Kalman 4 (ganho variável) com diferentes valores de θ (0,1-0,5-0,9) para todos os problemas estudados: Servo, Regulador e Servo Regulador.

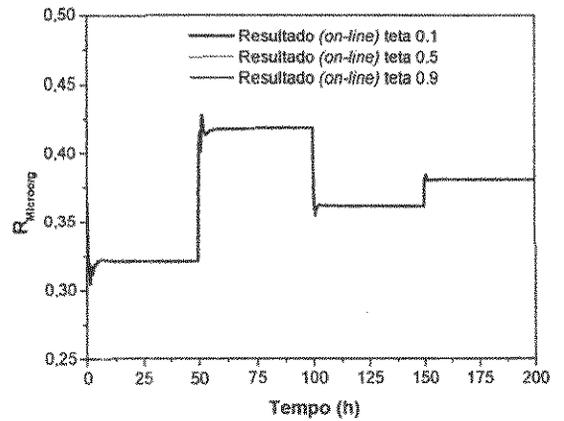
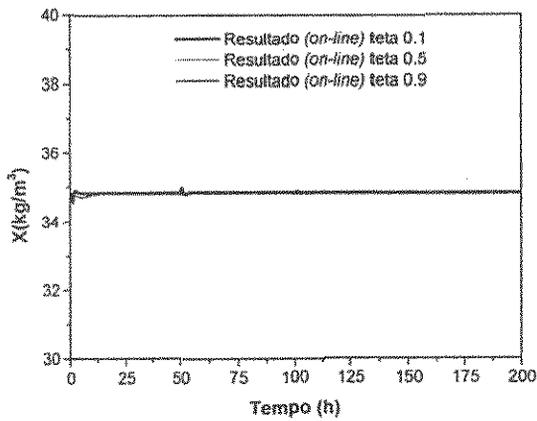
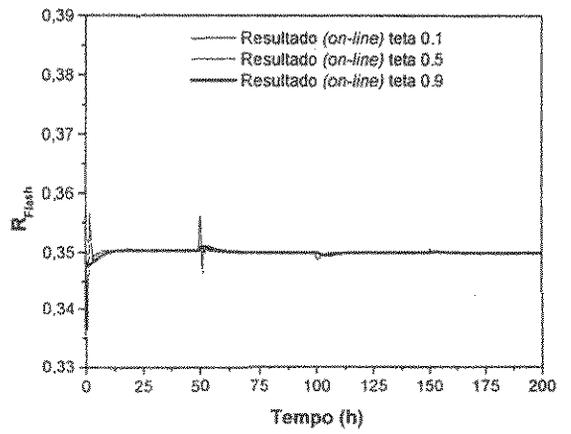
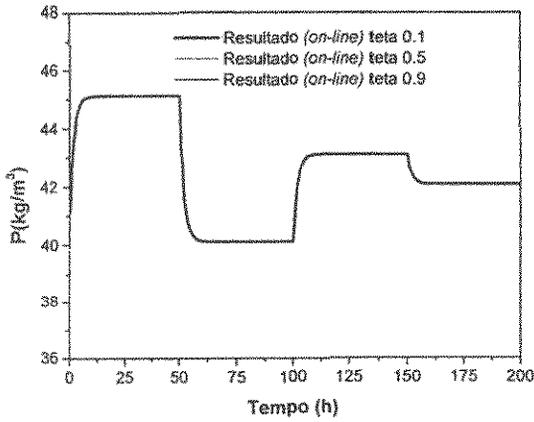
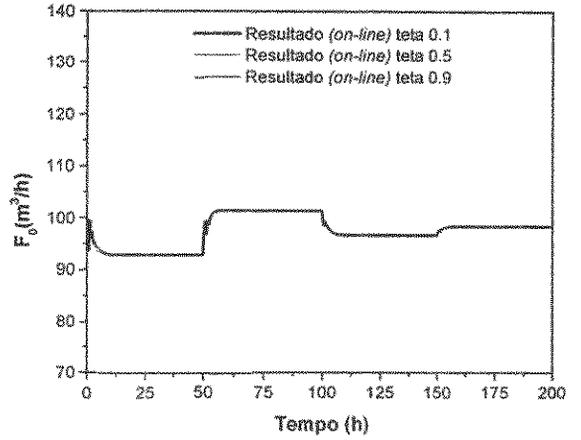
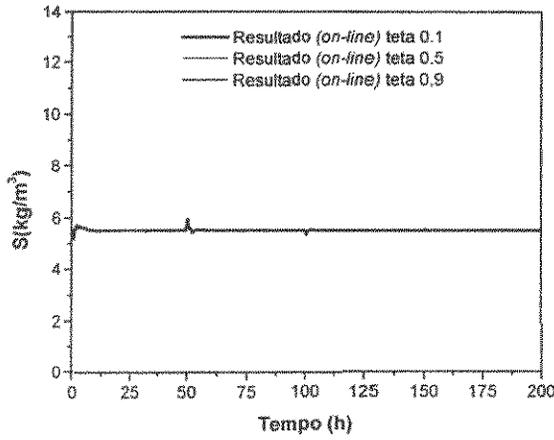
Problema do Tipo Servo (Kalman 2) variável S



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

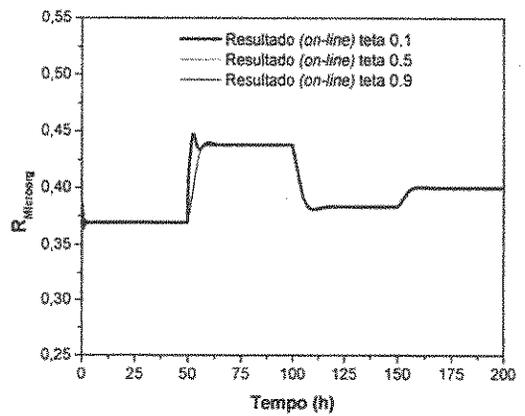
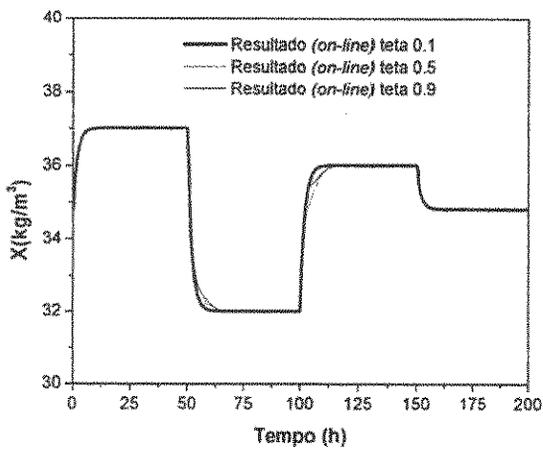
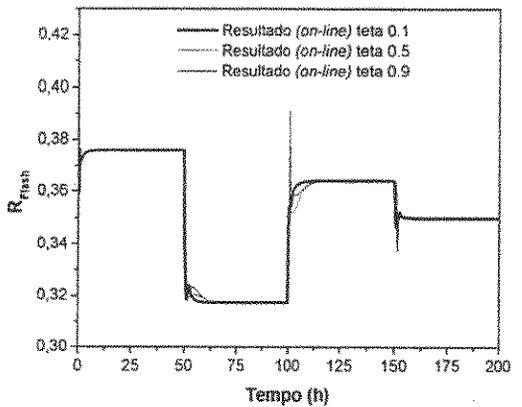
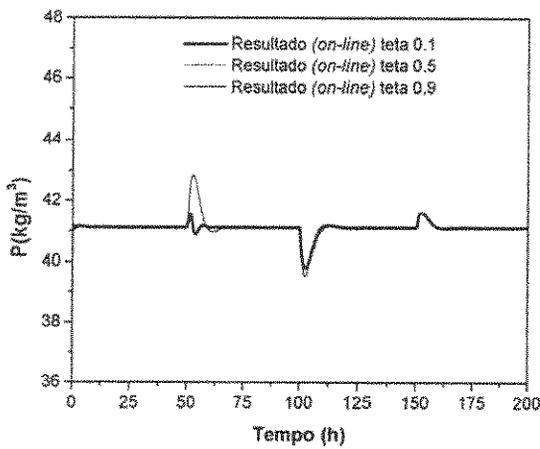
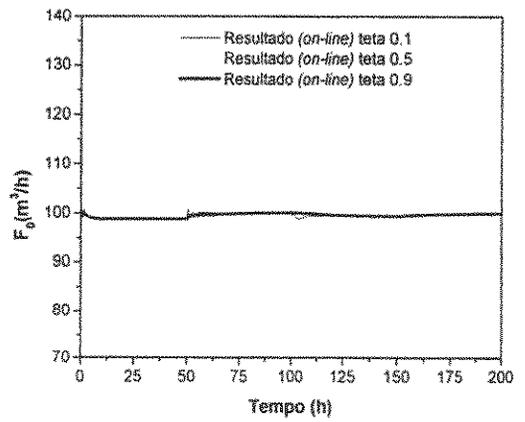
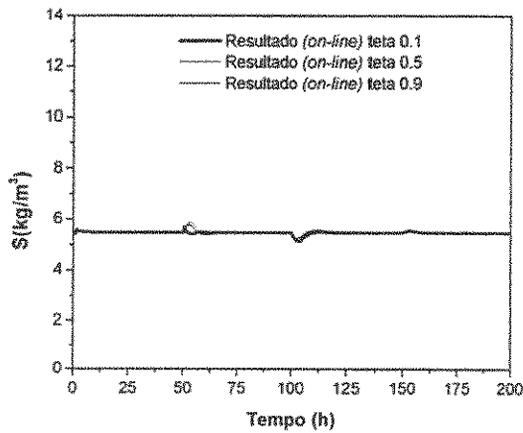
Problema do Tipo Servo (Kalman 2) variável P



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

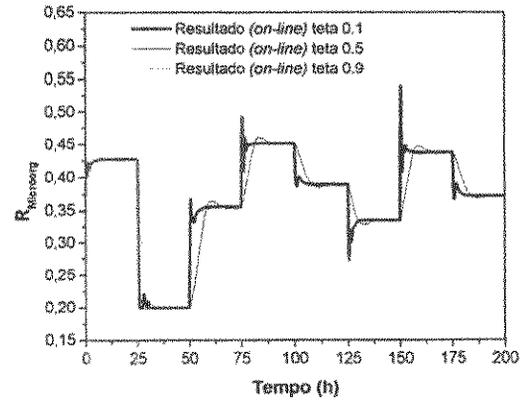
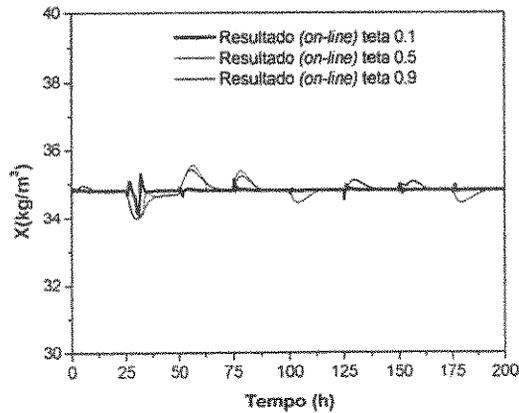
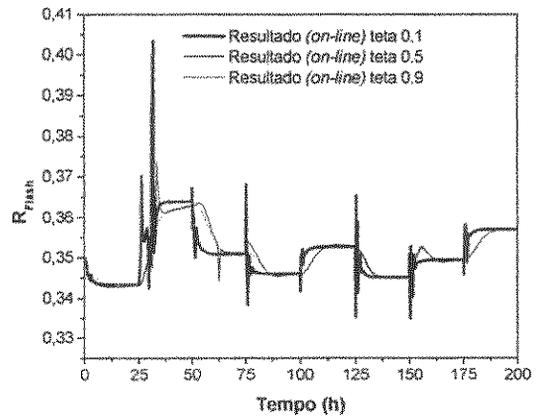
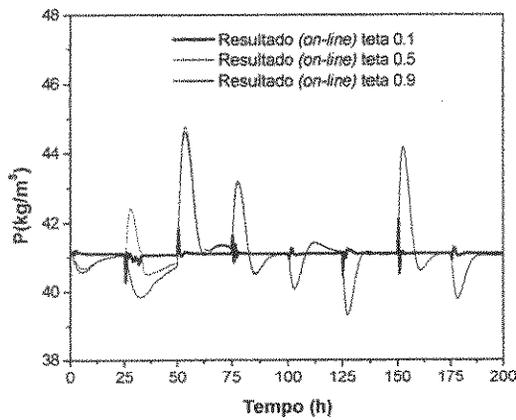
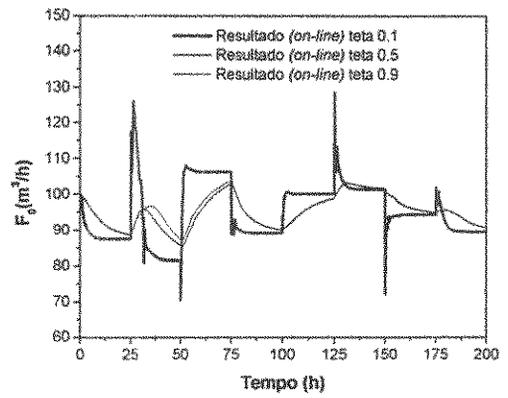
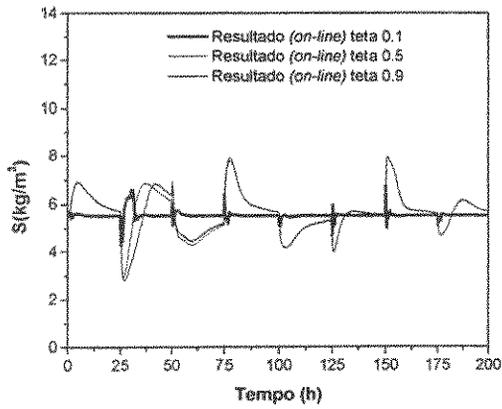
Problema do Tipo Servo (Kalman 2) variável X



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

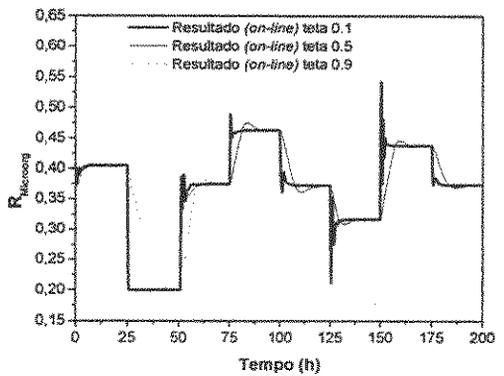
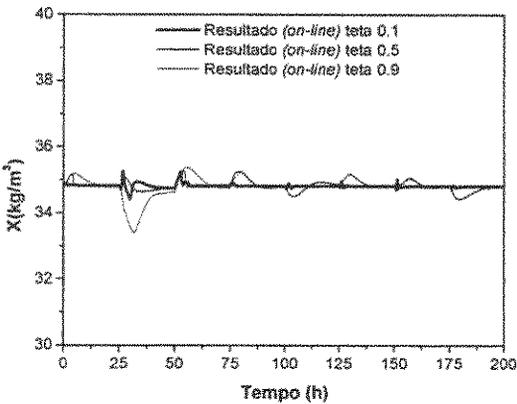
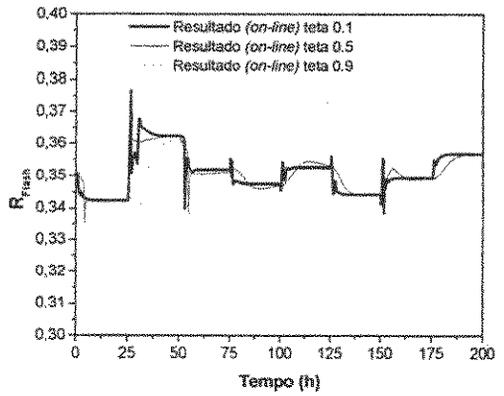
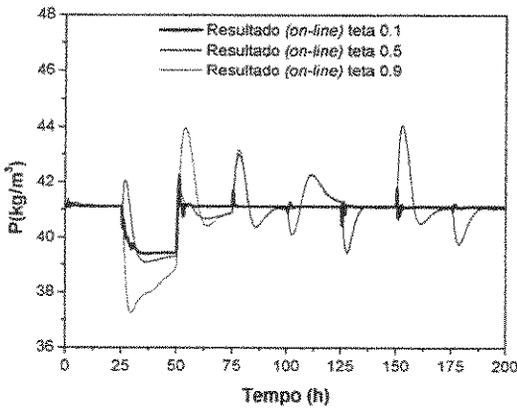
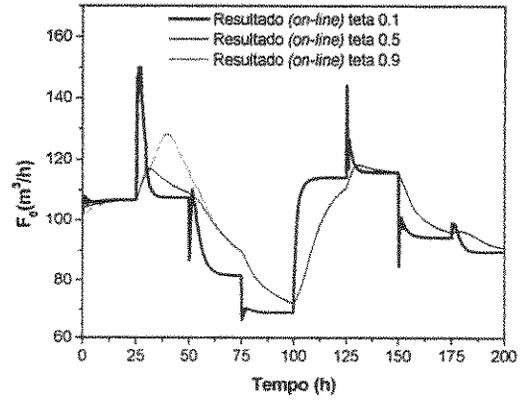
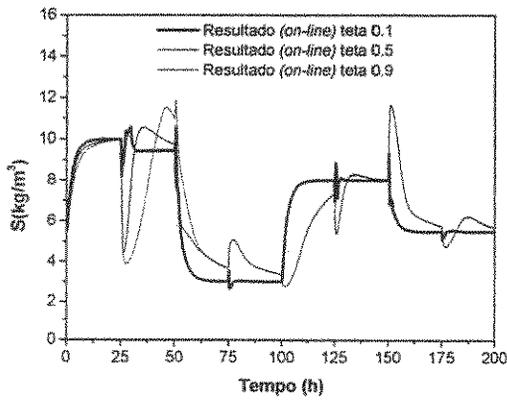
Problema do Tipo Regulador (Kalman 2)



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

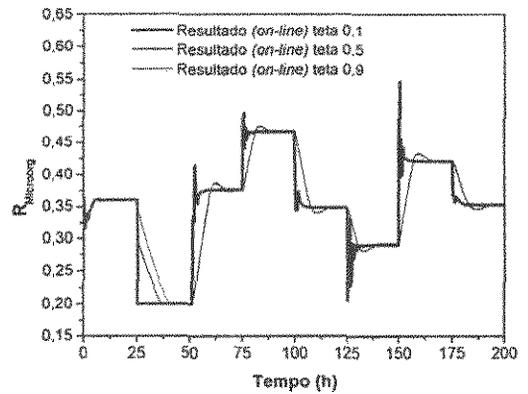
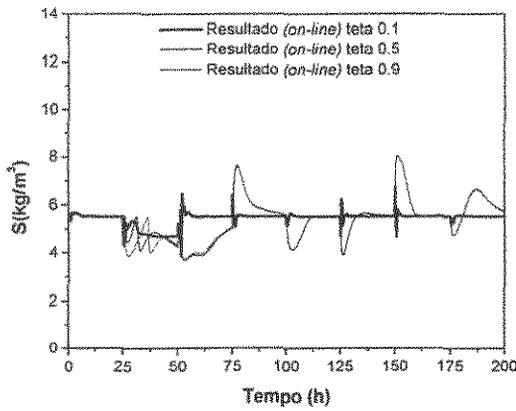
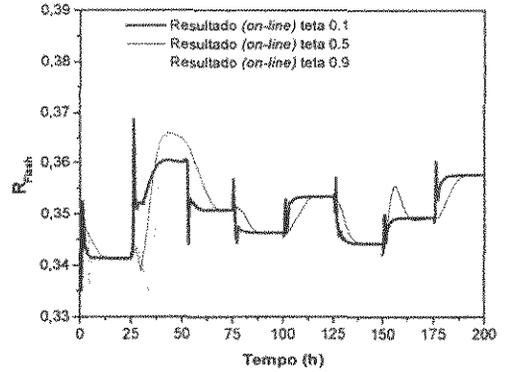
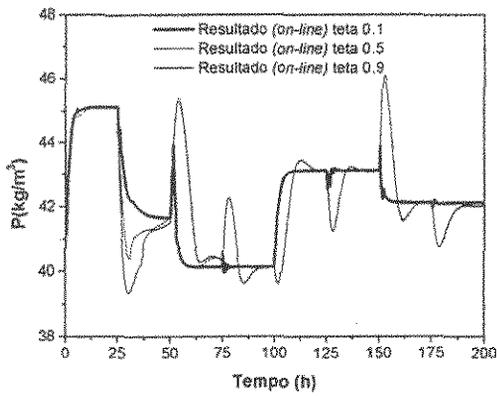
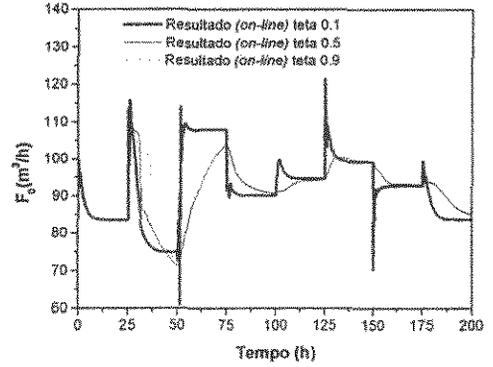
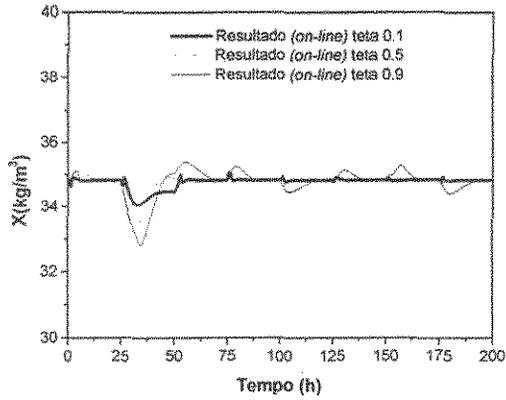
Problema do Tipo Servo e Regulador (Kalman 2) variável S



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

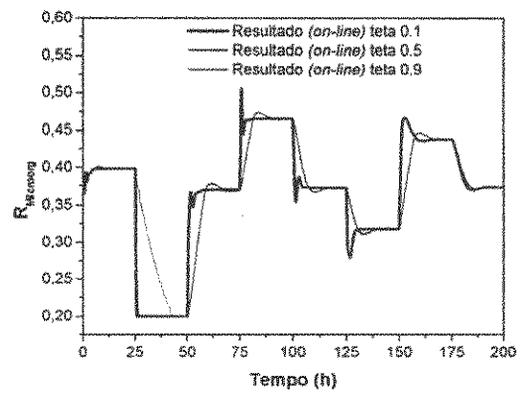
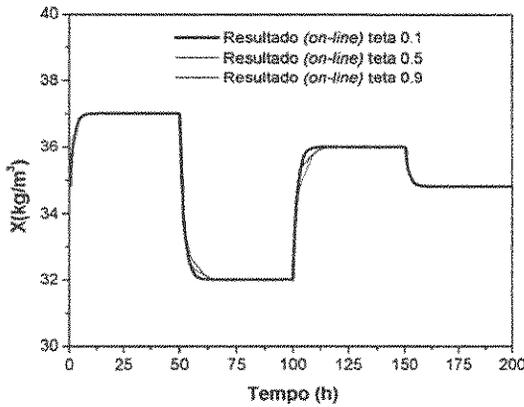
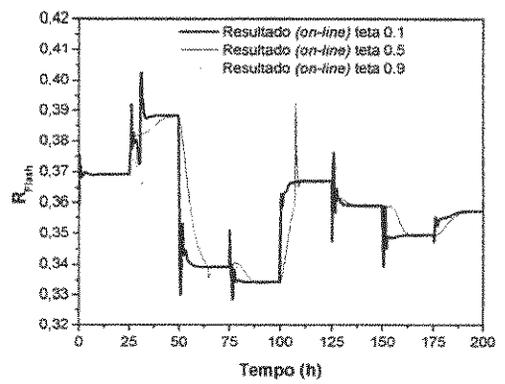
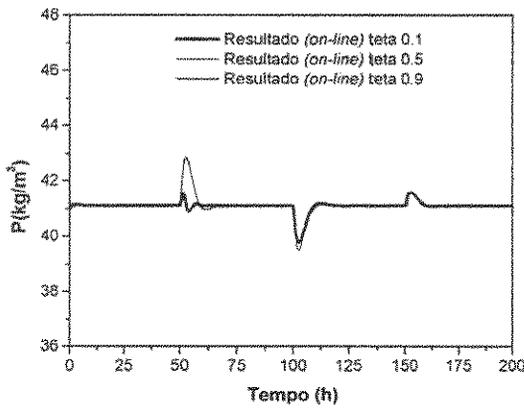
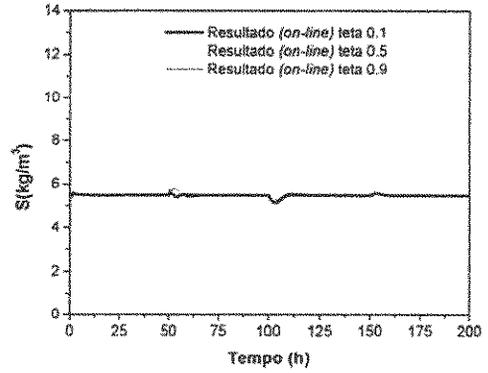
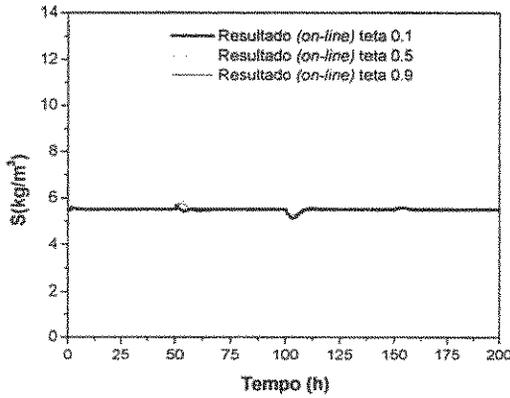
Problema do Tipo Servo e Regulador (Kalman 2) variável P



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

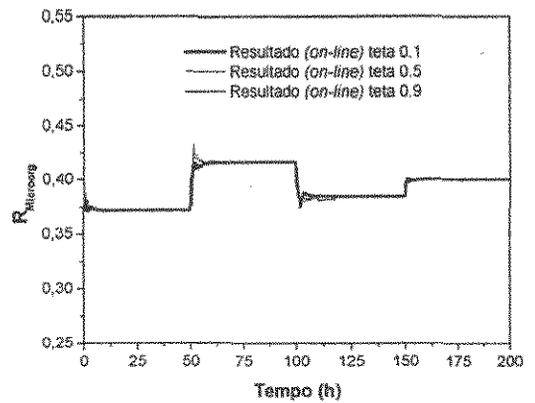
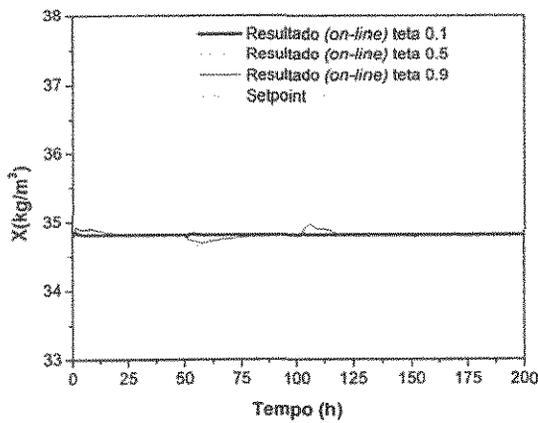
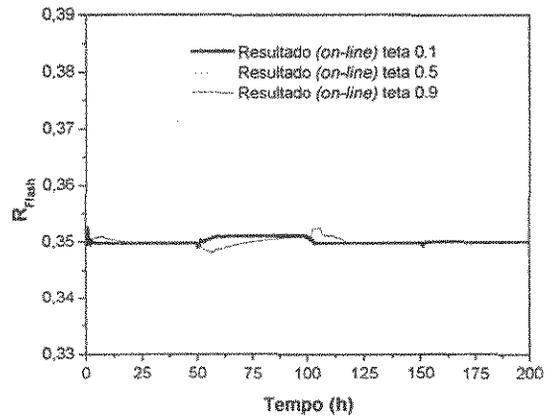
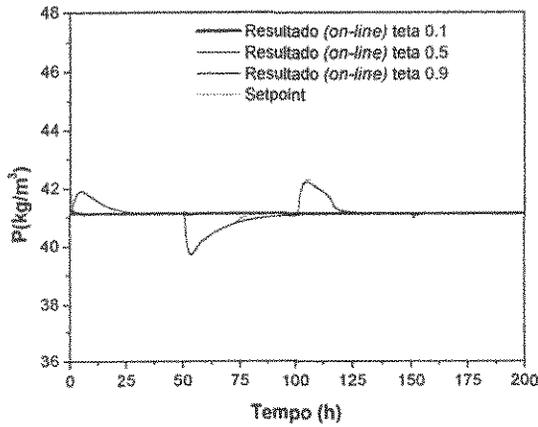
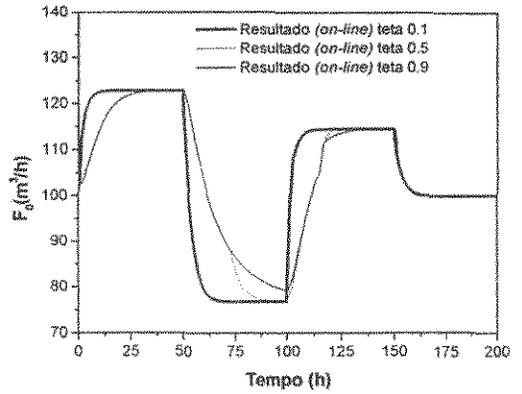
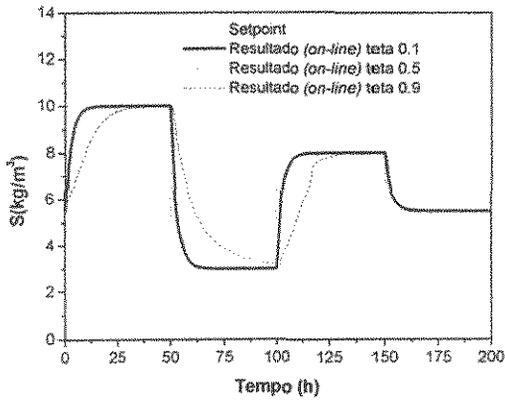
Problema do Tipo Servo e Regulador (Kalman 2) variável X



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

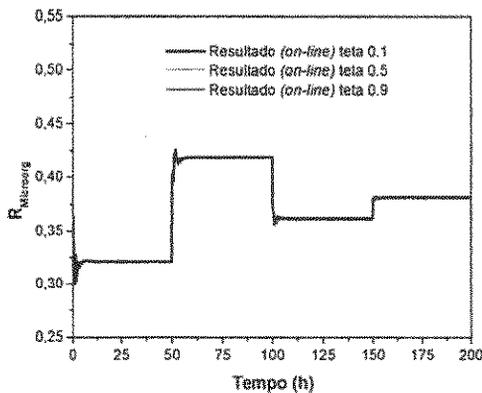
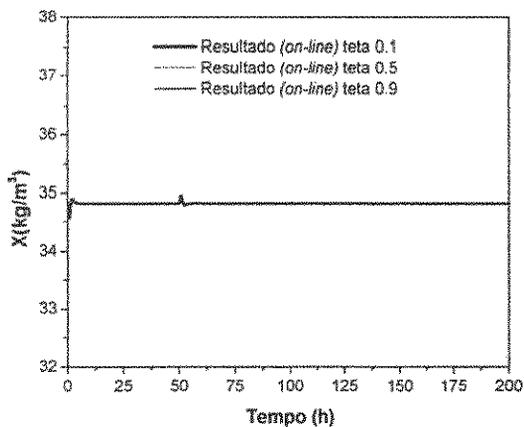
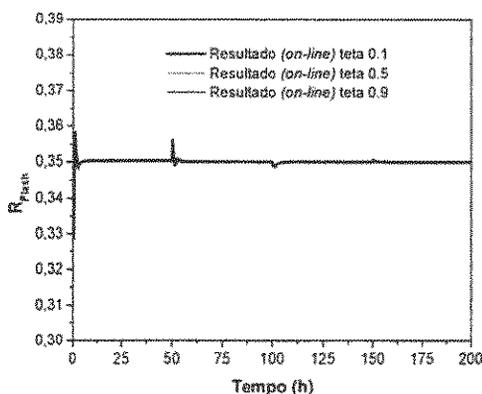
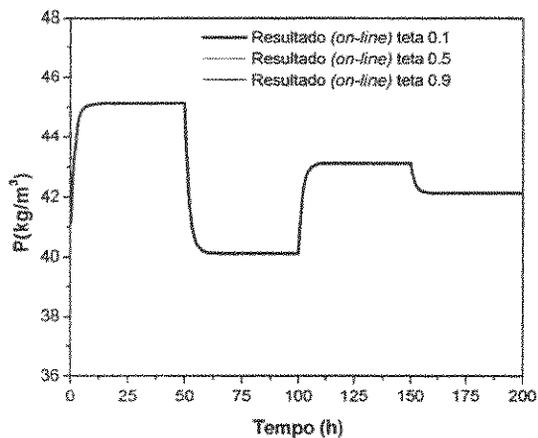
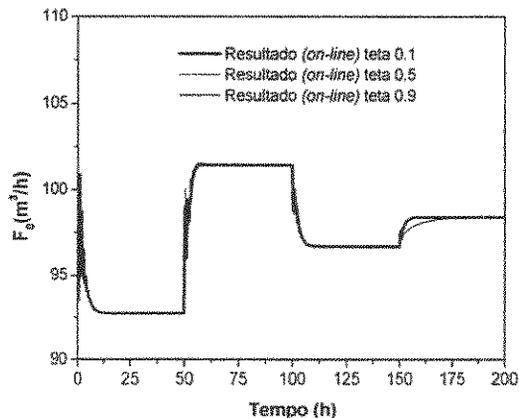
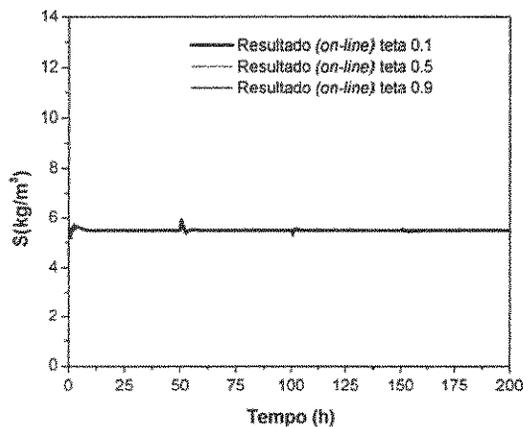
Problema do Tipo Servo (Kalman 4) variável S



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

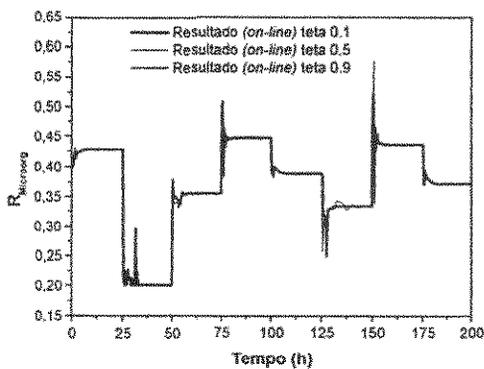
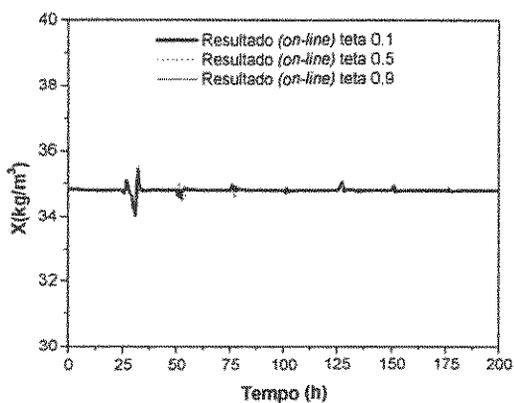
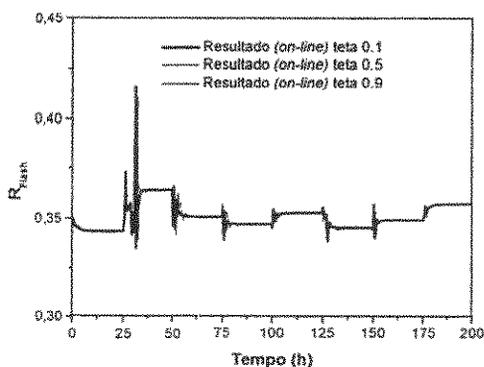
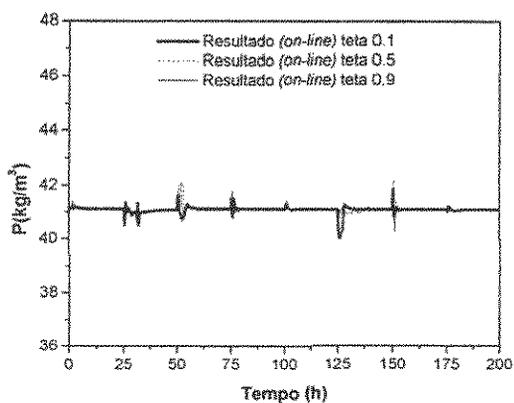
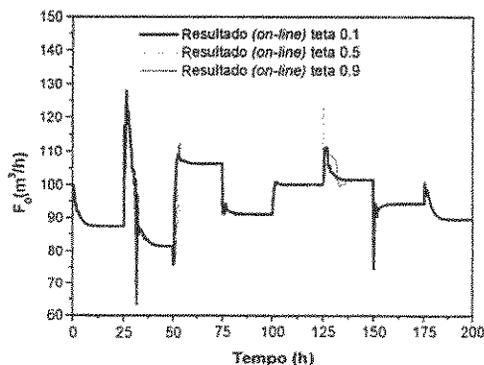
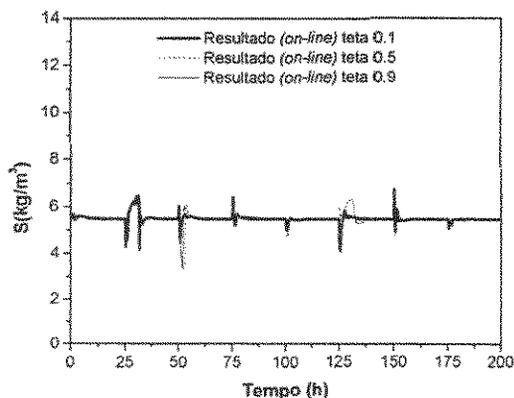
Problema do Tipo Servo (Kalman 4) variável P



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

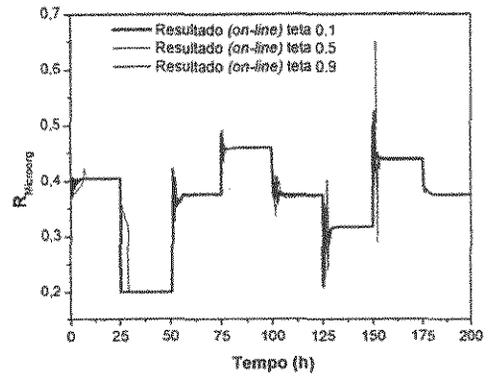
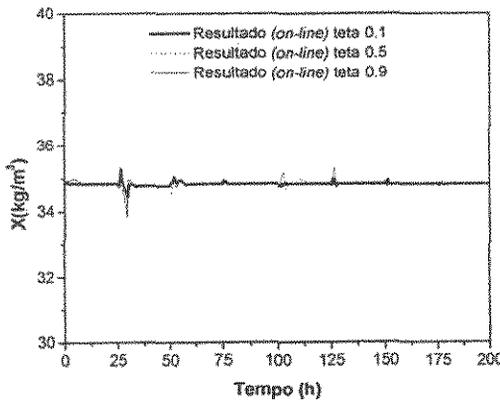
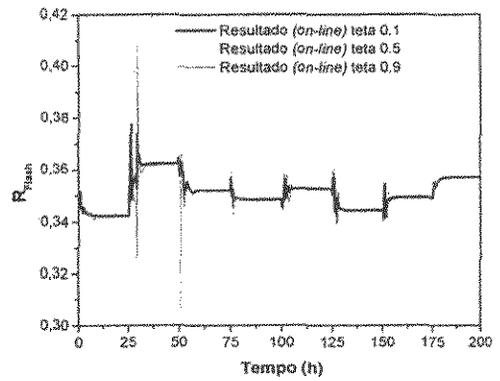
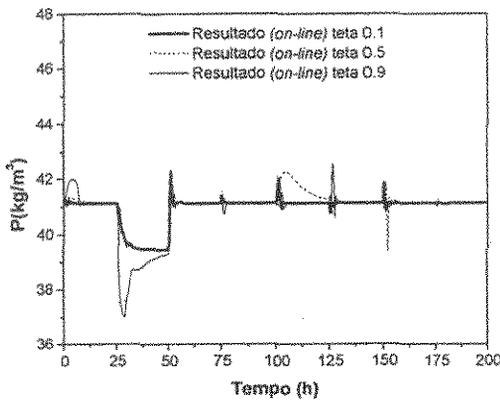
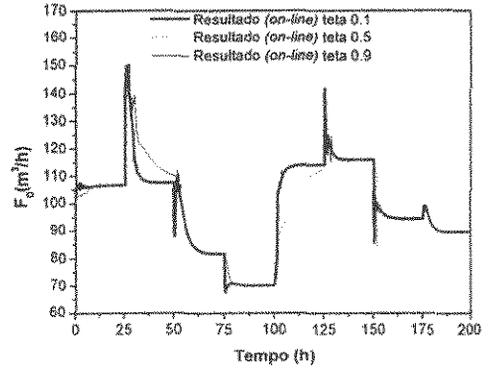
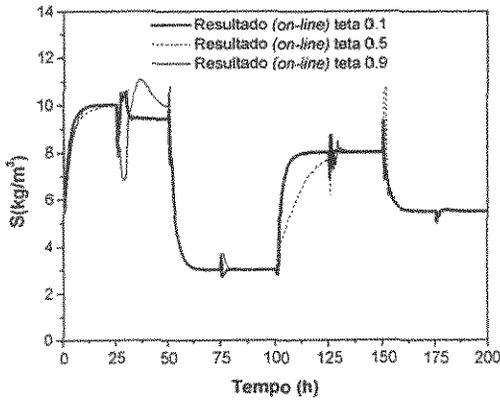
Problema do Tipo Regulador (Kalman 4)



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

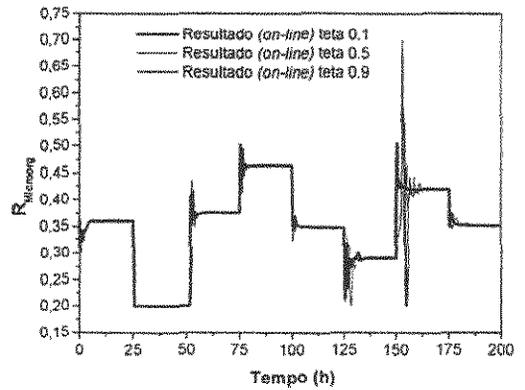
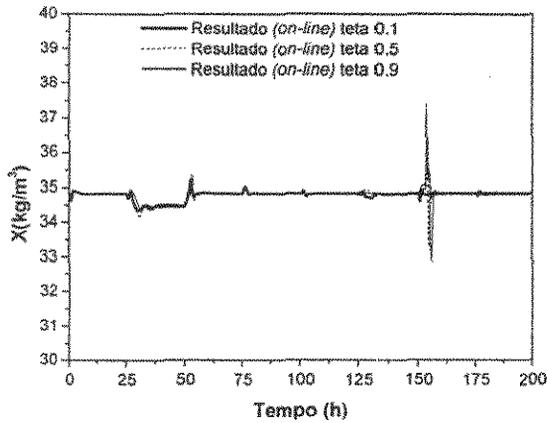
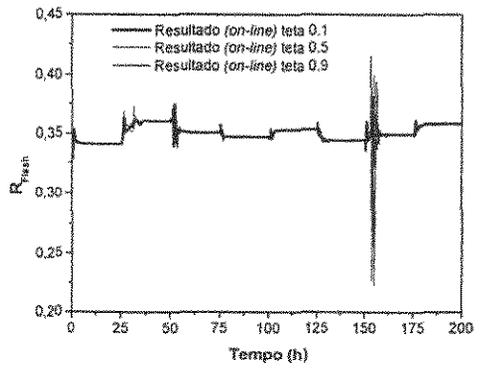
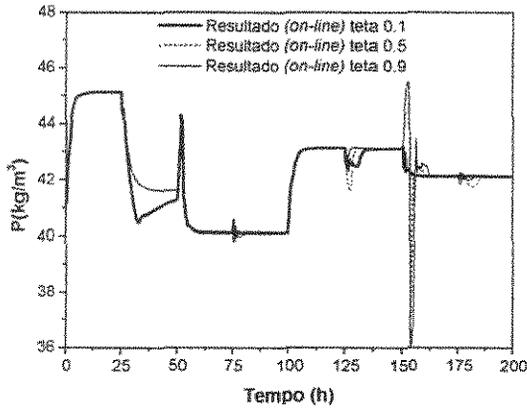
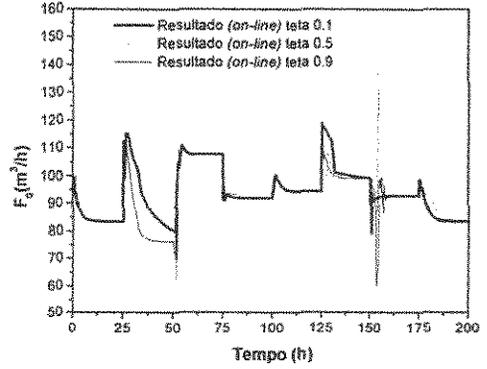
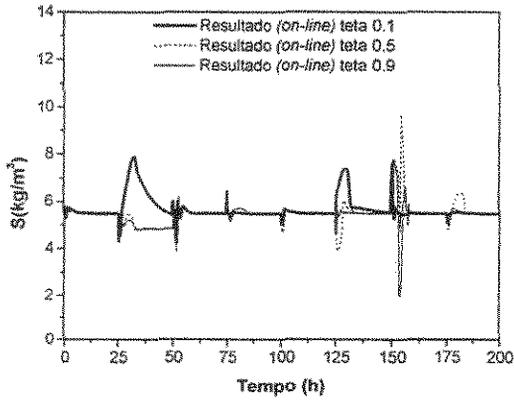
Problema do Tipo Servo e Regulador (Kalman 4) variável S



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

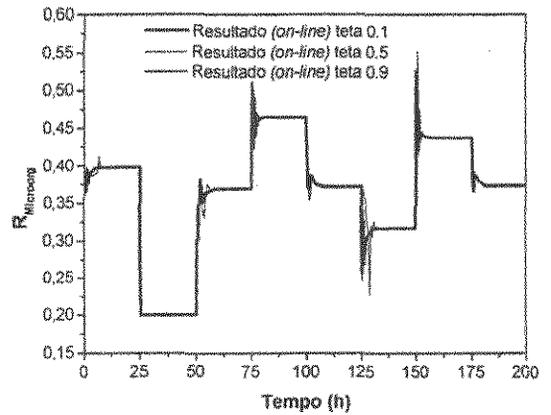
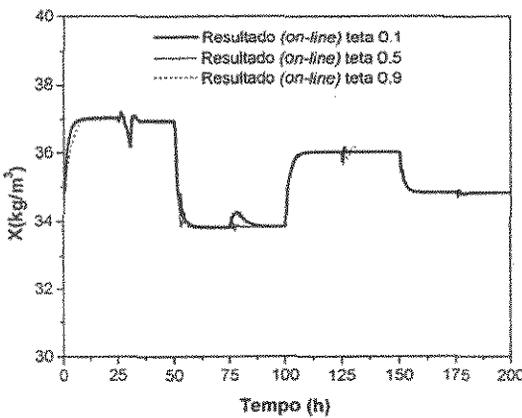
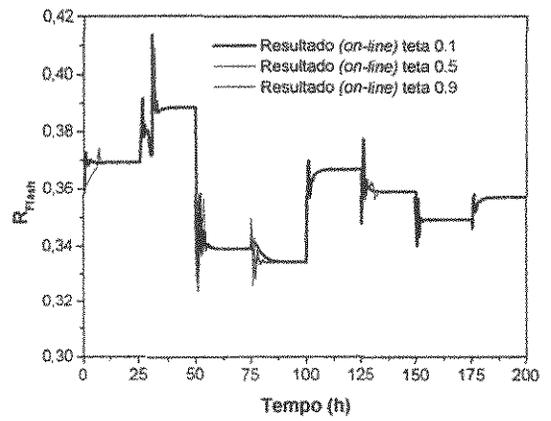
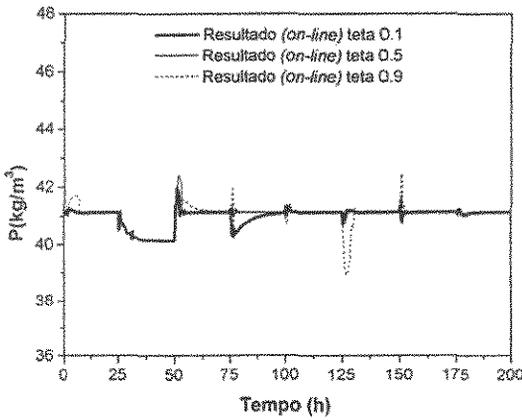
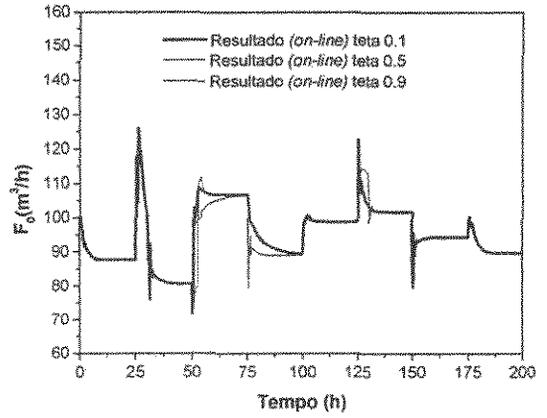
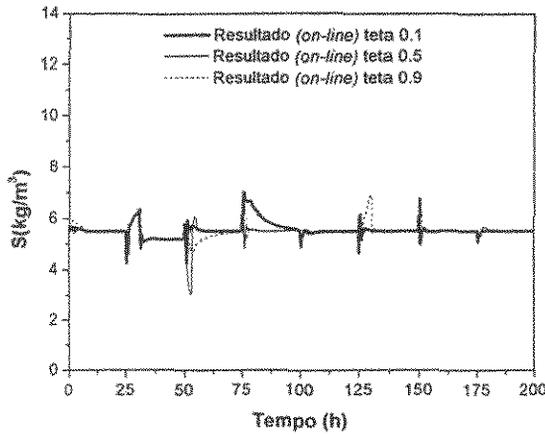
Problema do Tipo Servo e Regulador (Kalman 4) variável P



Resultado para Variáveis Controladas

Resultado para Variáveis Manipuladas

Problema do Tipo Servo e Regulador (Kalman 4) variável X



Resultado para Vari\u00e1veis Controladas

Resultado para Vari\u00e1veis Manipuladas