



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA QUÍMICA
DEPARTAMENTO DE PROCESSOS QUÍMICOS

PROJETO E APLICAÇÃO DE CONTROLADORES
BASEADOS EM MODELOS LINEARES, NEURAIS E
NEBULOSOS

LUIZ AUGUSTO DA CRUZ MELEIRO
Mestre em Engenharia Química - DEQ/UFSCar

Prof. Dr. Rubens Maciel Filho
Orientador

Campinas, dezembro de 2002



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA QUÍMICA
DEPARTAMENTO DE PROCESSOS QUÍMICOS

PROJETO E APLICAÇÃO DE CONTROLADORES
BASEADOS EM MODELOS LINEARES, NEURAIS E
NEBULOSOS

Tese de Doutorado apresentada à Faculdade
de Engenharia Química como parte dos re-
quisitos exigidos para a obtenção do título de
DOUTOR EM ENGENHARIA QUÍMICA.

LUIZ AUGUSTO DA CRUZ MELEIRO
Mestre em Engenharia Química - DEQ/UFSCar

Prof. Dr. Rubens Maciel Filho
Orientador

Campinas, dezembro de 2002

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

M482p Meleiro, Luiz Augusto da Cruz
Projeto e aplicação de controladores baseados em
modelos lineares, neurais e nebulosos / Luiz Augusto da
Cruz Meleiro. --Campinas, SP: [s.n.], 2002.

Orientador: Rubens Maciel Filho.
Tese (doutorado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Química.

1. Identificação de sistemas. 2. Controle preditivo. 3.
Controle automático. 4. Redes neurais (Computação). 5.
Algoritmos difusos. 6. Álcool. I. Maciel Filho, Rubens.
II. Universidade Estadual de Campinas. Faculdade de
Engenharia Química. III. Título.

Tese de Doutorado defendida por Luiz Augusto da Cruz Meleiro e aprovada em 17 de dezembro de 2002 pela banca examinadora constituída pelos doutores:

Prof. Dr. Rubens Maciel Filho - Orientador
Faculdade de Engenharia Química
Universidade Estadual de Campinas - UNICAMP

Prof. Dr. Wagner Caradori do Amaral
Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas - UNICAMP

Prof. Dr. Fernando José Von Zuben
Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas - UNICAMP

Prof. Dr. Maurício Bezerra de Souza Jr.
Escola de Química
Universidade Federal do Rio de Janeiro - UFRJ

Prof. Dr. Humberto Molinar Henrique
Faculdade de Engenharia Química
Universidade Federal de Uberlândia - UFU

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

Prof. Dr. Rubens Maciel Filho (Orientador)
Faculdade de Engenharia Química - UNICAMP

Prof. Dr. Wagner Caradori do Amaral
Faculdade de Engenharia Elétrica e de Computação - UNICAMP

Prof. Dr. Fernando José Von Zuben
Faculdade de Engenharia Elétrica e de Computação - UNICAMP

Prof. Dr. Maurício Bezerra de Souza Jr.
Escola de Química - Universidade Federal do Rio de Janeiro - UFRJ

Prof. Dr. Humberto Molinar Henrique
Faculdade de Engenharia Química - Universidade Federal de Uberlândia - UFU

Dedicatória

Dedico esta tese a Cristiane.

Por sua inestimável ajuda durante esses longos anos de tese, especialmente neste último, por seu enorme carinho, sua generosidade incomum e sua paciência inesgotável para enfrentar a minha rabugice com tanto amor e bom humor e, sobretudo, por sempre conseguir enxergar o que há de melhor em todas as coisas da vida.

Eu tive a sorte de encontrá-la. Duas vezes!

À memória de minha mãe e de meu avô, Aristides Meleiro.

Agradecimentos

Ao Prof. Dr. Rubens Maciel Filho, meu orientador, que durante os quatro anos de doutoramento proporcionou os meios necessários para que eu pudesse concluir a minha tese e realizar várias outras atividades que contribuíram para o meu aperfeiçoamento profissional e pessoal. Agradeço ainda pelo constante incentivo, confiança, respeito e amizade.

Ao Prof. Dr. Wagner Caradori do Amaral, a quem considero meu co-orientador, por sua atitude permanentemente criteriosa que contribuiu decisivamente para a qualidade final desta tese e dos trabalhos publicados na forma de artigos.

Ao Prof. Dr. Fernando José Von Zuben, pela intensa e profícua colaboração, especialmente na parte da tese relacionada aos conceitos e aplicações dos modelos neurais.

Aos colegas Aline, Eduardo “Ursão” e Ricardo Campello que, com o passar do tempo e depois de inúmeros cafezinhos, muitas cervejas e bate-papos, foram “promovidos” à categoria de grandes e queridos amigos.

Aos amigos Wilton, Nagel, Pedro e Osvaldo, pelos muitos *happy-hours* e grandes papos, onde nem sempre (ou quase nunca!) havia consenso!!!

A todos os outros amigos e colegas do LOPCA, por toda ajuda que recebi, nas incontáveis vezes, nos pequenos e grandes problemas.

Ao CENAPAD-SP/UNICAMP pelo suporte extra de *hardware* e *software*.

Ao CNPq pelo auxílio financeiro.

Por último e não menos importante, agradeço a todos os meus familiares e, sobretudo, aos meus pais e irmãos por todo amor que sempre recebi.

Resumo

O objetivo básico do presente trabalho é apresentar soluções alternativas para os principais problemas relacionados à identificação e ao controle de sistemas dinâmicos não lineares. Apresenta-se um estudo sobre a identificação de processos considerando diferentes estruturas não lineares: Os modelos neurais tradicionais (do tipo Perceptron Multicamadas), os recém desenvolvidos algoritmos neurais construtivos e os modelos nebulosos com estrutura hierárquica. Foi dado grande destaque ao principal problema apresentado pelos modelos não lineares do tipo caixa-preta, qual seja a necessidade da estimação de um grande número de parâmetros, principalmente quando se trata da identificação de sistemas de grande porte, pois o número de parâmetros a ser estimado cresce exponencialmente com o número de entradas do modelo. A utilização de modelos capazes de contribuir para contornar este problema, conhecido como “A Maldição da Dimensionalidade”, foi um dos pontos centrais considerados nesta tese. Utilizou-se como estudos de caso dois processos fermentativos para produção de etanol. Um deles, bastante representativo das usinas alcooleiras do Brasil, é composto por quatro fermentadores (dornas) ligados em série operando com reciclo de microorganismos. O outro processo apresenta uma configuração alternativa e utiliza apenas uma dorna acoplada a uma centrífuga, um tanque para tratamento das leveduras e um tanque *flash*, cuja principal função é extrair continuamente parte do etanol formado. Finalmente, apresenta-se o projeto de controladores preditivos (mono e multi-variáveis) baseados nos modelos não lineares previamente identificados. Verifica-se, através de simulações em malha fechada, que os algoritmos de controle não linear propostos neste trabalho provaram ser eficientes e robustos, pois forneceram bons resultados em problemas de controle dos tipos servo e regulatório.

Abstract

The basic purpose of this work is to present alternative solutions to the main problems related to the identification and control of non-linear dynamic systems. A study on process identification is presented considering different non-linear model structures: the traditional multi-layer perceptron models, the newly developed algorithms for constructive learning in neural networks, and the hierarchical neural fuzzy models. The main issue associated with the non-linear black-box models has received special attention: these models usually require the estimation of a large number of parameters. Generally, the number of parameters and data needed to provide a desired accuracy increases exponentially with the dimension of the input space of the mapping to be approximated. The use of models that can contribute to get around this problem, known as “Curse of Dimensionality”, was one of the key objectives of this work. Two fermentative processes for ethanol production were adopted as case studies. One of them is a typical large-scale industrial plant found in Brazil, and is composed of four tank reactors (fermentors) arranged in series and operated with cell recycling. The other case study is an extractive alcoholic fermentation process that presents an alternative configuration, combining just one fermentor with a centrifuge, a cell treatment unit, and a flash vessel to extract continuously part of the ethanol produced. Finally, the design and performance of mono and multi-variable predictive controllers based on the previously identified non-linear models are presented. The results obtained from closed-loop simulations attest the efficiency and robustness of the non-linear predictive controllers in servo and regulatory applications.

Sumário

Dedicatória	v
Agradecimentos	vi
Resumo	vii
Abstract	viii
Lista de Figuras	xvii
Lista de Tabelas	xxiii
1 INTRODUÇÃO	1
1.1 Organização da Tese	2
2 REDES NEURAIIS ARTIFICIAIS	5
2.1 Redes Neurais Artificiais <i>Feedforward</i> do tipo Perceptron Multicamada	6
2.2 Redes Neurais Artificiais Construtivas	10
2.2.1 Introdução	10
2.2.2 Aprendizado por Busca de Projeção (<i>Projection Pursuit Learning</i>)	12
2.2.2.1 Redes Neurais Artificiais via Métodos Construtivos . .	16

2.2.3	O Aprendizado Construtivo (<i>Unit-Growing Learning</i>) no Contexto de Redes Neurais Artificiais	20
2.2.3.1	O Método UGL	21
2.2.3.2	Determinação da Função de Ativação	27
2.2.3.3	Determinação das Projeções	29
2.2.3.4	O Procedimento de Retro-Ajuste	32
2.2.3.5	O Algoritmo de Aprendizado Construtivo	33
2.3	Considerações Finais Sobre o Capítulo	34
3	TÓPICOS EM MODELAGEM DE SISTEMAS	35
3.1	Introdução	36
3.2	Definições e Conceitos Fundamentais	39
3.2.1	Sistemas Lineares	39
3.2.2	Terminologia Básica	39
3.3	Representações Matemáticas Lineares de Tempo Discreto	42
3.3.1	Modelo de Resposta Finita ao Impulso (FIR)	42
3.3.2	Modelos com Funções de Base Ortonormal (OBF)	44
3.3.3	Modelo Auto-Regressivo com Entradas Externas (ARX)	48
3.3.4	Modelo Auto-Regressivo com Média Móvel e Entradas Externas (ARMAX)	49
3.3.5	Modelo de Erro na Saída (OE)	50
3.4	Representações Matemáticas Não Lineares de Tempo Discreto	52
3.4.1	Modelos NARX e NARMAX	52
3.4.2	Modelos com Funções de Base Radial	54
3.4.3	Modelos de Volterra	55
3.4.4	Modelos Nebulosos (<i>Fuzzy</i>)	56

3.4.4.1	Operações entre Conjuntos Nebulosos no mesmo Universo de Discurso	58
3.4.4.2	Operações entre Conjuntos Nebulosos em Universos de Discurso Distintos	59
3.4.4.3	Sistemas Nebulosos Lingüísticos	63
3.4.4.4	Sistemas Nebulosos Relacionais	65
3.4.4.5	Sistemas Nebulosos de Takagi-Sugeno	66
3.4.5	Modelos com Estrutura Hierárquica	67
3.4.5.1	Estrutura Relacional Nebulosa Simplificada	69
3.4.5.2	O Modelo Hierárquico Neuro-Nebuloso (HNN)	71
3.4.5.3	Formulação Matemática do Modelo HNN	72
3.4.5.4	A Estimação dos Parâmetros do Modelo HNN	73
3.5	Considerações Finais Sobre o Capítulo	74
4	IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS NÃO LINEARES UTILIZANDO MODELOS NEURAI	75
4.1	Introdução	77
4.2	O Procedimento Básico da Modelagem Neural	79
4.3	Realização de Experimentos	80
4.3.1	Determinação do Grau de Não Linearidade do Sistema	81
4.3.2	Escolha do Período de Amostragem	82
4.3.3	A “Maldição da Dimensionalidade”	82
4.3.4	Projeto do Sinal de Entrada	83
4.3.5	Preparação dos Dados para Modelagem	85
4.4	Seleção da Estrutura de Modelos Neurais	88
4.4.1	Os Regressores	88

4.4.2	Estruturas Neurais do tipo NNFIR e NNARX	90
4.4.3	Estrutura Neural do tipo NNARMAX	91
4.4.4	Estrutura Neural do tipo NNOE	91
4.5	Estimação de Modelos Neurais	95
4.5.1	Modelagem Direta	96
4.5.2	Modelagem Inversa	99
4.6	Validação de Modelos Neurais	102
4.6.1	Testes de Estimação do Erro de Generalização	103
4.6.2	Testes de Correlação ou Análise de Resíduos	104
4.6.3	Testes de Inspeção Visual	108
4.7	Reiniciando o Procedimento	111
4.7.1	Retreinamento da Rede Neural	111
4.7.2	Determinação da Arquitetura Neural Ótima	114
4.7.3	Realização de Novos Experimentos	115
4.8	Considerações Finais Sobre o Capítulo	115
5	CONTROLE PREDITIVO BASEADO EM MODELO	117
5.1	Introdução	118
5.1.1	Histórico da Estratégia MPC	119
5.1.2	A Estratégia de Controle Preditivo	122
5.2	Controle Preditivo Linear	126
5.2.1	Introdução	126
5.2.1.1	Modelos Lineares	126
5.2.2	MPCs Baseados em Modelos Lineares	127
5.2.2.1	O Controle Por Matriz Dinâmica - DMC	127

5.2.2.2	O Algoritmo DMC para Sistemas SISO	129
5.2.2.3	O Algoritmo DMC para Sistemas MIMO	137
5.3	Controle Preditivo Não Linear	141
5.3.1	Introdução	141
5.3.2	MPCs Baseados em Modelos Não Lineares	142
5.3.2.1	Formulação Geral do Controlador Preditivo Não Linear	145
5.4	Considerações Finais Sobre o Capítulo	147
6	ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO AL- COÓLICA CONTÍNUA	149
6.1	Introdução	151
6.2	Descrição do Processo	156
6.2.1	Discussões Preliminares	156
6.2.2	Modelagem Matemática	160
6.2.2.1	Balanco de Massa para cada Reator	161
6.2.2.2	Balancos de Energia	163
6.2.2.3	Equações para o Reciclo	166
6.3	Estratégias de Identificação para o Processo de Fermentação Contínua .	169
6.3.1	Determinação das Variáveis de Entrada, Saída e Perturbação . .	170
6.3.2	Geração dos Dados para Identificação do Processo	171
6.3.3	Determinação do Período de Amostragem	172
6.3.4	Determinação da Estrutura dos Modelos	173
6.3.5	Estimação e Validação dos Modelos	176
6.3.6	Identificação Utilizando Modelos Hierárquicos HNN	177
6.3.7	Identificação Utilizando Modelos Neurais MLP	181
6.3.8	Identificação Utilizando o Modelo Neural Construtivo UGL . . .	185

6.3.9	Identificação Utilizando o Modelo Linear ARX	191
6.4	Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável	196
6.4.1	Formulação Geral do Controlador Preditivo Não Linear Mono-Variável	198
6.4.2	Controle Preditivo Não Linear Baseado no Modelo HNN: MPC-HNN	201
6.4.3	Controle Preditivo Não Linear Baseado no Modelo MLP: MPC-MLP	206
6.4.4	Controle Preditivo Não Linear Baseado no Modelo UGL: MPC-UGL	210
6.4.5	Controle Preditivo Linear: O Controlador por Matriz Dinâmica - DMC	215
6.5	Conclusões	218
7	ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA	221
7.1	Introdução	223
7.2	Descrição do Processo	225
7.2.1	Modelagem Matemática	227
7.3	Estratégias de Identificação para o Processo Extrativo de Fermentação Alcoólica	234
7.3.1	Determinação das Variáveis de Entrada, Saída e Perturbação . .	234
7.3.1.1	Variáveis de Entrada	235
7.3.1.2	Variáveis de Saída	235
7.3.1.3	Variáveis Perturbação	235
7.3.2	Geração dos Dados para Identificação do Processo	236

7.3.3	Determinação do Período de Amostragem	236
7.3.4	Determinação da Estrutura dos Modelos	237
7.3.5	Estimação e Validação dos Modelos	237
7.3.6	Identificação Utilizando Modelos Neurais MLP	239
7.3.7	Identificação Utilizando o Modelo Neural Construtivo UGL	243
7.3.8	Identificação Utilizando o Modelo Linear ARX	249
7.4	Estratégias de Controle para o Processo Extrativo de Fermentação Al- coólica: Caso Multi-Variável	253
7.4.1	Formulação Geral do Controlador Preditivo Não Linear Multi- Variável	254
7.4.2	Controle Preditivo Não Linear Baseado no Modelo MLP	258
7.4.3	Controle Preditivo Não Linear Baseado no Modelo UGL	265
7.4.4	Controle Preditivo Linear: O Controlador por Matriz Dinâmica - DMC	272
7.5	Conclusões	278
8	CONSIDERAÇÕES FINAIS	279
8.1	Conclusões	279
8.2	Perspectivas Futuras	281
	ÍNDICE DE AUTORES	283
	REFERÊNCIAS BIBLIOGRÁFICAS	289

Lista de Figuras

2.1	Projeções bidimensionais de um conjunto de dados tridimensional. . . .	13
2.2	Rede neural resultante do processo de aprendizado construtivo.	18
3.1	Exemplo de conjuntos clássico e nebuloso para expressar as temperaturas “quente” e “fria”.	57
3.2	Funções de pertinência de três conjuntos nebulosos para a temperatura de um reator.	58
3.3	(a) Modelo Convencional; (b) Modelo Hierárquico.	67
4.1	Procedimento básico para identificação de sistemas.	79
4.2	Sinal de entrada com nível de ativação variando em intervalos constantes.	84
4.3	Sinal de entrada com nível de ativação variando em intervalos aleatórios.	85
4.4	Rede neural com estrutura NNFIR.	90
4.5	Rede neural com estrutura NNARX.	90
4.6	Rede neural com estrutura NNARMAX.	92
4.7	Rede neural com estrutura NNOE.	92
4.8	Identificação de uma planta utilizando RNA.	96
4.9	Estrutura de identificação série-paralelo.	97
4.10	Estrutura da dinâmica da planta.	98
4.11	Modelagem neural direta inversa.	100

4.12	Modelagem inversa especializada.	100
4.13	Predições um passo adiante (a) e simulação recursiva (b).	109
5.1	Representação esquemática dos elementos básicos do controle preditivo.	122
5.2	Representação do princípio do horizonte móvel: Situação no tempo t	125
5.3	Representação do princípio do horizonte móvel: Situação no tempo $t + 1$	125
5.4	Representação esquemática do modelo de convolução e seus coeficientes: Resposta do sistema a um degrau na variável de entrada.	129
5.5	Estrutura do controlador preditivo não linear utilizado nos capítulos 6 e 7.	145
6.1	Fluxograma básico do processo de produção de açúcar e álcool.	157
6.2	Representação esquemática da planta de fermentação alcoólica contínua.	159
6.3	Modelo hierárquico neuro-nebuloso para predição de S_4	175
6.4	Estrutura dos modelos NNARX.	176
6.5	Desempenho do modelo HNN para a variável S_4	178
6.6	Desempenho do modelo HNN para a variável P_4	179
6.7	Desempenho do modelo HNN para a variável X_4	179
6.8	Desempenho do modelo MLP com $n_{l_e} = 10$ para a variável S_4	183
6.9	Desempenho do modelo MLP com $n_{l_e} = 10$ para a variável P_4	183
6.10	Desempenho do modelo MLP com $n_{l_e} = 10$ para a variável X_4	184
6.11	Simulação recursiva do modelo MLP com $n_{l_e} = 3$	184
6.12	Simulação recursiva do modelo MLP com $n_{l_e} = 5$	184
6.13	Desempenho do modelo UGL com $n_{l_e} = 4$ para a variável S_4	187
6.14	Desempenho do modelo UGL com $n_{l_e} = 4$ para a variável P_4	187
6.15	Desempenho do modelo UGL com $n_{l_e} = 4$ para a variável X_4	188
6.16	Funções de ativação do neurônio 1.	189

6.17 Funções de ativação do neurônio 2.	189
6.18 Funções de ativação do neurônio 3.	190
6.19 Funções de ativação do neurônio 4.	190
6.20 Desempenho do modelo ARX para a variável S_4	193
6.21 Desempenho do modelo ARX para a variável P_4	194
6.22 Desempenho do modelo ARX para a variável X_4	194
6.23 Controlador MPC-HNN - Problema Servo: Referência e saída do processo (acima); variável manipulada (abaixo).	202
6.24 Perturbações na taxa de reciclo.	202
6.25 Controlador MPC-HNN - Problema Regulador: Referência e saída do processo (acima); variável manipulada (abaixo).	203
6.26 Perturbações na concentração de S_0	203
6.27 Controlador MPC-HNN - Problema Servo (S_0 variando em torno de 175 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).	204
6.28 Controlador MPC-HNN - Problema Servo (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).	204
6.29 Controlador MPC-HNN - Problema Servo (S_0 variando em torno de 185 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).	205
6.30 Controlador MPC-HNN - Problema Regulador (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).	205
6.31 Controlador MPC-MLP - Problema Servo: Referência e saída do processo (acima); variável manipulada (abaixo).	206
6.32 Controlador MPC-MLP - Problema Regulador: Referência e saída do processo (acima); variável manipulada (abaixo).	207

6.33	Controlador MPC-MLP - Problema Servo (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).	208
6.34	Controlador MPC-MLP - Problema Regulador (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).	209
6.35	Controlador MPC-UGL - Problema Servo: Referência e saída do processo (acima); variável manipulada (abaixo).	210
6.36	Controlador MPC-UGL - Problema Regulador: Referência e saída do processo (acima); variável manipulada (abaixo).	211
6.37	Controlador MPC-UGL - Problema Servo (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).	212
6.38	Controlador MPC-UGL - Problema Regulador (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).	212
6.39	Controlador MPC-UGL - Problema Servo para a Concentração de Etanol: Referência e saída do processo (acima); variável manipulada (abaixo).213	
6.40	Controlador MPC-UGL - Problema Regulador para a Concentração de Etanol: Referência e saída do processo (acima); variável manipulada (abaixo).	213
6.41	Controlador MPC-UGL - Problema Servo para P_4 (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).	214
6.42	Controlador MPC-UGL - Problema Regulador para P_4 (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); var. manipulada (abaixo).	214
6.43	Controlador DMC - Problema Servo: Referência e saída do processo (acima); variável manipulada (abaixo).	216

6.44	Controlador DMC - Problema Regulador: Referência e saída do processo (acima); variável manipulada (abaixo).	216
6.45	Controlador DMC - Problema Servo (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo). .	217
6.46	Controlador DMC - Problema Regulador (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo). .	217
7.1	Representação esquemática da planta de fermentação alcoólica extrativa.	225
7.2	Estrutura dos modelos neurais do tipo NNARX.	238
7.3	Desempenho do modelo MLP com $n_{l_e} = 10$ para a variável S	241
7.4	Desempenho do modelo MLP com $n_{l_e} = 10$ para a variável P	241
7.5	Desempenho do modelo MLP com $n_{l_e} = 10$ para a variável X	242
7.6	Desempenho do modelo UGL com $n_{l_e} = 5$ para a variável S	244
7.7	Desempenho do modelo UGL com $n_{l_e} = 5$ para a variável P	245
7.8	Desempenho do modelo UGL com $n_{l_e} = 5$ para a variável X	245
7.9	Funções de ativação do neurônio 1.	246
7.10	Funções de ativação do neurônio 2.	246
7.11	Funções de ativação do neurônio 3.	247
7.12	Funções de ativação do neurônio 4.	247
7.13	Funções de ativação do neurônio 5.	248
7.14	Desempenho do modelo linear ARX para a variável S	251
7.15	Desempenho do modelo linear ARX para a variável P	251
7.16	Desempenho do modelo linear ARX para a variável X	252
7.17	Perturbações agindo sobre o processo extrativo de fermentação alcoólica.	257
7.18	Controlador MPC-MLP - Problemas servo para a variável S e regulatório para as variáveis P e X	258
7.19	Controlador MPC-MLP - Variáveis manipuladas referentes à figura 7.18.	259

7.20	Controlador MPC-MLP - Problemas servo para a variável P e regulatório para as variáveis S e X	260
7.21	Controlador MPC-MLP - Variáveis manipuladas referentes à figura 7.20.	261
7.22	Controlador MPC-MLP - Problemas servo para a variável X e regulatório para as variáveis S e P	262
7.23	Controlador MPC-MLP - Variáveis manipuladas referentes à figura 7.22.	263
7.24	Controlador MPC-UGL - Problemas servo para a variável S e regulatório para as variáveis P e X	265
7.25	Controlador MPC-UGL - Variáveis manipuladas referentes à figura 7.24.	266
7.26	Controlador MPC-UGL - Problemas servo para a variável P e regulatório para as variáveis S e X	267
7.27	Controlador MPC-UGL - Variáveis manipuladas referentes à figura 7.26.	268
7.28	Controlador MPC-UGL - Problemas servo para a variável X e regulatório para as variáveis S e P	269
7.29	Controlador MPC-UGL - Variáveis manipuladas referentes à figura 7.28.	270
7.30	Controlador DMC - Problemas servo para a variável S e regulatório para as variáveis P e X	272
7.31	Controlador DMC - Variáveis manipuladas referentes à figura 7.30. . .	273
7.32	Controlador DMC - Problemas servo para a variável P e regulatório para as variáveis S e X	274
7.33	Controlador DMC - Variáveis manipuladas referentes à figura 7.32. . .	275
7.34	Controlador DMC - Problemas servo para a variável X e regulatório para as variáveis S e P	276
7.35	Controlador DMC - Variáveis manipuladas referentes à figura 7.34. . .	277

Lista de Tabelas

3.1	Número de parâmetros, m , dos modelos convencional e HNN em função do número de entradas, n , para $c = 5$	72
6.1	Parâmetros cinéticos do modelo (para <i>Saccharomyces cerevisiae</i>).	163
6.2	Estado estacionário de referência.	165
6.3	Parâmetros de projeto.	165
6.4	Parâmetros do modelo fenomenológico.	165
6.5	EQM de modelagem para os dados de validação - modelos HNN.	177
6.6	EQMN de modelagem para os dados de validação - modelos HNN.	178
6.7	EQM de modelagem para os dados de validação - modelos MLP.	181
6.8	EQMN de modelagem para os dados de validação - modelos MLP.	182
6.9	EQM de modelagem para os dados de validação - modelo UGL.	186
6.10	EQMN de modelagem para os dados de validação - modelo UGL.	186
7.1	Constantes físicas da equação de energia.	232
7.2	Valores do parâmetros cinéticos em função da temperatura (em °C).	233
7.3	EQM de modelagem para os dados de validação - Modelos MLP.	239
7.4	EQMN de modelagem para os dados de validação - Modelos MLP.	240
7.5	EQM de modelagem para os dados de validação - Modelo UGL.	243
7.6	EQMN de modelagem para os dados de validação - Modelo UGL.	244

Capítulo 1

INTRODUÇÃO

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 eles não são adequados para descrever processos com a multiplicidade de estados estacionários e as não linearidades característicos de grande parte dos processos da Engenharia Química.

Os recentes desenvolvimentos nos processos de produção e a crescente exigência de plantas com alta eficiência impõem restrições cada vez maiores aos seus sistemas de controle associados. Ao projetar sistemas de controle para operações de processos com alta eficiência e desempenho, deve-se garantir que o controlador seja capaz de atender as especificações de produção sem violar as restrições. A literatura tem mostrado que é necessário o desenvolvimento de algoritmos de controle avançado que sejam capazes de operar o sistema com elevados níveis de desempenho, conciliado a condições seguras. Com a crescente necessidade de se desenvolverem controladores de processos com desempenho cada vez melhores, várias estratégias de controle avançado têm sido propostas e, em processos da indústria química, destacam-se as aplicações da estratégia preditiva.

Dentre os desenvolvimentos mais recentes em sistemas de controle avançado, destacam-se aqueles caracterizados pela forte ligação entre as áreas de modelagem, utilizando principalmente as redes neurais artificiais e a lógica nebulosa (*fuzzy*), e o

projeto de sistemas de controle baseados em modelo.

Um dos objetivos desta tese é formalizar o procedimento para o uso de modelos não lineares, com ênfase nos modelos baseados em redes neurais artificiais e sistemas nebulosos, como uma estrutura de modelo genérica para a identificação de processos dinâmicos não lineares e o seu posterior uso no projeto de sistemas de controle avançado.

A utilização de modelos neurais e nebulosos com dimensão reduzida (número reduzido de parâmetros) é o principal foco da parte relacionada com a modelagem de processos. Duas estruturas não lineares são as principais alternativas propostas como novos modelos para os processos da indústria química: O modelo neural obtido pelo método de aprendizado construtivo e o modelo nebuloso com estrutura hierárquica.

Todos os controladores utilizados nesta tese foram projetados utilizando uma única estrutura de controle básica, que se mostrou robusta e flexível o suficiente para considerar o uso de qualquer tipo de modelo, lineares ou não lineares, podendo ainda ser adaptada para problemas mono e multi-variáveis.

A estratégia de controle preditivo não linear, baseada nos modelos neurais e nebulosos, apresentou um desempenho muito bom em todas as situações a que foi submetida, levando a crer que a sua utilização pode contribuir para a solução de alguns dos principais problemas de modelagem e de controle de processos da indústria química.

1.1 Organização da Tese

Apresentam-se a seguir, de forma sucinta, os principais assuntos abordados neste trabalho:

O capítulo 2 apresenta o principal modelo não linear utilizado nesta tese, as redes neurais artificiais. Inicia-se o capítulo com uma breve descrição das redes neurais artificiais e da formulação básica do modelo neural MLP, seguidos de uma rápida análise de alguns dos mais importantes algoritmos desenvolvidos para estimar os pesos de uma rede neural durante a etapa de treinamento. O ponto central deste capítulo é a apresentação da base teórica de um novo paradigma neural, as redes neurais construtivas, que despontam como uma classe de modelos não lineares bastante promissora.

1.1 Organização da Tese

Apresenta-se, no capítulo 3, uma breve revisão de alguns dos modelos utilizados com mais frequência na identificação de sistemas dinâmicos de tempo discreto lineares e não lineares. Outros modelos empregados com menos frequência na literatura são abordados neste capítulo devido à sua importância nas estratégias de identificação e controle propostas nesta tese. Neste contexto enquadra-se a estrutura hierárquica nebulosa, que foi utilizada no capítulo 6 como uma alternativa viável para fornecer modelos com dimensão reduzida e adequados para o uso em estratégias de controle baseadas em modelo.

O capítulo 4 teve como objetivo formalizar o procedimento geral de identificação de sistemas dinâmicos não lineares, tendo como modelo básico as redes neurais artificiais. No entanto, este procedimento pode ser aplicado a outros modelos não lineares, alguns dos quais descritos na seção 3.4.

O capítulo 5 enfoca a estratégia de controle preditivo baseada em modelo (MPC), utilizada para projetar todos os controladores desta tese. Apresenta-se um histórico sobre a evolução deste tipo de controlador e os motivos de sua grande aplicabilidade em processos da indústria química. Os conceitos e os elementos básicos dos controladores preditivos são discutidos em detalhes, tanto os algoritmos lineares (com ênfase no algoritmo DMC¹), quanto os algoritmos não lineares.

No capítulo 6, são apresentados os resultados da aplicação das técnicas lineares e não lineares de identificação e o controle do processo fermentativo mono-variável. Destaca-se a importância dos processos biotecnológicos para a sociedade contemporânea, representados neste trabalho pelo etanol. Também é apresentado nesta seção um breve histórico sobre a evolução do processo de produção do etanol associado ao contexto sócio-econômico característico de cada década, desde sua origem até os dias atuais. A planta, suas variáveis de interesse e suas faixas de operação são descritas, assim como a formulação do problema de controle. O procedimento de identificação do processo fermentativo é ilustrado utilizando-se três modelos não lineares (Hierárquico Neuro Nebuloso - HNN, Neural - MLP e Neural Construtivo - UGL) e um modelo linear do tipo ARX. Finalmente são apresentados os resultados do controle do processo fermentativo através da comparação entre os desempenhos dos controladores projetados com cada um dos modelos do processo.

¹Controle por Matriz Dinâmica (*Dynamic Matrix Control*) .

O capítulo 7 também trata da identificação e do controle de um processo de produção de etanol por via fermentativa. As principais diferenças em relação ao capítulo 6 são a configuração dos equipamentos e o número de variáveis controladas e manipuladas. A planta, neste caso, é composta por apenas uma dorna de fermentação acoplada a um tanque *flash*. As principais vantagens apresentadas por esta nova configuração são o aumento de produtividade em relação ao processo fermentativo convencional e a economia de energia, decorrente da ausência de trocadores de calor. Destaca-se a importância do desenvolvimento de novas técnicas que contribuam para o aumento da eficiência do processo de produção do etanol, para torná-lo economicamente mais competitivo em relação aos combustíveis tradicionais. O processo fermentativo é identificado utilizando-se dois modelos não lineares (Modelo Neural Convencional - MLP e Modelo Neural Construtivo - UGL) e um modelo linear do tipo ARX, todos utilizando a representação MIMO. Finalmente, apresentam-se os resultados obtidos quando os controladores preditivos resultantes (MPC-MLP, MPC-UGL e DMC) são testados em situações de controle multi-variável considerando, simultaneamente, os problemas servo e regulador para as variáveis controladas do processo.

As conclusões gerais sobre os resultados obtidos na tese e algumas sugestões para o desenvolvimento de trabalhos futuros são apresentadas no capítulo 8.

Capítulo 2

REDES NEURAIS ARTIFICIAIS

Preâmbulo

Apesar das redes neurais artificiais *feedforward* do tipo perceptron multicamada (MLP) serem a estrutura neural básica utilizada nesta tese, este tipo de modelo será apenas brevemente apresentado na seção 2.1. Esta estratégia foi escolhida pelo fato de já existir uma vasta e excelente literatura enfocando redes neurais artificiais do tipo MLP, abordando o tema tão ampla e profundamente quanto se queira. Assim, optou-se por aprofundar, neste capítulo, os conceitos sobre uma recente (e por isso, menos difundida) e potencialmente revolucionária abordagem de síntese de redes neurais *feedforward*, as *Redes Neurais Artificiais Construtivas*. Outro motivo que justifica o enfoque deste capítulo na abordagem construtiva é que a utilização deste tipo de modelo nas tarefas de identificação e controle de processos da indústria química é uma das principais inovações propostas nesta tese.

A seção 2.1 apresenta um breve histórico das redes neurais artificiais e a formulação básica do modelo MLP. Apresenta-se também nesta seção uma breve análise de alguns dos mais importantes algoritmos desenvolvidos para estimar os pesos de uma rede neural durante a etapa de treinamento.

O novo paradigma neural, o *modelo neural construtivo*, é apresentado na seção 2.2; sua base teórica - o conceito de busca por projeção (*projection pursuit*) - é discutida na seção 2.2.2 e uma formulação específica para os modelos neurais utilizando este conceito é apresentada na seção 2.2.2.1. Um aperfeiçoamento dos modelos neu-

rais construtivos, proposto anteriormente por BÄRMANN e BIEGLER-KÖNIG (1992), é apresentado na seção 2.2.3. A nova forma de representação do modelo de projeção, proposta por VON ZUBEN e NETTO (1995), gerou o algoritmo de aprendizado construtivo (UGL) capaz de construir modelos neurais não paramétricos com arquitetura otimizada. O algoritmo UGL é descrito sistematicamente nas seções 2.2.3.1 a 2.2.3.4 e resumido na forma de um algoritmo na seção 2.2.3.5.

2.1 Redes Neurais Artificiais *Feedforward* do tipo Perceptron Multicamada

No início dos anos 40, MCCULLOCH e PITTS (1943) propuseram um modelo matemático de um neurônio que passou a representar a unidade básica das redes neurais artificiais que foram implementadas nos anos que se seguiram. Outros pesquisadores, como HEBB (1949), se preocuparam com as leis de adaptação relacionadas com os sistemas neurais. ROSENBLATT (1958) aprimorou o modelo criado por MCCULLOCH e PITTS (1943), gerando uma arquitetura que implementava uma regra de aprendizado baseada na minimização de uma função de erro quadrática. O elemento fundamental resultante foi chamado de *Perceptron* e recebeu muita atenção por parte dos pesquisadores desta área. Posteriormente, MINSKY e PAPPERT (1969) realizaram uma análise rigorosa do perceptron, demonstrando a sua limitação. O resultado deste estudo causou um grande desinteresse nas pesquisas no campo das redes neurais artificiais. Somente nos anos 80, com o trabalho de HOPFIELD (1982) sobre redes neurais recorrentes e de KOHONEN (1982) sobre os mapas auto-organizáveis, é que se verificou o ressurgimento efetivo e definitivo das redes neurais artificiais. Em 1986, o grupo de processamento paralelo e distribuído - de RUMELHART e colaboradores - publicou uma série de resultados sobre o algoritmo de aprendizado por retropropagação para modelos neurais, o *back-propagation* (RUMELHART *et al.*, 1986), que impulsionou fortemente o desenvolvimento de modelos conexionistas. Este algoritmo se mostrou indispensável

2.1 Redes Neurais Artificiais *Feedforward* do tipo Perceptron Multicamada

para o desenvolvimento de redes neurais multicamadas, fazendo com que as limitações dos modelos neurais (representados até então pelo perceptron) apontadas por MINSKY e PAPPERT (1969) fossem superadas. Este trabalho proporcionou um grande impulso para a área e influenciou durante muito tempo as pesquisas neste campo. Uma excelente coleção de artigos sobre o desenvolvimento de modelos neurais pode ser encontrado no livro de ANDERSON e ROSENFELD (1988).

As redes neurais do tipo MLP são provavelmente o tipo de modelo neural mais utilizado nas tarefas de identificação e controle de sistemas dinâmicos não lineares. Uma das razões para este sucesso está relacionada ao fato destas estruturas serem capazes de modelar adequadamente tanto relações funcionais simples quanto complexas. Esta habilidade tem sido verificada através de um número significativo de aplicações teóricas (simulações) e práticas (NØRGAARD *et al.*, 2000).

As redes neurais utilizadas com mais frequência são as do tipo Perceptron Multicamadas com estrutura *feedforward*. Nestas redes neurais, todos os neurônios de uma camada se conectam com todos os neurônios da camada seguinte, não havendo conexão com neurônios da mesma camada ou camadas anteriores. Tais modelos apresentam a importante característica de serem aproximadores universais.

A saída de um único neurônio com m entradas é dada por:

$$x = f \left(\sum_{j=1}^m \omega_j x_j + b \right) \quad (2.1)$$

onde o *bias* (ou polarização) b e os pesos ω_j são constantes (ajustados durante o treinamento e mantidos fixos durante a operação) e f é uma função de ativação não linear. Há vários tipos de função de ativação, porém a utilizada com mais frequência é do tipo sigmóide:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$

Nas redes neurais multicamadas com estrutura *feedforward*, a saída de um neurônio (equação (2.2)) é conectada à entrada de outro neurônio, assim, uma rede

neural com uma única saída e apenas uma camada oculta é descrita por uma função não linear nos parâmetros dada por:

$$y(k) = f_s \left\{ \sum_{i=1}^r \omega_i f_i \left(\sum_{j=1}^m \omega_{ij} x_j + b_i \right) + b_s \right\} \quad (2.3)$$

onde f_s é a função de ativação e b_s é o termo de polarização do neurônio da camada de saída; ω_i são os pesos da saída de cada neurônio da camada oculta e ω_{ij} são os pesos de conexão da entrada j ao i -ésimo neurônio da camada oculta.

Como consequência da saída da rede neural ser descrita por uma função não linear, seus parâmetros devem ser determinados por algoritmos específicos para estimação não linear. Existem muitos tipos de algoritmos específicos para a estimação de parâmetros de redes neurais, dentre os quais destaca-se o método do gradiente como o algoritmo utilizado com maior frequência (embora não seja o mais eficiente).

O treinamento supervisionado de redes neurais artificiais com várias camadas envolve uma etapa crucial - a determinação dos pesos das conexões sinápticas - que pode ser vista como um problema de otimização não linear irrestrita, onde uma função do erro global é minimizada através do ajuste dos parâmetros (pesos) da rede neural. DE CASTRO (1998) implementou (incluindo refinamentos às propostas já existentes) alguns dos mais citados algoritmos de otimização específicos para o treinamento supervisionado de redes neurais, dando ênfase aos métodos que recorrem à informação de 2ª ordem, pois tais métodos são atualmente considerados os mais eficientes para o treinamento supervisionado de redes neurais do tipo MLP.

Segundo VON ZUBEN (1996) e DE CASTRO (1998), existe um consenso em torno dos métodos de otimização chamados *Métodos do Gradiente Conjugado*, pois eles são capazes de tratar de problemas de otimização de grande escala de maneira efetiva. Contudo, o método do *Gradiente Conjugado* (GC), a exemplo da maioria dos métodos de otimização de 2ª ordem, utiliza um procedimento de busca unidimensional para a determinação da taxa de aprendizagem. Este procedimento envolve um grande número de avaliações da função ou de sua derivada, o que torna o processo bastante custoso do ponto de vista computacional. Uma variação do método GC proposto por MØLLER (1993) - o método do *Gradiente Conjugado Escalonado* (GCE) - evita a busca unidi-

2.1 Redes Neurais Artificiais *Feedforward* do tipo Perceptron Multicamada

mensional a cada iteração utilizando a abordagem de Levenberg-Marquardt para fazer um escalonamento do passo de ajuste. O método do *Gradiente Conjugado Escalonado Modificado* (GCEM) foi implementado por DE CASTRO e VON ZUBEN (1998a) a partir de um resultado obtido por PEARLMUTTER (1994) e caracteriza um importante avanço em relação aos métodos convencionais, pois permite o cálculo exato da informação de 2ª ordem com custo computacional associado de mesma ordem que o exigido para o cálculo da informação de 1ª ordem (VON ZUBEN, 1996).

Assim como os demais modelos não lineares, as redes neurais artificiais também apresentam o problema relacionado com a determinação da estrutura dos modelos, ou seja, deve-se determinar *a priori* os regressores que serão utilizados na estrutura (ou topologia) da rede neural. Este problema envolve também a escolha do número de neurônios da(s) camada(s) oculta(s), o número de camadas e o tipo da função de ativação. Trabalhos recentes desenvolvidos por VON ZUBEN (1996), VON ZUBEN e NETTO (1995; 1997) e LIMA (2000), entre outros, apresentam um novo paradigma de redes neurais artificiais (baseado na abordagem construtiva com aprendizado pelo critério de busca por direções de projeção) como uma solução eficiente para estes problemas. Esta abordagem, apresentada na seção 2.2, é empregada no presente trabalho como uma das alternativas para a identificação de processos não lineares.

2.2 Redes Neurais Artificiais Construtivas

2.2.1 Introdução

As redes neurais artificiais (RNA) têm sido largamente utilizadas para a identificação e controle de sistemas dinâmicos não lineares. Uma das principais razões para este sucesso é a capacidade de aproximação universal dos modelos neurais, isto é, tais modelos são capazes de aproximar com precisão arbitrária qualquer mapeamento contínuo definido em um domínio compacto (HORNIK *et al.*, 1989; KOSKO, 1992). Contudo, devido à sua estrutura genérica, as RNA normalmente requerem a estimação de um grande número de parâmetros. Problemas relacionados com os procedimentos computacionais necessários para alcançar bons resultados, incluindo a definição da dimensão da rede neural, a escolha das funções de ativação e a busca pelo conjunto de pesos ótimo, sempre representaram um obstáculo para o uso mais amplo das redes neurais artificiais (KOSKO, 1997; HAYKIN, 1999). Outro aspecto a ser considerado é a capacidade de generalização associada às técnicas de aprendizado supervisionado quando aplicados a aproximadores universais. Apesar de serem bastante utilizadas em problemas de aproximação de funções e identificação de processos dinâmicos, redes neurais do tipo MLP podem apresentar-se mal-condicionadas, caso a sua dimensão não seja determinada de forma adequada. Isto está relacionado ao fato de que redes neurais com dimensão muito reduzida podem não exibir flexibilidade suficiente para lidar com certos problemas e, por outro lado, redes com dimensão excessiva geralmente exibem pouca capacidade de generalização (GEMAN *et al.*, 1992).

Até recentemente, a determinação da dimensão adequada de uma rede neural para uma dada aplicação dependia da experiência do usuário em (exaustivos) procedimentos de tentativa e erro, fato que impulsionou o desenvolvimento de algoritmos que fossem capazes de determinar automaticamente a arquitetura da rede neural. Atualmente, o projeto automático da arquitetura de uma rede neural vem se tornando parte do procedimento de treinamento, através de uma exploração mais eficiente da informação disponível para o aprendizado supervisionado (DE CASTRO *et al.*, 1999).

Os dois principais métodos desenvolvidos para realizar esta tarefa são os *Métodos de Poda* (LE CUN *et al.*, 1990; HASSIBI e STORK, 1993) e os *Métodos Construtivos*

2.2 Redes Neurais Artificiais Construtivas

(BÄRMANN e BIEGLER-KÖNIG, 1992; ROOSEN e HASTIE, 1994; VON ZUBEN e NETTO, 1995). A estratégia empregada nos métodos de poda é determinar, já de início, uma arquitetura neural que seja (em princípio) grande o suficiente para descrever o sistema de interesse e, a partir desta arquitetura, os pesos cuja contribuição para o desempenho da rede não for significativa são gradualmente eliminados até que uma arquitetura mais parcimoniosa seja alcançada. O procedimento geralmente adotado por estes métodos é classificar os pesos segundo algum critério de modo que o algoritmo tenha algum indicativo sobre quais pesos devem ser eliminados. Dois dos algoritmos de poda mais empregados são o *Optimal Brain Surgeon* e o *Optimal Brain Damage* (NØRGAARD *et al.*, 2000). O grande inconveniente desta abordagem é que, se existir mais de uma arquitetura adequada para solucionar o problema de aproximação de um dado mapeamento, o algoritmo sempre convergirá para a solução de maior dimensão (VON ZUBEN, 1996).

Em contrapartida, a essência dos métodos construtivos é iniciar uma rede neural com dimensão reduzida (geralmente com um neurônio na camada intermediária) e ir adicionando unidades extras até que uma solução adequada seja obtida. A determinação da arquitetura de rede inicial nestes métodos está vinculada à determinação de critérios de parada do processo construtivo e a busca pela dimensão ótima é feita sempre a partir de arquiteturas com dimensão reduzida. Esta característica permite que as etapas *forward* e *backward* inerentes ao processo de treinamento demandem menos esforço computacional. Além disso, as redes neurais obtidas pelo método construtivo podem ser descritas utilizando um conjunto menor de parâmetros, apresentando a vantagem adicional de tornar mais evidente o papel individual de cada neurônio (DE CASTRO *et al.*, 1999).

Dentre os vários algoritmos construtivos reportados na literatura, o método utilizado nesta tese é baseado no *Aprendizado por Busca de Projeção*¹ (VON ZUBEN e NETTO, 1995; VON ZUBEN, 1996; VON ZUBEN e NETTO, 1997) e será apresentado com mais detalhes nas seções subseqüentes.

¹*Projection Pursuit Learning.*

2.2.2 Aprendizado por Busca de Projeção (*Projection Pursuit Learning*)

A complexidade da tarefa de modelagem aumenta significativamente com a dimensão do espaço de entrada e uma das maneiras utilizadas para reduzir este problema é realizando uma transformação no espaço de entradas. Algumas técnicas de modelagem empírica transformam o espaço de entradas para reduzir a sua dimensão tirando proveito de possíveis correlações entre os dados. Um dos métodos mais empregados com esta finalidade são os métodos baseados em projeção, que podem envolver a combinação linear ou não linear das entradas. Todos os métodos baseados em projeção, projetam as entradas em um hiperplano para explorar adequadamente as correlações entre elas ou para aproximar melhor as saídas e assim reduzir a dimensionalidade e o erro de aproximação do modelo empírico (UTOJO e BAKSHI, 1995).

O método baseado na técnica de regressão por busca de projeção (*Projection Pursuit Regression* - PPR), proposto por FRIEDMAN e STUETZLE (1981), é uma técnica de regressão não paramétrica que utiliza a composição aditiva de funções de expansão ortogonal a uma direção adequadamente fixada, construindo uma aproximação da função desejada através de um somatório de funções suaves de dimensão reduzida (*ridge functions*), que dependem das projeções dos dados. Este método faz parte da classe de métodos baseados em projeções lineares e, dessa forma, exploram as relações lineares entre as entradas projetando-as em um hiperplano antes da aplicação das funções básicas (CARREIRA-PERPIÑÁN, 1996).

Considerando, por simplicidade e sem perda de generalidade, uma função com apenas uma saída, os modelos de regressão por busca de projeção podem ser escritos da seguinte forma (LIMA, 2000):

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^n c_j f_j(\mathbf{v}_j^T \mathbf{x}) \quad (2.4)$$

onde $\mathbf{x} \in \Re^m$ é o vetor de variáveis de entrada, $\mathbf{v}_j \in \Re^m$ são os coeficientes da combinação linear ou a direção de projeção (pesos de entrada), f_j são as funções básicas (ou funções de ativação, no caso de redes neurais), c_j são os coeficientes de regressão, \hat{y} é a

2.2 Redes Neurais Artificiais Construtivas

saída aproximada e n é o número de coeficientes de regressão e de funções básicas.

O Aprendizado por Busca de Projeção (*Projection Pursuit Learning* - PPL) é um método inspirado em técnicas estatísticas baseadas nos modelos de regressão por busca de projeção (PPR) e pode ser visto como um método analítico para analisar dados.

O conceito de projeção é ilustrado na figura (2.1). Considere o exemplo em que, dado um conjunto de dados tridimensional, o objetivo é classificar os dados com características semelhantes. A projeção dos dados sobre o plano definido por e_2 e e_3 não é muito informativa pois os dados continuam confusos, sem uma estrutura característica aparente. Contudo, a projeção dos dados sobre o plano definido por e_1 e e_2 mostra claramente a presença de dois *clusters*.

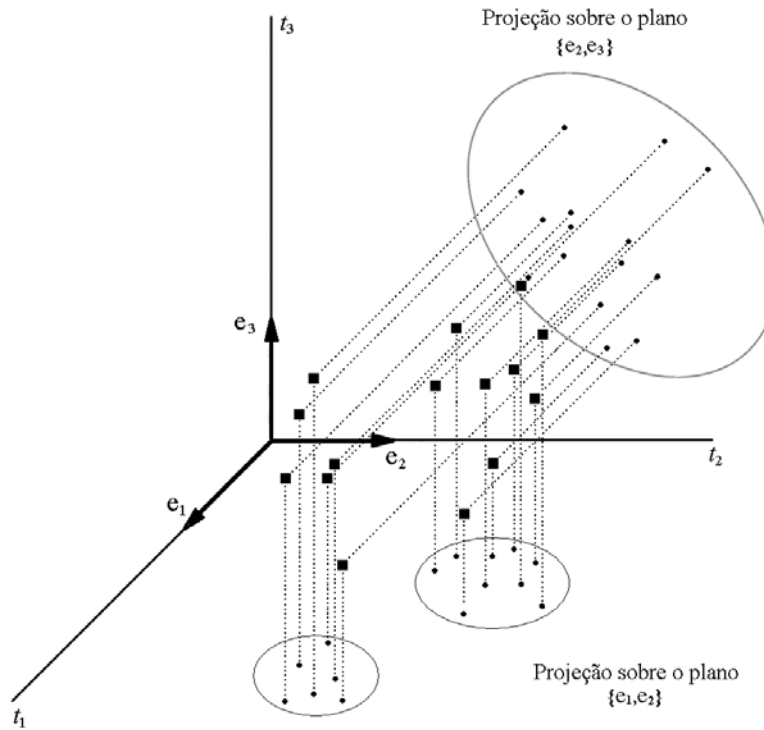


Figura 2.1: Projeções bidimensionais de um conjunto de dados tridimensional.

Este exemplo bastante simples serve para demonstrar que diferentes projeções do mesmo conjunto de dados podem revelar diferentes aspectos da estrutura de dados, embora algumas projeções possam falhar em revelar qualquer estrutura (CARREIRA-PERPIÑÁN, 1996). Na verdade, existe um número infinito de projeções que podem ser

obtidas e o Método da Busca de Projeções (*Projection Pursuit Learning* - *PPL*) foi desenvolvido para resolver este tipo de problema.

O ponto central do método PPL é encontrar “visões interessantes” de dimensão reduzida (geralmente 1, 2 ou 3) de um conjunto de dados multi-variável. O algoritmo PPL associa o valor de uma função, chamado de índice de projeção, a toda projeção de dimensão reduzida obtida a partir do conjunto de dados original (de dimensão superior). O índice de projeção assume valores maiores para projeções que revelam uma estrutura interessante, ou seja, aquelas que revelam parcelas representativas da estrutura original.

FRIEDMAN e STUETZLE (1981) desenvolveram um modelo de aproximação utilizando funções aditivas tomando como base o conjunto de dados (pontos no espaço de aproximação). Este modelo, que representa um caso particular da forma genérica dada pela equação (2.4), é reescrito aqui por conveniência:

$$\hat{g}_n(\mathbf{x}) = \sum_{j=1}^n f_j(\mathbf{v}_j^T \mathbf{x}) \quad (2.5)$$

Modelos deste tipo utilizam uma composição aditiva, cujos termos correspondem a funções escalares de expansão ortogonal a uma direção adequadamente fixada. O produto escalar $\mathbf{v}_j^T \mathbf{x}$ pode ser interpretado, a menos de uma constante, como uma projeção de \mathbf{x} na direção \mathbf{v}_j^T , sendo este o motivo pelo qual estes modelos de aproximação também recebem a denominação de modelos de aproximação por busca de projeção. A projeção consiste de operações lineares em que um mapeamento de uma determinada dimensão tem suprimidas algumas de suas estruturas de modo a tornar possível a sua representação em espaços de menor dimensão, sendo que qualquer mapeamento visto através de uma projeção é uma “sombra” do mapeamento original.

As combinações lineares podem ser vistas como projeções uni-direcionais e, portanto, projeções de \mathbb{R}^m em \mathbb{R}^p (com $p < m$) são mapeamentos lineares. Assim, pode-se escrever estas projeções da seguinte forma:

$$z_j = \mathbf{v}_j^T \mathbf{x} \quad (2.6)$$

onde \mathbf{v}_j é o vetor que define a transformação linear (ou direção de projeção) e z_j

2.2 Redes Neurais Artificiais Construtivas

representa os pontos (dados projetados) sobre o hiperplano de projeção de dimensão p . A determinação de uma direção de projeção está obviamente ligada à determinação de \mathbf{v}_j (CARREIRA-PERPIÑÁN, 1996).

Uma etapa importante deste método é justamente a determinação do critério de busca por projeções. A busca destas direções de projeção envolve uma série de manipulações do conjunto de dados.

O problema que se segue é definir que tipo de característica se deseja detectar no conjunto de dados original para depois obter uma medida do grau em que esta característica é exibida nos dados projetados, $\{z_j\}$. Esta medida é normalmente chamada de *Índice de Projeção*, $I(\{z\})$. O índice de projeção pode ser escrito na forma $I(\{\mathbf{v}_j^T \mathbf{x}\})$, evidenciando que o problema da busca de projeção pode ser encarado como um problema de otimização no qual \mathbf{v}_j deve ser escolhido de modo a produzir um máximo local de $I(\{z\})$. O índice de projeção é uma função real definida em \Re^m (CARREIRA-PERPIÑÁN, 1996):

$$I\{z : z \in \Re^m\} \longrightarrow I(\{z\}) \in \Re \quad (2.7)$$

Para evitar que o problema de maximização seja independente do tamanho dos vetores de projeção e também para obter direções não correlacionadas, o método PPL utiliza direções mutuamente ortogonais e de tamanho unitário. Assim o problema de otimização deve ser escrito considerando as seguintes restrições:

$$\max I(\{\mathbf{v}_j^T \mathbf{x}\}) \text{ sujeito a } \mathbf{v}_j^T \mathbf{v}_j = 1 \quad (2.8)$$

Uma vez que a determinação das direções de projeção depende da maximização do índice de projeção, várias propostas para a formulação deste índice numérico de projeção foram apresentadas na literatura, cada uma evidenciando um conjunto particular de características a serem atendidas pelos dados projetados. VON ZUBEN (1996) apresenta uma discussão detalhada sobre a motivação para a definição de um índice de projeção adequado e sobre a dificuldade de se obter um único índice de projeção que atenda a todas as expectativas. O autor faz ainda um estudo comparativo entre vários

dos índices propostos, indicando a solução que conduz ao melhor resultado. Embora não exista um índice de projeção que se aplique em todos os casos, a utilização de um índice de desempenho para determinar a direção de projeção aumenta a probabilidade de se encontrarem direções de projeção *interessantes* quando comparado com a determinação feita de forma arbitrária.

Após a direção de projeção ter sido determinada, ajusta-se a forma da função mono-variável de modo que a sua expansão na direção ortogonal à direção de projeção forneça a melhor aproximação possível dos dados originais. Finalmente, a busca seqüencial de direções de projeção é implementada repetindo o seguinte procedimento até que o modelo de aproximação concorde com os dados originais em todas as direções: determinadas a direção de projeção ótima e a função de expansão ortogonal correspondente, remove-se do conjunto de dados original a estrutura resultante da projeção dos dados nesta direção. O processo é reiniciado até que nenhuma outra projeção revele qualquer estrutura.

Desse modo, através de um procedimento iterativo e construtivo, cada novo procedimento de busca de direção tem que aproximar apenas as informações que não foram representadas pelos procedimentos anteriores. Obtêm-se, desta forma, funções de aproximação multi-variáveis formadas pela composição aditiva de funções mono-variáveis com a forma geral dada pela equação (2.5) (VON ZUBEN, 1996).

2.2.2.1 Redes Neurais Artificiais via Métodos Construtivos

Uma rede neural pode ser vista como um modelo *paramétrico* se a sua arquitetura (estrutura de conexão e número de neurônios da camada intermediária) for definida *a priori*. Se, por outro lado, a sua arquitetura puder ser definida em função do problema de aproximação, a rede neural corresponde a um modelo *não paramétrico*².

As redes neurais não paramétricas utilizadas nesta tese para gerar modelos dinâmicos de processos químicos (posteriormente utilizados no projeto de controladores

²A terminologia *não paramétrico* não significa ausência de parâmetros; ela é empregada para designar um modelo neural cuja arquitetura é definida automaticamente em função do problema de aproximação, ou seja, quando a estrutura das conexões e o número de neurônios escondidos não são definidos *a priori*.

2.2 Redes Neurais Artificiais Construtivas

preditivos) foram propostas por VON ZUBEN (1996) e configuram um avanço em relação aos modelos neurais construtivos propostos inicialmente por BÄRMANN e BIEGLER-KÖNIG (1992). Esta nova forma de representação do modelo de projeção (VON ZUBEN e NETTO, 1995; VON ZUBEN, 1996; VON ZUBEN e NETTO, 1997) otimiza a condição de resolubilidade³ do problema de aproximação, transformando o problema de obtenção de modelos de projeção em um problema de aprendizado construtivo (seção 2.2.3) para redes neurais com uma camada intermediária.

Os modelos de projeção descritos pela equação (2.5) podem ser generalizados para representar funções com um número de saídas $r > 1$. A forma mais geral deste tipo de modelo, a aproximação multi-variável, considera possíveis associações entre as variáveis de saída e pode ser representado da seguinte forma (VON ZUBEN, 1996):

$$\hat{g}_{n,k}(\mathbf{x}) = \sum_{j=1}^n w_{kj} f_j(\mathbf{v}_j^T \mathbf{x}), \quad k = 1, \dots, r \quad (2.9)$$

O modelo de aproximação fornecido por uma rede neural artificial com uma camada intermediária (figura 2.2) para aproximar o conjunto de dados $\{(\mathbf{x}_l, \mathbf{s}_l)\}_{l=1}^N$, gerado da forma $\mathbf{s}_l = g(\mathbf{x}_l) + \boldsymbol{\epsilon}_l$, sendo $\mathbf{x}_l \in X \subset \mathbb{R}^m$ e $\mathbf{s}_l, \boldsymbol{\epsilon}_l \in \mathbb{R}^r$, pode ser escrito do seguinte modo:

$$\hat{s}_{kl} = RN_k^{[n]}(\mathbf{V}, \mathbf{w}_k, \mathbf{x}_l) = \sum_{j=1}^n w_{kj} f_j(\mathbf{v}_j^T \mathbf{x}_l + v_{j0}) + w_{k0}, \quad \text{com} \quad \begin{cases} k = 1, \dots, r \\ l = 1, \dots, N \end{cases} \quad (2.10)$$

onde

³Significa, de modo simplificado, que uma rede neural com uma camada intermediária com neurônios não lineares e uma camada de saída com neurônios lineares é teoricamente capaz de aprender exatamente um dado mapeamento se o número de neurônios escondidos for igual ao número de exemplos a ser aprendido (VON ZUBEN e NETTO, 1995). Uma discussão mais aprofundada sobre o assunto pode ser encontrada em BÄRMANN e BIEGLER-KÖNIG (1992).

$$\mathbf{V} = \begin{bmatrix} v_{10} & v_{11} & \cdots & v_{1m} \\ v_{20} & \ddots & & \vdots \\ \vdots & & & \vdots \\ v_{n0} & \cdots & & v_{nm} \end{bmatrix} \text{ e } \mathbf{W} = \begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1n} \\ w_{20} & \ddots & & \vdots \\ \vdots & & & \vdots \\ w_{r0} & \cdots & & w_{rn} \end{bmatrix}$$

sendo que $\mathbf{w}_k \in \mathbb{R}^n$ é o vetor coluna formado pelos elementos da k -ésima linha da matriz de pesos sinápticos \mathbf{W} e $\mathbf{v}_j \in \mathbb{R}^m$ é o vetor coluna formado pelos elementos da j -ésima linha da matriz de pesos \mathbf{V} , com exceção dos elementos da primeira coluna.

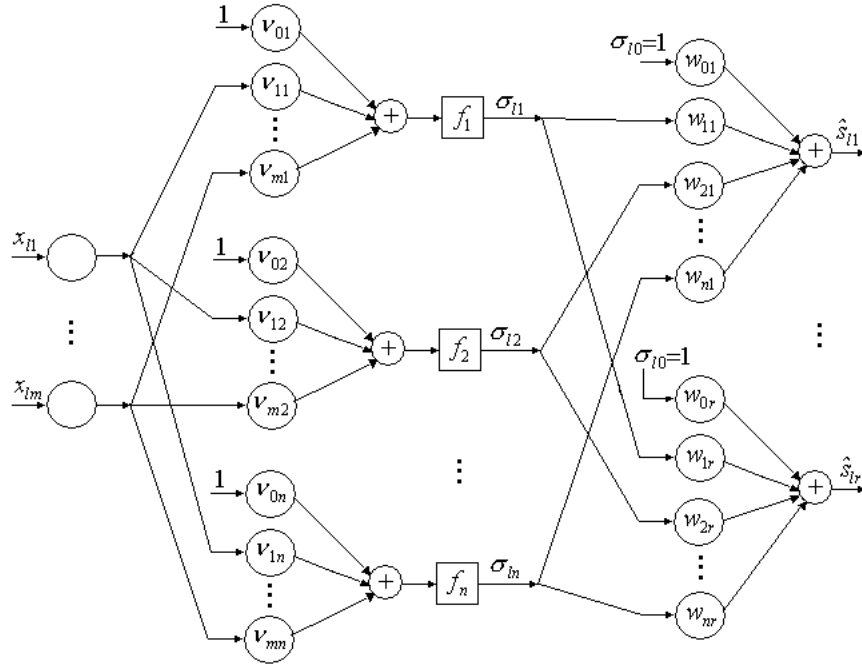


Figura 2.2: Rede neural resultante do processo de aprendizado construtivo.

A equação (2.10) também pode ser escrita na forma matricial:

$$\Sigma \mathbf{W}^T = \hat{\mathbf{S}}^T \quad (2.11)$$

onde

2.2 Redes Neurais Artificiais Construtivas

$$\Sigma = \begin{bmatrix} 1 & \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ 1 & \sigma_{21} & \ddots & & \vdots \\ \vdots & \vdots & & & \vdots \\ 1 & \sigma_{N1} & \cdots & & \sigma_{Nn} \end{bmatrix} \text{ e } \hat{\Sigma} = \begin{bmatrix} \hat{s}_{11} & \hat{s}_{12} & \cdots & \hat{s}_{1N} \\ \hat{s}_{21} & \ddots & & \vdots \\ \vdots & & & \vdots \\ \hat{s}_{r1} & \cdots & & \hat{s}_{rN} \end{bmatrix}$$

com $\sigma_{lj} = f_j(\mathbf{v}_j^T \mathbf{x}_l + v_{j0})$.

Comparando os modelos fornecidos pelas equações (2.9) e (2.10), nota-se que a única diferença são os termos referentes às polarizações dos neurônios presentes apenas na equação (2.10). Dessa forma, verifica-se que uma rede neural não polarizada com uma camada interna poderia ser implementada através de métodos estatísticos para gerar modelos na forma da equação (2.9). A polarização, característica dos modelos neurais, pode então ser adicionada para produzir modelos de aproximação, \hat{g} , formados por uma composição aditiva de funções f_j ($j = 1, \dots, n$) adequadamente transladadas por v_{j0} , rotacionadas por \mathbf{v}_j e escalonadas por $\|\mathbf{v}_j\|$. Para cada saída k , as avaliações de f_j em projeções de \mathbf{x} na direção \mathbf{v}_j são transladadas por w_{k0} e escalonadas por w_{kj} (VON ZUBEN, 1996).

Uma característica marcante deste algoritmo é que, ao invés de utilizarem funções de ativação fixas do tipo sigmoidal e efetuar o treinamento da rede com todos os neurônios ao mesmo tempo, o método de aprendizado construtivo utiliza funções de ativação mais flexíveis e treinam um neurônio de cada vez. Utiliza-se o procedimento do retro-ajuste (*back fitting*) para ajustar ciclicamente as conexões de cada neurônio introduzido, mantendo fixos os parâmetros das outras unidades já instaladas, até que não haja variações no desempenho do modelo (DE CASTRO *et al.*, 1999).

A função de ativação escolhida para as redes construtivas utilizadas nesta tese foram os polinômios de Hermite⁴. Assim, cada neurônio da camada intermediária pode ser representado como sendo uma combinação linear de funções ortonormais dada por:

$$f(z) = \sum_{r=1}^P c_r h_r(z) \quad (2.12)$$

⁴Os motivos para a escolha deste tipo polinômio são apresentados na seção 2.2.3.2.

onde $h_r(z)$ é um polinômio de Hermite e P é a ordem da função de ativação (parâmetro definido pelo usuário) que controla o grau de flexibilidade da função de ativação.

Na seção 2.2.3 são descritas todas as etapas do procedimento iterativo para se obter um modelo neural com uma camada intermediária, m entradas e r saídas (figura 2.2), cujos número de neurônios e forma da função de ativação são determinados automaticamente utilizando o algoritmo de aprendizado construtivo por busca de projeção.

2.2.3 O Aprendizado Construtivo (*Unit-Growing Learning*) no Contexto de Redes Neurais Artificiais

Embora os problemas relacionados com a estimação adequada do conjunto de pesos das redes neurais convencionais tenham sido resolvidos com relativo sucesso através da aplicação de métodos avançados de otimização não linear de 1ª e 2ª ordem (BATTITI, 1992; VAN DER SMAGT, 1994), outros dois obstáculos, i.e., a definição da dimensão da rede neural e a escolha das funções de ativação, ainda são resolvidos heurísticamente para a maioria das aplicações através de exaustivos procedimentos de tentativa e erro.

Avanços bastante significativos no desenvolvimento de algoritmos de treinamento supervisionado para RNAs foram proporcionados por BÄRMANN e BIEGLER-KÖNIG (1992) através de um tipo de método de aprendizado construtivo onde neurônios extras podem ser adicionados à camada intermediária da rede neural durante o processo de treinamento até que a rede seja capaz de aproximar a função com a precisão desejada. Este método permite que a dimensão ótima de uma RNA com uma camada intermediária seja determinada automaticamente, dado o conjunto (não necessariamente ótimo) de funções de ativação. Apesar de determinar automaticamente quantos neurônios escondidos devem ser utilizados para aproximar uma função com uma dada precisão, este método apresenta uma séria restrição, qual seja, todos os neurônios escondidos usam a mesma função de ativação.

2.2 Redes Neurais Artificiais Construtivas

Conforme reportado por VON ZUBEN e NETTO (1995), alguns autores (HWANG *et al.*, 1994; LAY *et al.*, 1994) aplicaram modelos paramétricos com funções de base ortonormal com sucesso na busca de funções de ativação ótimas. ROOSEN e HASTIE (1994), por sua vez, empregaram uma estratégia não paramétrica utilizando *splines* suavizantes (*automatic smoothing splines*) com o mesmo objetivo. A vantagem desta abordagem sobre a anterior é a sua flexibilidade e a ausência de parâmetros especificados pelo usuário.

VON ZUBEN e NETTO (1995; 1996; 1997) propuseram um algoritmo de *Aprendizado Construtivo* (“Unit-Growing Learning” - UGL) que se mostrou capaz de superar a restrição de utilizar a mesma função de ativação em todos os neurônios escondidos. Nesta abordagem, a técnica do *Aprendizado por Busca de Projeção* (“Projection Pursuit Learning” - PPL) é aplicada em associação com a otimização da condição de resolubilidade, de modo que cada função de ativação pode ser ótima e individualmente definida para cada problema de aproximação. O método determina não somente o número ótimo de neurônios escondidos, mas também a melhor função de ativação para cada neurônio em função do conjunto de treinamento. Uma vez que o processo de treinamento age individualmente nos neurônios escondidos, uma função de ativação pertinente pode ser desenvolvida iterativamente para cada neurônio em função do conjunto de treinamento. As redes neurais construtivas utilizadas nesta tese são descritas em detalhes em VON ZUBEN e NETTO (1996) e VON ZUBEN (1996) e empregam os polinômios de Hermite para gerar as funções de ativação não lineares dos neurônios escondidos.

A otimização da condição de resolubilidade propicia o projeto de redes neurais com dimensão reduzida, fato que constitui um passo importante no sentido de melhorar a capacidade de generalização dos modelos neurais.

2.2.3.1 O Método UGL

Considere o problema de regressão cujo objetivo é fornecer a melhor aproximação de uma função contínua multi-dimensional não modelada, $G(\cdot)$, definida do seguinte modo:

$$G(\cdot) : \Re^{1 \times m} \rightarrow \Re^{1 \times r} \quad (2.13)$$

Partindo de N pares de vetores entrada-saída derivados de um processo de amostragem com um erro aditivo tal que

$$\mathbf{s}_{(l)} = G(\mathbf{x}_{(l)}) + \boldsymbol{\epsilon}_{(l)} \quad (2.14)$$

obtem-se para $l = 1, \dots, N$:

$$(\mathbf{x}_{(l)}, \mathbf{s}_{(l)}) = ([x_{l1} \ x_{l2} \ \cdots \ x_{lm}] , [s_{l1} \ s_{l2} \ \cdots \ s_{lr}]) \quad (2.15)$$

O objetivo da tarefa de regressão é construir um estimador, $\hat{G}(\cdot)$, que forneça a melhor aproximação de $G(\cdot)$ de modo que tal estimador seja capaz de prever $\mathbf{s}_{(t)}$, dado $\mathbf{x}_{(t)}$ ($t \neq l$), da seguinte maneira:

$$\hat{\mathbf{s}}_{(t)} = \bar{\mathbf{s}} + \hat{G}(\mathbf{x}_{(t)}) \quad (2.16)$$

onde $\bar{\mathbf{s}} \in \Re^{1 \times r}$ é a média amostrada do conjunto de dados da saída desejada, dada por

$$\bar{\mathbf{s}} = [\bar{s}_1 \ \bar{s}_2 \ \cdots \ \bar{s}_r] = \frac{1}{N} \sum_{l=1}^N \mathbf{s}_{(l)} \quad (2.17)$$

As redes neurais artificiais com uma camada intermediária podem ser utilizadas para estimar a matriz de resposta da saída, $\mathbf{S} \in \Re^{N \times r}$, dada a matriz de variáveis independentes, $\mathbf{X} \in \Re^{N \times m}$:

$$\hat{\mathbf{s}}_{lk} = \bar{\mathbf{s}}_k + \sum_{j=1}^n \left[w_{jk} f_j \left(\sum_{i=1}^m v_{ij} x_{li} \right) \right], \text{ com } \begin{matrix} k = 1, \dots, r \\ l = 1, \dots, N \end{matrix} \quad (2.18)$$

2.2 Redes Neurais Artificiais Construtivas

onde v_{ij} ($\mathbf{V} \in \Re^{m \times n}$) é o peso da camada intermediária conectando o i -ésimo elemento da entrada ao j -ésimo neurônio escondido, w_{jk} ($\mathbf{W} \in \Re^{n \times r}$) representa o peso da camada de saída conectando o j -ésimo neurônio escondido ao k -ésimo neurônio da camada de saída e $f_j : \Re \rightarrow \Re$ é a função de ativação ajustável do j -ésimo neurônio escondido.

Introduzindo uma coluna adicional preenchida com números “1” como a primeira coluna de \mathbf{X} , pode-se definir $\mathbf{v}_j \in \Re^{(m+1)}$ como a j -ésima coluna de \mathbf{V} e $\mathbf{x}_l \in \Re^{(m+1)}$ como o vetor coluna correspondente à l -ésima coluna da matriz estendida \mathbf{X} , levando à seguinte expressão para os sinais de entrada para a camada de saída ($l = 1, \dots, N$):

$$\begin{cases} \sigma_{l0} &= 1 \\ \sigma_{lj} &= f_j \left(\sum_{i=1}^m v_{ij} x_{il} + v_{0j} \right) = f_j(\mathbf{v}_j^T \mathbf{x}_l) \end{cases} \quad (2.19)$$

onde os vetores \mathbf{v}_j ($j = 1, \dots, n$) representam as direções de projeção.

Usando as equações (2.18) e (2.19), a condição de resolubilidade implica que o seguinte sistema de equações lineares tem uma solução (BÄRMANN e BIEGLER-KÖNIG, 1992):

$$\mathbf{\Sigma} \cdot \mathbf{W} = \mathbf{S} \quad (2.20)$$

onde

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_{10} & \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{20} & \sigma_{21} & \ddots & & \vdots \\ \vdots & \vdots & & & \vdots \\ \sigma_{N0} & \sigma_{N1} & \cdots & \cdots & \sigma_{Nn} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} w_{01} & w_{02} & \cdots & w_{0r} \\ w_{11} & \ddots & & \vdots \\ \vdots & & & \vdots \\ w_{n1} & \cdots & & w_{nr} \end{bmatrix} \text{ e}$$

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1r} \\ s_{21} & \ddots & & \vdots \\ \vdots & & & \vdots \\ s_{N1} & \cdots & & s_{Nr} \end{bmatrix}.$$

De modo equivalente, esta condição requer que todo vetor coluna de \mathbf{S} , representados por \mathbf{s}_k ($k = 1, \dots, r$), pertença ao sub-espço coberto pelos vetores coluna de $\mathbf{\Sigma}$, representados por $\boldsymbol{\sigma}_j$ ($j = 1, \dots, n$). Deve-se notar que a matriz dos pesos \mathbf{W} não contribui para a resolubilidade e que a habilidade para representar uma matriz \mathbf{S} arbitrária aumenta com o número n de vetores coluna de $\mathbf{\Sigma}$ (número de neurônios da camada intermediária). Uma discussão detalhada sobre a otimização da condição de resolubilidade é feita por VON ZUBEN e NETTO (1995). O *Aprendizado Construtivo* (Unit-Growing Learning - UGL) é então apresentado como uma solução alternativa para este problema.

A grande vantagem do algoritmo de aprendizado construtivo (UGL) proposto por VON ZUBEN e NETTO (1997) quando comparado com a abordagem de BÄRMANN e BIEGLER-KÖNIG (1992) é que o primeiro utiliza neurônios escondidos com funções de ativação a serem definidas durante o processo de aprendizado, podendo assim serem não apenas distintas entre si, mas também tais que reflitam as não linearidades específicas presentes nos dados de treinamento. O algoritmo UGL realiza um procedimento iterativo que concilia a otimização da condição de resolubilidade e a técnica do *Aprendizado por Busca de Projeção* (Projection Pursuit Learning - PPL) para determinar uma função de ativação adequada, assim como os pesos das conexões correspondentes a cada neurônio, individualmente. Este procedimento resulta em um algoritmo de aprendizado computacional mais eficiente e acurado, capaz de gerar redes neurais de uma camada intermediária com um número reduzido de neurônios.

Enquanto os métodos do tipo *back-propagation* em batelada aplicam passos de otimização para estimar os pesos de todas as camadas simultaneamente, o método de treinamento construtivo opera neurônio a neurônio, iniciando com uma rede neural com um único neurônio escondido e, dinamicamente, vai adicionado novos neurônios até que a tarefa de aproximação tenha sido concluída.

Representando a j -ésima linha de \mathbf{W} por $\mathbf{w}_{(j)}$ ($j = 0, 1, \dots, n$), a equação (2.18) pode ser expressa como:

$$\hat{\mathbf{S}} = \mathbf{\Sigma} \cdot \mathbf{W} = \sum_{j=0}^n \boldsymbol{\sigma}_j \mathbf{w}_{(j)} \quad (2.21)$$

2.2 Redes Neurais Artificiais Construtivas

com σ_0 sendo um vetor preenchido com números “1” e σ_j ($j = 1, \dots, n$) vetores colunas cujos elementos são definidos de acordo com a equação (2.19).

Para $n = 0$, o seguinte problema de mínimos quadrados linear é resolvido (onde $\|\cdot\|$ é a norma L_2):

$$\min_{\mathbf{w}_{(0)}} \left\| \sigma_0 \mathbf{w}_{(0)} - \mathbf{S} \right\| = \min_{\mathbf{w}_{(0)}} \left\| \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} [\mathbf{w}_{01} \ \mathbf{w}_{02} \ \cdots \ \mathbf{w}_{0r}] - \mathbf{S} \right\| \quad (2.22)$$

Para um dado $n > 0$, seja \mathbf{D}_n a matriz que contém a informação ainda não representada por uma rede neural com uma camada intermediária contendo $n - 1$ neurônios, isto é:

$$\mathbf{D}_n = \mathbf{S} - \sum_{j=0}^{n-1} \sigma_j \mathbf{w}_{(j)} \quad (2.23)$$

então, deve-se resolver o seguinte problema de otimização:

$$\min_{\sigma_n, \mathbf{w}_{(n)}} \left\| \sigma_n \mathbf{w}_{(n)} - \mathbf{D}_n \right\| \quad (2.24)$$

cuja solução força σ_n a ficar otimamente alinhado com as colunas de \mathbf{D}_n .

A direção que fornece o alinhamento ótimo (σ_n^*) no sentido dos mínimos quadrados, é fornecida pelo componente principal da matriz $\mathbf{D}_n \mathbf{D}_n^T$. Admitindo que $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$ são os autovalores de $\mathbf{D}_n \mathbf{D}_n^T$, com os autovetores correspondentes dados por $\mathbf{u}_1^n, \mathbf{u}_2^n, \dots, \mathbf{u}_N^n$ ($\mathbf{u}_l^n \in \Re^{N \times 1}; l = 1, \dots, N$), então o componente principal de $\mathbf{D}_n \mathbf{D}_n^T$ é \mathbf{u}_1^n (ou $-\mathbf{u}_1^n$).

Denotando o componente principal genericamente por \mathbf{u}_n , então σ_n é ótimo para algum $\mathbf{v}_n \in \Re^{(m \times 1)}$ e $f_n : \Re \rightarrow \Re$ que maximize o produto escalar normalizado (BÄRMANN e BIEGLER-KÖNIG, 1992):

$$\boldsymbol{\sigma}_n^* = \arg \max_{\boldsymbol{\sigma}_n} \frac{(\boldsymbol{\sigma}_n^T \mathbf{u}_n)^2}{\boldsymbol{\sigma}_n^T \boldsymbol{\sigma}_n} \quad (2.25)$$

A solução para este problema de otimização é obtido através do seguinte procedimento iterativo envolvendo f_n e \mathbf{v}_n (VON ZUBEN e NETTO, 1997):

1. Iniciando com um valor fixo para \mathbf{v}_n , encontrar uma f_n ótima (seção 2.2.3.2);
2. Com f_n fixa, determinar um \mathbf{v}_n ótimo (seção 2.2.3.3), e;
3. Com este novo \mathbf{v}_n fixo, encontrar uma nova f_n ótimo e assim sucessivamente.

O procedimento iterativo deve continuar até a convergência, que tem a garantia de existir (VON ZUBEN e NETTO, 1997).

Deve-se notar que, diferente das versões anteriores do aprendizado por busca de projeção (HWANG *et al.*, 1994; ROOSEN e HASTIE, 1994), a abordagem proposta por VON ZUBEN e NETTO (1995; 1997) não utiliza qualquer valor estimado de $\mathbf{w}_{(n)}$ para obter $\boldsymbol{\sigma}_n^*$.

Uma vez obtido $\boldsymbol{\sigma}_n^*$ da equação (2.25), ou seja, f_n^* e \mathbf{v}_n^* (valores ótimos), determina-se um valor para $\mathbf{w}_{(n)}$ que resolva o problema de mínimos quadrados dado pela equação (2.24), com $\boldsymbol{\sigma}_n = \boldsymbol{\sigma}_n^*$ fixo. Este valor ótimo assume a seguinte forma:

$$\mathbf{w}_{(n)}^* = (\boldsymbol{\sigma}_n^{*T} \boldsymbol{\sigma}_n^*)^{-1} \boldsymbol{\sigma}_n^{*T} \mathbf{D}_n \quad (2.26)$$

onde $\boldsymbol{\sigma}_n^{*T} \boldsymbol{\sigma}_n^*$ é um escalar.

Neurônios adicionais são introduzidos na camada intermediária, mantendo inalterados os parâmetros associados aos neurônios escondidos já existentes, enquanto a condição $\|\mathbf{D}_{n+1}\| > \delta$ for verificada, sendo δ um limite previamente especificado.

Outro ponto a ser destacado é que o procedimento iterativo para determinar o número ótimo de neurônios escondidos, n , proposto pelas abordagens de HWANG *et al.* (1994) e ROOSEN e HASTIE (1994) envolvem três grupos de variáveis $[\mathbf{v}_n, f_n, \mathbf{w}_{(n)}]$,

2.2 Redes Neurais Artificiais Construtivas

enquanto que na abordagem de VON ZUBEN e NETTO (1995; 1997) o procedimento iterativo é reduzido para dois grupos de variáveis, $[\mathbf{v}_n, f_n]$, seguido pelo cálculo do terceiro grupo $[\mathbf{w}_{(n)}]$ de uma forma fechada e ótima, com ganho garantido no desempenho computacional e na acurácia.

2.2.3.2 Determinação da Função de Ativação

Descreve-se a seguir o procedimento para a determinação de uma função de ativação ótima, f_j , que resolva a equação (2.25) para um dado \mathbf{v}_n fixo arbitrariamente.

No problema de regressão com múltiplas saídas, cada variável de resposta é estimada como uma combinação linear das ativações f_j ($j = 1, \dots, n$), conforme descrito na equação (2.18). Para cada j , uma função de ativação não linear suave ótima f_j é determinada a partir de uma combinação linear de funções ortonormais, mantendo constantes os vetores de peso $\mathbf{v}_j \in \Re^{(m \times 1)}$.

Para cada neurônio j ($j = 1, \dots, n$), dado $\mathbf{u}_1^j \in \Re^{(N \times 1)}$, que é o componente principal de $\mathbf{D}_j \mathbf{D}_j^T$ (ver página 25), e com $\mathbf{v}_j \in \Re^{(m \times 1)}$ fixo, são obtidas N amostras entrada-saída do mapeamento não linear a ser realizado pela função de ativação f_j :

$$(z_l, y_l) = (\mathbf{x}_{(l)} \mathbf{v}_j, \mathbf{u}_{1l}^j), \quad l = 1, \dots, N \quad (2.27)$$

Para definir uma curva suave, \hat{f}_j , que realize a melhor aproximação destes N pontos, utilizam-se modelos paramétricos uni-dimensionais baseados em combinações lineares de funções ortonormais. Os polinômios de Hermite foram as funções ortonormais utilizadas para implementar este algoritmo UGL pelo fato de fornecerem uma aproximação suave e o cálculo rápido e preciso dos valores funcionais e de suas derivadas.

Para um dado conjunto $\{(z_l, y_l)\}_{l=1}^N$, a função de suavização pode ser obtida por uma combinação linear de funções ortonormais de Hermite de ordem P :

$$f(z) = \sum_{i=0}^P c_i h_i(z) \quad (2.28)$$

onde c_i são os coeficientes de regressão e $h_i(z)$ são as funções ortonormais de Hermite.

Os polinômios de Hermite, $H_i(z)$, podem ser obtidos recursivamente da seguinte forma:

$$\begin{aligned} H_0(z) &= 1, \\ H_1(z) &= 2z, \\ H_{p+1}(z) &= 2(zH_p(z) - pH_{p-1}(z)), \quad p \geq 1 \end{aligned} \quad (2.29)$$

Os polinômios de Hermite são definidos como ortogonais quando estão associados com a função peso $\phi(z) = \pi^{-(1/4)}e^{-(z^2/2)}$, no intervalo $(-\infty, \infty)$:

$$\int_{-\infty}^{\infty} \phi^2(z) H_{p1}(z) H_{p2}(z) dz = \begin{cases} 0, & \text{se } p1 \neq p2 \\ p! 2^p, & \text{se } p1 = p2 = p \end{cases} \quad (2.30)$$

As funções ortonormais de Hermite podem então ser definidas como:

$$h_p(z) = \frac{1}{\sqrt{p! 2^p}} H_p(z) \phi(z), \quad -\infty < z < \infty \quad (2.31)$$

Utilizando funções de Hermite de ordem P , as saídas y_l ($l = 1, \dots, N$) podem ser estimadas da seguinte forma:

$$\hat{y}_l = f_j(z_l) = \sum_{p=1}^P c_p h_p(z_l) \quad (2.32)$$

Definindo:

$$\mathbf{H} = \begin{bmatrix} h_1(z_1) & h_2(z_1) & \cdots & h_P(z_1) \\ h_1(z_2) & \ddots & & h_P(z_2) \\ \vdots & & \ddots & \vdots \\ h_1(z_N) & \cdots & \cdots & h_P(z_N) \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_P \end{bmatrix} \quad \text{e} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad (2.33)$$

2.2 Redes Neurais Artificiais Construtivas

o valor ótimo de \mathbf{c} é obtido através do seguinte problema de otimização:

$$\arg \min_{\mathbf{c}} \left\| \mathbf{y} - \mathbf{H}\mathbf{c} \right\| \quad (2.34)$$

cuja solução ótima, \mathbf{c}^* , no sentido dos mínimos quadrados (assumindo que $P \leq N$) é dada por:

$$\mathbf{c}^* = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y} \quad (2.35)$$

Deve-se notar que colunas de \mathbf{H} podem ser sempre eliminadas, para cada j , de modo que a matriz $\mathbf{H}^T \mathbf{H}$ será sempre inversível (VON ZUBEN e NETTO, 1997).

As principais vantagens apresentadas pelas funções de Hermite, quando utilizadas como funções de ativação dos neurônios escondidos das redes neurais construtivas, devem-se à sua capacidade de fornecer uma aproximação suave para um dado mapeamento e às suas propriedades especiais (funções ortonormais) de poderem ser computadas analiticamente e recursivamente, não requerendo a armazenagem da função suavizada nem do intervalo de interpolação. O maior problema apresentado pelos polinômios de Hermite é a necessidade da determinação *a priori* da ordem P dos polinômios, principalmente porque este parâmetro pode ser crítico para o sucesso da aproximação. Valores de P elevados podem, a princípio, diminuir o erro de aproximação, contudo, existe a possibilidade de que o erro de generalização aumente (LIMA, 2000).

2.2.3.3 Determinação das Projeções

Considera-se a seguir o problema da determinação de uma direção de projeção ótima, \mathbf{v}_n^* , que resolva o problema da equação (2.25) para uma f_n arbitrariamente fixa. O primeiro passo é reescrever a equação (2.25) em uma forma equivalente, utilizando a equação (2.19) e considerando o fato de que f_n agora é uma função fixa:

$$\mathbf{v}_n^* = \arg \max_{\mathbf{v}_n} \frac{(\boldsymbol{\sigma}_n^T \mathbf{u}_n)^2}{\boldsymbol{\sigma}_n^T \boldsymbol{\sigma}_n} \quad (2.36)$$

O cálculo dos valores funcionais dos polinômios de Hermite e de suas derivadas pode ser realizado com pouco esforço computacional, o que facilita a obtenção das informações das derivadas de primeira e segunda ordem a partir da função objetivo definida pela equação (2.36). A qualidade da solução \mathbf{v}_n^* geralmente depende do valor inicial atribuído a \mathbf{v}_n (VON ZUBEN, 1996).

Baseado nas propriedades especiais dos polinômios de Hermite:

$$\frac{dH_p(z)}{dz} = 2p H_{p-1}(z) \quad (2.37)$$

as derivadas de h_p e de f em relação a z podem ser calculadas analiticamente (VON ZUBEN e NETTO, 1995):

$$\frac{dh_p(z)}{dz} = \sqrt{2p} h_{p-1}(z) - z h_p(z) \quad (2.38)$$

$$\frac{d^2 h_p(z)}{dz^2} = 2\sqrt{p(p-1)} h_{p-2}(z) - 2\sqrt{2p} z h_{p-1}(z) + (z^2 - 1)h_p(z) \quad (2.39)$$

$$f'(z) = \frac{df_j(z)}{dz} = \sum_{p=1}^P c_p \frac{dh_p(z)}{dz} \quad (2.40)$$

$$f''(z) = \frac{d^2 f_j(z)}{dz^2} = \sum_{p=1}^P c_p \frac{d^2 h_p(z)}{dz^2} \quad (2.41)$$

permitindo o desenvolvimento de algoritmos de otimização não lineares que utilizam derivadas para ajustar os pesos da camada intermediária $\mathbf{v}_j \in \Re^{m \times 1}$.

Seja $\mathbf{f}_j : \Re^{N \times m} \times \Re^{m \times 1} \rightarrow \Re^{N \times 1}$ definido da seguinte forma:

$$\mathbf{f}_j(\mathbf{x}, \mathbf{v}_j) = \boldsymbol{\sigma}_j = [f_j(\mathbf{x}_{(1)} \mathbf{v}_j) f_j(\mathbf{x}_{(2)} \mathbf{v}_j) \cdots f_j(\mathbf{x}_{(N)} \mathbf{v}_j)]^T \quad (2.42)$$

2.2 Redes Neurais Artificiais Construtivas

Para cada j ($j = 1, \dots, n$), dada uma função de ativação fixa f_j , o problema de otimização descrito na equação (2.25) fornece:

$$\boldsymbol{\sigma}_j^* = \arg \max_{\boldsymbol{\sigma}_j} \frac{(\boldsymbol{\sigma}_j^T \mathbf{u}_j)^2}{\boldsymbol{\sigma}_j^T \boldsymbol{\sigma}_j} \Rightarrow \mathbf{v}_j^* = \arg \max_{\mathbf{v}_j} \frac{(\mathbf{f}_j(\mathbf{x}, \mathbf{v}_j)^T \cdot \mathbf{u}_j)^2}{\mathbf{f}_j(\mathbf{x}, \mathbf{v}_j)^T \cdot \mathbf{f}_j(\mathbf{x}, \mathbf{v}_j)} \quad (2.43)$$

A função objetivo pode ser expressa da seguinte forma (VON ZUBEN e NETTO, 1995):

$$\frac{(\mathbf{f}_j(\mathbf{x}, \mathbf{v}_j)^T \cdot \mathbf{u}_j)^2}{\mathbf{f}_j(\mathbf{x}, \mathbf{v}_j)^T \cdot \mathbf{f}_j(\mathbf{x}, \mathbf{v}_j)} = \frac{q_j(\mathbf{v}_j)^2}{h_j(\mathbf{v}_j)} = F_j(\mathbf{v}_j) \quad (2.44)$$

O gradiente de $F_j(\cdot)$ em relação ao vetor de parâmetros ajustáveis $\mathbf{v}_j \in \Re^{m \times 1}$ fornece:

$$\frac{\partial F_j(\mathbf{v}_j)}{\partial \mathbf{v}_j} = 2 \frac{q_j(\mathbf{v}_j)}{h_j(\mathbf{v}_j)} \frac{\partial q_j(\mathbf{v}_j)}{\partial \mathbf{v}_j} - \frac{q_j(\mathbf{v}_j)^2}{h_j(\mathbf{v}_j)^2} \frac{\partial h_j(\mathbf{v}_j)}{\partial \mathbf{v}_j} \quad (2.45)$$

onde

$$\frac{\partial q_j(\mathbf{v}_j)}{\partial \mathbf{v}_j} = \left(\frac{\partial \mathbf{f}_j(\mathbf{x}, \mathbf{v}_j)}{\partial \mathbf{v}_j} \right)^T \cdot \mathbf{u}_j \quad (2.46)$$

$$\frac{\partial h_j(\mathbf{v}_j)}{\partial \mathbf{v}_j} = 2 \left(\frac{\partial \mathbf{f}_j(\mathbf{x}, \mathbf{v}_j)}{\partial \mathbf{v}_j} \right)^T \cdot \mathbf{f}_j(\mathbf{x}, \mathbf{v}_j) \quad (2.47)$$

e

$$\frac{\partial \mathbf{f}_j(\mathbf{x}, \mathbf{v}_j)}{\partial \mathbf{v}_j} = \begin{bmatrix} x_{11} f_j'(\mathbf{x}_{(1)} \mathbf{v}_j) & \cdots & x_{1m} f_j'(\mathbf{x}_{(1)} \mathbf{v}_j) \\ \vdots & \ddots & \vdots \\ x_{N1} f_j'(\mathbf{x}_{(N)} \mathbf{v}_j) & \cdots & x_{Nm} f_j'(\mathbf{x}_{(N)} \mathbf{v}_j) \end{bmatrix} \quad (2.48)$$

com f'_j sendo calculada através da equação (2.40). A informação de segunda ordem pode ser determinada de modo similar.

Conforme destacado anteriormente, a solução deste problema de otimização não linear depende do valor inicial de \mathbf{v}_j . Segundo VON ZUBEN e NETTO (1995), já existem procedimentos estatísticos que podem ser aplicados para fornecer valores iniciais confiáveis.

2.2.3.4 O Procedimento de Retro-Ajuste

Para determinar a dimensão de uma rede neural com uma camada intermediária, toda estratégia de aprendizado construtivo deve ser submetida a um procedimento de retro-ajuste (*backfitting*). No contexto das redes neurais, o retro-ajuste é o processo que promove o reajuste dos neurônios escondidos já existentes logo após a introdução de um novo neurônio. Cada neurônio recém introduzido tem a garantia de estar otimamente definido para a configuração atual da rede neural. Contudo, com a introdução de neurônios adicionais na camada intermediária, o que era ótimo para a configuração anterior, não mais o será para as configurações subseqüentes. Assim, após a introdução de um neurônio escondido adicional, por exemplo o n -ésimo neurônio, todo j -ésimo neurônio ($j = 1, \dots, n$) deve ser atualizado na seqüência e ciclicamente até a convergência.

Para todo j ($j = 1, \dots, n$), o procedimento de retro-ajuste resolverá, seqüencialmente, o seguinte problema de otimização:

$$\min_{\sigma_j, \mathbf{w}_{(j)}} \left\| \sigma_j \mathbf{w}_{(j)} - \mathbf{D}_n^{[j]} \right\| \quad (2.49)$$

com

$$\mathbf{D}_n^{[j]} = \mathbf{S} - \sum_{\substack{i=0 \\ i \neq j}}^n \sigma_i \mathbf{w}_{(i)} \quad (2.50)$$

2.2 Redes Neurais Artificiais Construtivas

2.2.3.5 O Algoritmo de Aprendizado Construtivo

As técnicas convencionais utilizadas para estimar os pesos de uma rede neural aplicam métodos de otimização para ajustar todos os pesos de todas as camadas a cada iteração (época) do processo de treinamento. O método de aprendizado construtivo (UGL), ao contrário, realiza o procedimento de aprendizado neurônio a neurônio e só acrescenta outra unidade à estrutura da rede neural após o conjunto atual de neurônios ter realizado a melhor aproximação possível para um dado mapeamento.

De modo mais específico, o procedimento UGL aplica o método dos mínimos quadrados para estimar os pesos da camada de saída, uma regressão não linear paramétrica uni-dimensional para estimar a função de ativação não linear de cada neurônio escondido e um procedimento de otimização não linear, utilizando derivadas, para estimar os pesos da camada intermediária (VON ZUBEN e NETTO, 1995).

O algoritmo de aprendizado construtivo para múltiplas saídas é apresentado a seguir (VON ZUBEN e NETTO, 1995; VON ZUBEN, 1996):

1. Dados $\mathbf{X} \in \mathbb{R}^{N \times m}$ e $\mathbf{S} \in \mathbb{R}^{N \times r}$, faça $j = 0$, $\mathbf{D} = \mathbf{S}$ e defina valores adequadamente pequenos para ϵ, δ , e γ (limiares utilizados como critérios de parada);
2. Utilizando \mathbf{X} e \mathbf{D} , faça $j = j + 1$ e atribua:
 - um valor inicial para $\mathbf{v}_j \in \mathbb{R}^{m \times 1}$;
 - uma forma inicial para a função de ativação (parâmetros de $f_j(\cdot)$);
3. Extraia os valores médios $\mathbf{m}_{(j)} \in \mathbb{R}^{1 \times r}$ das colunas de \mathbf{D} , resultando em $\tilde{\mathbf{D}} = \mathbf{D} - \mathbf{1}_{N \times r} \mathbf{m}_{(j)}$, onde $\mathbf{1}_{N \times r}$ é uma matriz cujos elementos são números “1”;
4. Determine a direção \mathbf{u}_j (o componente principal de $\tilde{\mathbf{D}}\tilde{\mathbf{D}}^T$);
5. Compute $\boldsymbol{\sigma}_j$ utilizando a equação (2.42) e ajuste $\beta_1 = (\boldsymbol{\sigma}_j^T \mathbf{u}_j)^2 / \boldsymbol{\sigma}_j^T \boldsymbol{\sigma}_j$;
6. Com a direção de projeção \mathbf{v}_j fixa, obtenha uma nova $f_j(\cdot)$ utilizando as equações (2.27), (2.28) e (2.35);
7. Dada a nova $f_j(\cdot)$, calcule \mathbf{v}_j^* que resolva otimamente o problema (2.43);
8. Calcule $\boldsymbol{\sigma}_j$ utilizando a equação (2.42) e ajuste $\beta_2 = (\boldsymbol{\sigma}_j^T \mathbf{u}_j)^2 / \boldsymbol{\sigma}_j^T \boldsymbol{\sigma}_j$;

9. Se $|\beta_1 - \beta_2|/\beta_1 > \epsilon$, faça $\beta_1 = \beta_2$ e repita os passos 6 a 9.
10. Faça $\mathbf{w}_{(j)} = (\boldsymbol{\sigma}_j^T \boldsymbol{\sigma}_j)^{-1} \boldsymbol{\sigma}_j^T \tilde{\mathbf{D}}$;
11. Estime $\hat{\mathbf{S}} = \sum_{a=1}^j (\boldsymbol{\sigma}_a \mathbf{w}_{(a)} + \mathbf{1}_{N \times r} \mathbf{m}_a)$ e faça $E^{velho} = \|\mathbf{S} - \hat{\mathbf{S}}\|$;
12. Para cada b ($b = 1, \dots, j$), faça:

Calcule $\mathbf{D} = \mathbf{S} - \sum_{\substack{a=1 \\ a \neq b}}^j (\boldsymbol{\sigma}_a \mathbf{w}_{(a)} + \mathbf{1}_{N \times r} \mathbf{m}_a)$; faça $j = b$ e repita os passos 3 a 9;
13. Estime $\hat{\mathbf{S}} = \sum_{a=1}^j (\boldsymbol{\sigma}_a \mathbf{w}_{(a)} + \mathbf{1}_{N \times r} \mathbf{m}_a)$ e faça $E^{novo} = \|\mathbf{S} - \hat{\mathbf{S}}\|$;
14. Se $|E^{velho} - E^{novo}|/E^{velho} > \gamma$, faça $E^{velho} = E^{novo}$ e repita os passos 12 a 14;
15. Faça $\mathbf{D} = \tilde{\mathbf{D}} - \hat{\mathbf{S}}$. Se $\|\mathbf{D}\| > \delta$, retorne ao passo 2.

2.3 Considerações Finais Sobre o Capítulo

O objetivo principal deste capítulo foi apresentar a base teórica necessária para o entendimento de um novo paradigma neural, o *modelo neural construtivo*. Este modelo não linear foi apresentado nesta tese como uma solução viável e bastante promissora para os problemas de identificação de processos da indústria química que, não raro, apresentam dimensões (número de entradas e saídas) bastante elevadas. Os modelos neurais obtidos através do algoritmo de aprendizado construtivo (UGL), apresentam-se como uma alternativa bastante atrativa aos modelos não lineares usualmente empregados (incluindo os modelos neurais MLP “tradicionais”) na modelagem de processos químicos. A grande vantagem do algoritmo UGL é que o seu uso propicia o projeto automático de redes neurais com dimensão reduzida, gerando modelos neurais não paramétricos, com uma camada intermediária e com capacidade de generalização potencialmente maior.

Capítulo 3

TÓPICOS EM MODELAGEM DE SISTEMAS

Preâmbulo

Este capítulo apresenta uma breve revisão de algumas das representações matemáticas (modelos) utilizadas com mais frequência na identificação de sistemas dinâmicos de tempo discreto lineares e não lineares. Outros modelos empregados com menos frequência na literatura são abordados neste capítulo devido à sua importância nas estratégias de identificação e controle propostas nesta tese.

A seção 3.1 apresenta uma discussão inicial sobre a motivação para o estabelecimento de diferentes descrições matemáticas de sistemas dinâmicos. Na seção 3.2 são formalizados a terminologia e os conceitos básicos sobre *sistemas lineares*, *estrutura e preditor* do modelo e a definição de *vetor de regressão*. As principais características das *representações lineares discretas* do tipo FIR, Série Ortonormal, ARX, ARMAX e do Modelo de Erro na Saída são apresentadas na seção 3.3. As *representações não lineares discretas* do tipo NARX, NARMAX, Funções de Base Radial, Modelos de Volterra, Modelos Nebulosos e Modelos com Estrutura Hierárquica são discutidas na seção 3.4. A formalização destes conceitos é importante não só no âmbito da modelagem de sistemas mas também no contexto do Capítulo 4, onde serão estabelecidas algumas analogias entre a identificação de sistemas utilizando modelos lineares e não lineares.

3.1 Introdução

Entende-se por *modelo*, a representação matemática de um sistema real que, idealmente, pode ser utilizada para representar ao menos as suas principais características. As técnicas de modelagem podem ser agrupadas em duas grandes, e não necessariamente desconexas, categorias (AMARAL, 1980; SÖDERSTRÖM e STOICA, 1989; AGUIRRE *et al.*, 1998). A primeira é a modelagem pela física do processo, também conhecida como análise física-matemática ou ainda modelagem fenomenológica ou determinística. A segunda é conhecida como modelagem a partir de amostragem, identificação experimental ou ainda identificação de sistemas (LJUNG, 1999).

Existem vários motivos para o estabelecimento de descrições matemáticas de sistemas dinâmicos, dentre elas destacam-se as aplicações do modelo obtido em: simulação do comportamento dinâmico do processo em diferentes condições operacionais, incorporando possíveis perturbações e a subsequente avaliação dos impactos causados; projeto e dimensionamento de equipamentos; detecção de falhas em sensores; otimização das condições operacionais; treinamento de operadores e projeto de sistemas de controle.

Dependendo do nível de conhecimento inicial do sistema, o problema de modelagem pode ser abordado de maneiras diferentes. A terminologia *modelagem caixa-branca* é utilizada quando o sistema é modelado totalmente a partir de leis físico-químicas (modelagem fenomenológica). Os modelos fenomenológicos são, em tese, potencialmente úteis para a adequada investigação do comportamento de sistemas reais nas mais variadas condições de operação. Se a identificação é baseada exclusivamente em dados amostrados, assumindo muito pouco ou nenhum conhecimento sobre o sistema, o processo de identificação é chamado de *modelagem caixa-preta*. Uma abordagem intermediária, a *modelagem caixa-cinza*, não exige um profundo conhecimento *a priori* sobre o sistema mas permite que esta informação, caso esteja disponível, seja incorporada ao modelo (AGUIRRE *et al.*, 1998).

Por vezes, o trabalho associado à construção de um modelo baseado em leis fenomenológicas (balanços de massa, energia e quantidade de movimento) é extenuante ou mesmo impossível de ser realizado. Em tais situações, as demais técnicas de identificação de sistemas se apresentam como uma alternativa viável e natural.

3.1 Introdução

Identificação de sistemas, segundo AGUIRRE (2000), é a área de conhecimento que estuda e desenvolve técnicas e algoritmos para obter (identificar) modelos de sistemas dinâmicos a partir de dados gerados pelo próprio sistema, ou seja, são procedimentos que permitem desenvolver modelos matemáticos a partir de dados e sinais observados. O interesse no desenvolvimento de metodologias para obtenção de modelos matemáticos a partir de dados observados (amostrados) e não apenas a partir de equações diferenciais e algébricas tem sido crescente desde a década de 60. Existem muitas razões para essa mudança de interesse, dentre elas destacam-se (AGUIRRE, 2000): *i)* a crescente complexidade dos sistemas com os quais se precisa lidar, o que nem sempre permite representar o processo através de suas equações fundamentais¹; e *ii)* a disponibilização no mercado de computadores pessoais (PCs) com boa capacidade de processamento a preços acessíveis. Este fato viabilizou o uso de PCs em tarefas como a coleta direta de dados de sistemas, além do desenvolvimento, a partir desses dados, de modelos matemáticos capazes de explicar o comportamento do sistema de interesse².

O grande apelo da modelagem caixa-preta é devido principalmente à dificuldade na obtenção de um tratamento matemático genérico para sistemas não lineares, pois é praticamente impossível representar adequadamente todo tipo de conhecimento em sistemas não lineares. Um outro problema é que, mesmo que algum conhecimento sobre o sistema esteja disponível, relacioná-lo corretamente a uma descrição dinâmica do sistema contínua no tempo (isto é, em termos de equações diferenciais) pode ser uma tarefa extremamente difícil. Portanto, a opção de escolher a modelagem caixa-preta não é incomum, mesmo quando um certo nível de conhecimento sobre o sistema está disponível. Contudo, um bom entendimento sobre as leis físicas que regem o comportamento do sistema é sempre útil e facilita bastante o processo de identificação. Tal conhecimento pode estar relacionado à ordem do sistema, à dinâmica principal do sistema (rápida ou lenta), à frequência de amostragem adequada, às características de estabilidade, à faixa de operação em que se pretende operar o sistema, tempos de atraso, grau de não linearidade, entre outros.

Por outro lado, a despeito da maioria dos sistemas apresentarem ao menos uma leve não linearidade, a maior parte da literatura sobre identificação de sistemas lida com

¹Modelagem fenomenológica.

²Modelagem empírica ou identificação de sistemas.

identificação de sistemas lineares. Existem várias razões para isto, mas algumas das mais importantes são (NØRGAARD *et al.*, 2000):

- Muitos sistemas podem ser bem descritos por um modelo linear e, quando não é razoável usar um único modelo para toda a faixa de operação, muitas vezes é possível identificar diferentes modelos lineares para os diferentes regimes da faixa de operação.
- Do ponto de vista computacional, a implementação de modelos lineares é menos complicada.
- Similarmente, a análise estatística também é mais simples quando se trata de sistemas lineares.
- É muito mais simples projetar controladores para um sistema descrito por modelos lineares.

3.2 Definições e Conceitos Fundamentais

3.2.1 Sistemas Lineares

Sistemas lineares podem ser definidos formalmente como todo sistema que satisfaz o *princípio da superposição* (KALMAN *et al.*, 1969; AGUIRRE, 2000). Este princípio pode ser resumido da seguinte forma: Seja um sistema que ao ser excitado pela entrada $u_1(t)$ produz a saída $y_1(t)$ e quando excitado pela entrada $u_2(t)$ produz $y_2(t)$. Se tal sistema satisfizer o princípio da superposição então, quando for excitado por $au_1(t) + bu_2(t)$, sua saída será $ay_1(t) + by_2(t)$, sendo a e b constantes reais.

De acordo com NØRGAARD *et al.* (2000), um sistema é denominado linear se for possível representá-lo através de um modelo que assume a seguinte forma:

$$y(t) = G(q^{-1})u(t) + H(q^{-1})\nu(t) \quad (3.1)$$

onde G e H são funções de transferência definidas no operador de atraso, q^{-1} , e $\nu(t)$ é um sinal de ruído branco que é independente das entradas passadas e que pode ser caracterizado por alguma função de densidade probabilística. O operador de atraso atua sobre o sinal conforme descrito a seguir:

$$q^{-d}x(t) = x(t - d) \quad (3.2)$$

onde d é um múltiplo do período de amostragem.

No caso de sistemas multi-variáveis (MIMO), $u(t)$, $y(t)$ e $\nu(t)$ são vetores enquanto G e H são matrizes polinomiais em q^{-1} .

3.2.2 Terminologia Básica

Considere que o **Sistema Verdadeiro** (linear) seja descrito por:

$$y(t) = G_0(q^{-1})u(t) + H_0(q^{-1})\nu_0(t) \quad (3.3)$$

onde $\nu_0(t)$ é um ruído branco independente do sinal de entrada $u(t)$.

A **Estrutura do Modelo**, \mathcal{M} , é um conjunto parametrizado de modelos candidatos:

$$\mathcal{M}(\theta) : \{G(q^{-1}, \theta), H(q^{-1}\theta) | \theta \in \mathcal{D}_m\} \quad (3.4)$$

$$y(t) = G(q^{-1}, \theta)u(t) + H(q^{-1}, \theta)\nu(t) \quad (3.5)$$

onde θ denota os p parâmetros a serem estimados e \mathcal{D}_m é algum subconjunto de \mathbb{R}^p dentro do qual a busca por um modelo deve ser conduzida.

Uma representação alternativa para o modelo descrito pela equação (3.5), geralmente denominada **Preditor do Modelo** (um passo adiante), é dada por:

$$\hat{y}(t | t-1, \theta) = H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t) + [1 - H^{-1}(q^{-1}, \theta)]y(t) \quad (3.6)$$

A inclusão de θ como um novo argumento na equação (3.6) implica que a estrutura do modelo representa um conjunto de modelos.

A estrutura do modelo pode ser escrita da seguinte forma:

$$\hat{y}(t | t-1, \theta) = \boldsymbol{\varphi}^T(t)\boldsymbol{\theta} \quad (3.7)$$

onde $\boldsymbol{\theta}$ é o vetor de parâmetros e $\boldsymbol{\varphi}$ é o **Vetor de Regressão** (ou regressores) do modelo que contém as entradas e saídas passadas.

A exigência básica para o conjunto \mathcal{D}_m é (SÖDERSTRÖM e STOICA, 1989):

$$\mathcal{D}_m = \left\{ \theta \left| \begin{array}{ll} H^{-1}(q^{-1}, \theta)G^{-1}(q^{-1}, \theta) & \text{assintoticamente estável} \\ H^{-1}(q^{-1}, \theta) & \text{assintoticamente estável} \\ G(0, \theta) = 0, \quad H(0, \theta) = 1 \end{array} \right. \right\} \quad (3.8)$$

3.2 Definições e Conceitos Fundamentais

porém, conhecimentos sobre as características físicas do sistema podem impor restrições adicionais sobre o conjunto. A condição de estabilidade garante que o preditor seja estável. A hipótese $G(0) = 0$ é incluída para assegurar que as predições dependam somente das entradas passadas enquanto que $H(0) = 1$ assegura que as predições dependam somente das saídas medidas passadas (NØRGAARD *et al.*, 2000).

Uma outra exigência básica é que a estrutura do modelo seja “grande o suficiente” para descrever o sistema real, ou seja

$$\mathcal{S} \in \mathcal{M} \tag{3.9}$$

Contudo, na maior parte dos casos, considera-se uma estrutura de modelo mais simples do que a representada pela equação (3.5). As simplificações, que são feitas considerando o processo estacionário no sentido amplo e densidade espectral racional (SÖDERSTRÖM e STOICA, 1989), diferem umas das outras basicamente devido às considerações feitas sobre a densidade espectral do ruído e sobre como o ruído é adicionado ao sistema. Algumas das simplificações mais freqüentemente empregadas assim como os modelos resultantes serão discutidos nas seções 3.3 e 3.4.

3.3 Representações Matemáticas Lineares de Tempo Discreto

Conforme visto na seção 3.1, um modelo matemático de um sistema pode ser definido como uma representação matemática desse sistema. Contudo, há diversas maneiras de representar o mesmo modelo matemático. Cada uma das possíveis maneiras de representar um modelo matemático é chamada de *representação*. Existem algumas representações matemáticas que são especialmente adequadas para a identificação de sistemas utilizando-se algoritmos já estabelecidos para estimação de parâmetros. A seguir, serão apresentadas as representações matemáticas utilizadas com mais frequência na identificação de sistemas lineares.

Considere o seguinte modelo geral:

$$A(q^{-1})y(k) = q^{-d} \frac{B(q^{-1})}{F(q^{-1})}u(k) + \frac{C(q^{-1})}{D(q^{-1})}\nu(k) \quad (3.10)$$

onde q^{-1} é o operador de atraso definido de forma que $y(k)q^{-1} = y(k-1)$, $\nu(k)$ é o ruído branco e $A(q^{-1}), B(q^{-1}), C(q^{-1}), D(q^{-1})$ e $F(q^{-1})$ são polinômios definidos da seguinte forma:

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_{n_y}q^{-n_y} \\ B(q^{-1}) &= b_0 + b_1q^{-1} + \dots + b_{n_u}q^{-n_u} \\ C(q^{-1}) &= 1 + c_1q^{-1} + \dots + c_{n_\xi}q^{-n_\xi} \\ D(q^{-1}) &= 1 + d_1q^{-1} + \dots + d_{n_d}q^{-n_d} \\ F(q^{-1}) &= 1 + f_1q^{-1} + \dots + f_{n_f}q^{-n_f} \end{aligned} \quad (3.11)$$

3.3.1 Modelo de Resposta Finita ao Impulso (FIR)

O *somatório de convolução* entre a resposta ao impulso, $h(k)$, de um sistema e um sinal de entrada, $u(k)$, resulta na saída do sistema com ruído, $e(k)$, representado da seguinte forma:

3.3 Representações Matemáticas Lineares de Tempo Discreto

$$y(k) = \sum_{j=0}^{\infty} h(j)u(k-j) + \nu(k) \quad (3.12)$$

A equação (3.12) é conhecida como modelo de resposta infinita ao impulso. Se $H(q)$ for estável, então existe $M < \infty$ tal que $h(k) < \epsilon, \forall k > M$. Portanto, truncando a equação (3.12), obtém-se o modelo de resposta finita ao impulso, FIR³:

$$y(k) = \sum_{j=0}^M h(j)u(k-j) + e(k) \quad (3.13)$$

O modelo FIR pode ser obtido a partir da equação (3.10) tomando-se $A(q^{-1}) = C(q^{-1}) = D(q^{-1}) = F(q^{-1}) = 1$ e $B(q^{-1})$ um polinômio arbitrário de ordem M , ou seja, $n_u = M$ (equação (3.11)). Nesse caso, o modelo FIR (3.13) pode ser escrito como (AGUIRRE, 2000):

$$y(k) = q^{-d}B(q^{-1})u(k) + e(k) \quad (3.14)$$

Uma vez que $C(q^{-1})/D(q^{-1}) = 1$, o ruído $e(k)$ acrescentado na saída $y(k)$ é branco pois, neste caso, $e(k) = \nu(k)$.

O modelo FIR também pode ser construído a partir da equação (3.5), através da seguinte escolha:

$$G(q^{-1}, \theta) = q^{-d}B(q^{-1}) \quad H(q^{-1}, \theta) = 1 \quad (3.15)$$

cujos preditor é dado por

$$\hat{y}(k | k-1, \theta) = q^{-d}B(q^{-1})u(k) \quad (3.16)$$

³*Finite Impulse Response.*

que por sua vez pode ser representado na forma descrita pela equação (3.7) com o vetor de regressão, $\varphi(k)$, e o vetor de parâmetros, θ , definidos da seguinte forma:

$$\varphi(k) = [u(k-d) \quad \dots \quad u(k-d-n_u)]^T \quad (3.17)$$

$$\theta = [b_0 \quad \dots \quad b_{n_u}]^T \quad (3.18)$$

Um sistema dinâmico não pode ser descrito exatamente através de um modelo FIR. Contudo, se o sistema for estável e a sua resposta ao impulso decair rapidamente, o sistema normalmente pode ser bem aproximado por um modelo FIR se $B(q^{-1})$ for selecionado com os primeiros n_u coeficientes da resposta ao impulso (NØRGAARD *et al.*, 2000).

3.3.2 Modelos com Funções de Base Ortonormal (OBF)

As funções de base ortonormal (OBF⁴) vêm sendo aplicadas na identificação de sistemas dinâmicos lineares e não lineares há várias décadas. As primeiras aplicações das OBF neste contexto são atribuídas a WIENER (1956) e BROOME (1965).

É sabido que um sistema linear estável pode ser completamente caracterizado por sua resposta ao impulso $h(k)$. Admitindo que o processo seja causal⁵ e que $h(k)$ pertença ao espaço de Lebesgue ($L_2[0, \infty[$), então o processo pode ser modelado, através de $h(k)$, por uma série de funções ortonormais:

$$h(k) = \sum_{i=1}^{\infty} c_i \phi_i(k) \quad (3.19)$$

onde $\phi_i(k)$ é uma base de funções ortonormais e c_i são os parâmetros associados ao desenvolvimento em série da resposta ao impulso $h(k)$. Esses parâmetros são definidos da seguinte forma:

⁴ *Orthonormal Basis Function.*

⁵ Sua saída não depende das entradas em tempos futuros.

3.3 Representações Matemáticas Lineares de Tempo Discreto

$$c_i = \sum_{k=0}^{\infty} h(k) \phi_i(k) \quad (3.20)$$

As funções de base utilizadas com mais frequência são as de Laguerre e Kautz pois elas formam uma base completa no espaço de Lebesgue (OLIVEIRA *et al.*, 1999). Isso significa que qualquer função descrita neste espaço pode ser modelada com precisão arbitrária através de uma combinação linear da série descrita pela equação (3.19).

A base de funções ortonormais de Laguerre é caracterizada pela utilização de funções de transferência com apenas um pólo:

$$\Phi_{laguerre,i}(q^{-1}) = \sqrt{1-p^2} \frac{q^{-1}(q^{-1}-p)^{i-1}}{(1-pq^{-1})^i}, \quad i = 1, \dots \quad (3.21)$$

onde $p = \{p : p \in \Re \text{ e } |p| < 1\}$ é o pólo que parametriza as funções ortonormais e q^{-1} é o operador de atraso.

Para fins práticos de modelagem, considera-se somente uma quantidade finita de funções, ϕ_i , para representar as dinâmicas do processo. Isto é feito truncando-se a equação (3.19) da seguinte forma:

$$\tilde{h}(k) = \sum_{i=1}^n c_i \phi_i(k) \quad (3.22)$$

onde $\tilde{h}(k)$ é a aproximação de ordem n de $h(k)$, sendo n escolhido de forma arbitrária com implicações na precisão da representação do processo.

Assim, pode-se representar a relação entrada-saída de um processo através de uma série truncada de funções ortonormais do seguinte modo:

$$\tilde{y}(k) = \sum_{i=1}^n c_i \Phi_i(q^{-1}) u(k) \quad (3.23)$$

ou, alternativamente

$$\tilde{y}(k) = \sum_{i=1}^n c_i l_i(k) \quad (3.24)$$

onde $u(k)$ é o sinal de controle, $\tilde{y}(k)$ é a saída do processo e $l_i(k)$ é a saída da i -ésima função ortonormal $\Phi_i(q^{-1})$.

Pode-se escrever a equação (3.24) em uma forma mais compacta empregando a notação vetorial, ou seja:

$$\tilde{y}(k) = \mathbf{c}^T \mathbf{l}(k) \quad (3.25)$$

com \mathbf{c} e \mathbf{l} definidos como

$$\mathbf{c} = [c_1 \ c_2 \ \dots \ c_n]^T \quad (3.26)$$

$$\mathbf{l}(k) = [l_1(k) \ l_2(k) \ \dots \ l_n(k)]^T \quad (3.27)$$

onde o vetor \mathbf{c} contém os coeficientes da série ortonormal e o vetor \mathbf{l} contém as saídas das n funções ortonormais.

A saída do modelo gerada pelas OBF pode então ser descrita como sendo a soma ponderada de todas as saídas das funções de base, onde cada função que forma uma base é definida através da especificação de um ou mais pólos. Estes pólos geralmente são selecionados a partir do conhecimento *a priori* sobre a dinâmica do processo.

Uma vez que as funções de Laguerre são recursivas, isto é, a função ortonormal $\Phi_i(q^{-1})$ pode ser calculada a partir de $\Phi_{i-1}(q^{-1})$, pode-se então reescrever o modelo do processo, descrito pela equação (3.23), através da representação no espaço de estados (OLIVEIRA *et al.*, 1999):

$$\mathbf{l}(k+1) = \mathbf{A}\mathbf{l}(k) + \mathbf{b}u(k) \quad (3.28)$$

Desse modo, um modelo baseado na série ortonormal de Laguerre com somente uma dinâmica (um pólo) pode ser descrito da seguinte forma:

3.3 Representações Matemáticas Lineares de Tempo Discreto

$$\begin{aligned}\mathbf{l}(k+1) &= A\mathbf{l}(k) + \mathbf{b}u(k) \\ \tilde{y}(k) &= \mathbf{c}^T\mathbf{l}(k)\end{aligned}\tag{3.29}$$

que pode ser facilmente adaptado para incorporar o atraso de transporte ao sinal de entrada simplesmente trocando $u(k)$ por $u(k - \tau)$:

$$\begin{aligned}\mathbf{l}(k+1) &= A\mathbf{l}(k) + \mathbf{b}u(k - \tau) \\ \tilde{y}(k) &= \mathbf{c}^T\mathbf{l}(k)\end{aligned}\tag{3.30}$$

Uma grande vantagem desta abordagem é que o modelo final é linear nos parâmetros que, portanto, podem ser estimados através de métodos de identificação clássicos como os mínimos quadrados. Além disso, o sucesso do emprego de modelos dinâmicos baseados em funções de base ortonormais na modelagem e controle de sistemas lineares e não lineares também se deve às seguintes características (CAMPELLO *et al.*, 2001): *i)* Ausência de realimentação da saída (o que evita todos os problemas associados à realimentação de erros de predição); *ii)* Não há a necessidade de determinar previamente a ordem do modelo nem o atraso de transporte, eliminando assim a etapa (geralmente árdua) de determinação dos regressores; *iii)* É possível aumentar a capacidade de aproximação dos modelos simplesmente aumentando o número de funções; *iv)* Os modelos OBF apresentam grande tolerância às dinâmicas não modeladas e sensibilidade reduzida aos parâmetros estimados; e *v)* Capacidade de lidar de forma robusta com atrasos de transporte.

Recentemente, OLIVEIRA *et al.* (1999) apresentaram estudos sobre aplicação de modelos neurais e nebulosos desenvolvidos a partir de bases de funções ortonormais na identificação de sistemas dinâmicos.

Seguindo esta abordagem, CAMPELLO *et al.* (2001) apresentaram uma alternativa para o problema da estimação do grande número de parâmetros associados à utilização de modelos com estrutura NFIR através do emprego de modelos baseados em funções ortonormais. Neste trabalho (CAMPELLO *et al.*, 2001), modelos lineares e não lineares baseados nas funções ortonormais de Laguerre foram utilizados para modelar um sistema industrial de grande porte. Os modelos não lineares, dentre eles um

modelo de Volterra de segunda ordem, foram capazes de descrever adequadamente o comportamento dinâmico do processo para previsões em horizontes longos de tempo de predição, mesmo na presença de perturbações e de ruídos na variável medida.

3.3.3 Modelo Auto-Regressivo com Entradas Externas (ARX)

A estrutura do modelo ARX⁶ corresponde a reescrever a equação (3.10) tomando $C(q^{-1}) = D(q^{-1}) = F(q^{-1}) = 1$ e $A(q^{-1})$ e $B(q^{-1})$ como polinômios arbitrários, resultando em:

$$A(q^{-1})y(k) = B(q^{-1})u(k) + \nu(k) \quad (3.31)$$

Como o ruído $\nu(k)$ aparece explicitamente na equação (3.31), o modelo ARX é classificado como pertencendo à classe de modelos com *erro na equação* (AGUIRRE, 2000). A exemplo do modelo FIR, o modelo ARX também pode ser reescrito na forma:

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})}u(k) + \frac{1}{A(q^{-1})}\nu(k) \quad (3.32)$$

Diferente do modelo FIR, o ruído adicionado à saída, $e(k) = \nu/A(q^{-1})$, não é branco, ou seja, neste modelo o ruído é modelado como ruído branco filtrado por um filtro auto-regressivo.

Uma forma alternativa de representar a estrutura do modelo ARX, baseada na equação (3.5), corresponde escolher:

$$G(q^{-1}, \theta) = q^{-d} \frac{B(q^{-1})}{A(q^{-1})} \quad H(q^{-1}, \theta) = \frac{1}{A(q^{-1})} \quad (3.33)$$

cujo preditor toma a seguinte forma:

$$\begin{aligned} \hat{y}(k | k-1, \theta) &= q^{-d} B(q^{-1})u(k) + [1 - A(q^{-1})]y(k) \\ &= \boldsymbol{\varphi}^T(k) \boldsymbol{\theta} \end{aligned} \quad (3.34)$$

⁶ *AutoRegressive with eXogenous inputs.*

3.3 Representações Matemáticas Lineares de Tempo Discreto

com

$$\boldsymbol{\varphi}(k) = [y(k-1) \ \dots \ y(k-n_y), \ u(k-d) \ \dots \ u(k-d-n_u)]^T \quad (3.35)$$

$$\boldsymbol{\theta} = [-a_1 \ \dots \ -a_{n_y}, \ b_0 \ \dots \ b_{n_u}]^T \quad (3.36)$$

Embora a função de transferência G agora tenha pólos, ainda há somente uma relação algébrica entre a predição e as entradas passadas e saídas medidas (NØRGAARD *et al.*, 2000).

3.3.4 Modelo Auto-Regressivo com Média Móvel e Entradas Externas (ARMAX)

O modelo ARMAX⁷ pode ser obtido a partir do modelo geral representado pela equação (3.10) tomando-se $D(q^{-1}) = F(q^{-1}) = 1$ e $A(q^{-1})$, $B(q^{-1})$ e $C(q^{-1})$ como polinômios arbitrários, resultando em:

$$A(q^{-1})y(k) = B(q^{-1})u(k) + C(q^{-1})\nu(k) \quad (3.37)$$

ou ainda:

$$\begin{aligned} y(k) &= \frac{B(q^{-1})}{A(q^{-1})}u(k) + \frac{C(q^{-1})}{A(q^{-1})}\nu(k) \\ y(k) &= H(q^{-1})u(k) + e(k) \end{aligned} \quad (3.38)$$

Assim como o modelo ARX, o modelo ARMAX também é classificado como um modelo de *erro na equação*. Neste caso, o erro é modelado como um processo de média móvel e o ruído adicionado à saída, $e(k)$ (não branco), é modelado como um ruído branco filtrado por um filtro ARMA (AGUIRRE, 2000).

⁷ *AutoRegressive Moving Average with eXogenous inputs.*

Escolhendo os polinômios G e H (equação (3.5)) de modo que

$$G(q^{-1}, \theta) = q^{-d} \frac{B(q^{-1})}{A(q^{-1})} \quad H(q^{-1}, \theta) = \frac{C(q^{-1})}{A(q^{-1})} \quad (3.39)$$

obtem-se o seguinte preditor:

$$\begin{aligned} \hat{y}(k | k-1, \theta) &= q^{-d} \frac{B(q^{-1})}{C(q^{-1})} u(k) + \left(1 - \frac{A(q^{-1})}{C(q^{-1})}\right) y(k) \\ &= q^{-d} B(q^{-1}) u(k) + [1 - A(q^{-1})] y(k) + [C(q^{-1}) - 1] \epsilon(k, \theta) \\ &= \boldsymbol{\varphi}^T(k, \theta) \boldsymbol{\theta} \end{aligned} \quad (3.40)$$

onde $\epsilon(k, \theta) = y - \hat{y}(k | k-1, \theta)$ representa o *erro de predição* ou *resíduo*. Os vetores de regressão e de parâmetros são definidos, respectivamente, por

$$\begin{aligned} \boldsymbol{\varphi}(k, \theta) &= [y(k-1) \quad \dots \quad y(k-n_y), \quad u(k-d) \quad \dots \quad u(k-d-n_u), \\ &\quad \epsilon(k, \theta) \quad \dots \quad \epsilon(k-n_\xi, \theta)]^T \end{aligned} \quad (3.41)$$

$$\boldsymbol{\theta} = [-a_1 \quad \dots \quad -a_{n_y}, \quad b_0 \quad \dots \quad b_{n_u}, \quad c_1 \quad \dots \quad c_{n_\xi}]^T \quad (3.42)$$

Devido à presença do polinômio C , o preditor do modelo ARMAX apresenta pólos. Deste modo, C deve ter suas raízes dentro do círculo unitário no plano z para garantir a estabilidade do preditor. Além disso, a presença dos pólos implica que o vetor de regressão depende dos parâmetros do modelo, o que faz com que a sua estimação seja mais elaborada (NØRGAARD *et al.*, 2000). A dependência dos parâmetros do modelo está indicada pela inclusão de θ como argumento de $\boldsymbol{\varphi}$ na equação (3.41).

3.3.5 Modelo de Erro na Saída (OE)

O modelo com a estrutura OE⁸, também conhecido como *modelo com estrutura paralela*, é utilizado quando o único tipo de ruído que afeta o sistema é do tipo ruído branco (NØRGAARD *et al.*, 2000). Esta estrutura de modelos é representada por

⁸ *Output Error*.

3.3 Representações Matemáticas Lineares de Tempo Discreto

$$y(k) = q^{-d} \frac{B(q^{-1})}{F(q^{-1})} u(k) + e(k) \quad (3.43)$$

que corresponde a escolher G e H (3.5) como

$$G(q^{-1}, \theta) = q^{-d} \frac{B(q^{-1})}{F(q^{-1})}; \quad H(q^{-1}, \theta) = 1 \quad (3.44)$$

e leva ao seguinte preditor:

$$\begin{aligned} \hat{y}(k \mid k-1, \theta) &= q^{-d} \frac{B(q^{-1})}{F(q^{-1})} u(k) \\ &= q^{-d} B(q^{-1}) u(k) + [1 - F(q^{-1})] \hat{y}(k \mid k-1, \theta) \\ &= \boldsymbol{\varphi}^T(k, \theta) \boldsymbol{\theta} \end{aligned} \quad (3.45)$$

onde

$$\boldsymbol{\varphi}(k, \theta) = [\hat{y}(k-1 \mid \theta) \quad \dots \quad \hat{y}(k-n_f \mid \theta), \quad u(k-d) \quad \dots \quad u(k-d-n_u)]^T \quad (3.46)$$

$$\boldsymbol{\theta} = [-f_1 \quad \dots \quad -f_{n_f}, \quad b_0 \quad \dots \quad b_{n_u}]^T \quad (3.47)$$

Para que o preditor seja estável, as raízes de F devem estar dentro do círculo unitário no plano z .

3.4 Representações Matemáticas Não Lineares de Tempo Discreto

Recentemente tem-se notado um grande interesse na aplicação de representações não lineares na modelagem de sistemas reais. Esta mudança de enfoque, da abordagem linear para a não linear, pode ser em parte explicada pela crescente disponibilidade de computadores com capacidade de processamento cada vez maior, o que possibilita a análise e a reprodução de sistemas com comportamentos dinâmicos mais complexos. Contudo, a obtenção de modelos não lineares é significativamente mais trabalhosa do que a de modelos lineares, além do fato de que ainda há certas etapas na modelagem de sistemas não lineares que não estão bem estabelecidas (AGUIRRE *et al.*, 1998).

3.4.1 Modelos NARX e NARMAX

Os modelos com estrutura NARX⁹ são modelos de tempo discreto que fornecem o valor da saída do sistema, $y(k)$, em função de valores passados da própria saída e da entrada, u , ou seja:

$$y(k) = F[y(k-1), \dots, y(k-n_y), u(k-d), \dots, u(k-n_u)] \quad (3.48)$$

onde n_y , n_u e d são os atrasos na saída, na entrada e o tempo morto, respectivamente.

Assim como os sistemas lineares, os modelos não lineares podem apresentar polarização quando se utiliza o estimador de mínimos quadrados na etapa de estimação dos parâmetros do modelo. Isso pode ocorrer quando o ruído ou erro na equação de regressão não for branco, ou seja, quando o ruído for auto-correlacionado (colorido) e o modelo apresentar regressores da saída (AGUIRRE, 2000). A fim de evitar esta polarização de parâmetros, é comum incluir termos de ruído no modelo. Modelos que apresentam esta estrutura são chamados de NARMAX¹⁰ e são escritos genericamente da seguinte forma:

⁹*Nonlinear AutoRegressive with eXogenous inputs.*

¹⁰*Nonlinear AutoRegressive Moving Average with eXogenous inputs.*

3.4 Representações Matemáticas Não Lineares de Tempo Discreto

$$y(k) = F[y(k-1), \dots, y(k-n_y), u(k-d), \dots, u(k-n_u)e(k), e(k-1), \dots, e(k-n_e)] \quad (3.49)$$

onde $e(k)$ é o ruído e n_e é o atraso no modelo relativo ao ruído.

Uma dificuldade inerente a esses tipos de modelos é determinar a função não linear F . Duas das representações mais utilizadas para a função F são a polinomial e a racional.

Um modelo polinomial NARMAX sem atraso puro de tempo é escrito da seguinte forma:

$$y(k) = \sum_i c_i \prod_{j=1}^{n_y} y(k-j) \prod_{r=1}^{n_u} u(k-r) \prod_{q=0}^{n_e} e(k-q) \quad (3.50)$$

Enquanto que um modelo NARMAX racional é dado por:

$$y(k) = \frac{\sum_i c_i \prod_{j=1}^{n_y} y(k-j) \prod_{r=1}^{n_u} u(k-r) \prod_{q=1}^{n_e} e(k-q)}{\sum_i d_i \prod_{j=1}^{d_y} y(k-j) \prod_{r=1}^{d_u} u(k-r) \prod_{q=1}^{d_e} e(k-q)} + e(k) \quad (3.51)$$

Os modelos polinomiais representados genericamente pela equação (3.50), apesar de permitirem o uso de regressores não lineares nos sinais de saída, entrada e ruído, são lineares nos parâmetros. Isto significa que modelos deste tipo podem ser representados da forma

$$y(k) = \boldsymbol{\psi}^T \boldsymbol{\theta} \quad (3.52)$$

ou seja, $\boldsymbol{\psi}$ e $\boldsymbol{\theta}$ têm a mesma estrutura dos preditores lineares (seção 3.2.2; eq. (3.7)). Consequentemente, o vetor de parâmetros $\boldsymbol{\theta}$ pode ser estimado com o algoritmo clássico dos mínimos quadrados (LJUNG, 1999).

Quando comparados com os modelos polinomiais, os modelos racionais apresentam uma estrutura mais flexível e por isso tendem a ser mais eficientes na modelagem

de sistemas não lineares. Por outro lado, os modelos racionais, além de serem mais sensíveis a ruídos, são não lineares nos parâmetros e por isso precisam de algoritmos mais sofisticados para estimá-los (AGUIRRE, 2000).

3.4.2 Modelos com Funções de Base Radial

As representações não lineares baseadas nas funções de base radial (RBF¹¹) podem ser interpretadas como uma técnica de interpolação global com boas propriedades locais (AGUIRRE, 2000).

As RBF são mapeamentos do tipo

$$f(\mathbf{y}) = \omega_0 + \sum_i \omega_i \phi(\|\mathbf{y} - \mathbf{c}_i\|) \quad (3.53)$$

onde ω_i são os pesos, \mathbf{c}_i são os centros, $\|\cdot\|$ é a norma euclidiana e $\phi(\cdot) : \Re^+ \rightarrow \Re$ é uma função de base radial, geralmente escolhida *a priori*, como por exemplo a função de base gaussiana:

$$\phi(\|\mathbf{y} - \mathbf{c}_i\|) = \exp\left(-\frac{(\|\mathbf{y} - \mathbf{c}_i\|)^2}{\sigma_i^2}\right) \quad (3.54)$$

onde σ define a abertura da gaussiana.

As funções de base radial são casos particulares de redes neurais (AGUIRRE, 2000) mas, ao contrário destas, são lineares nos parâmetros ω_i , conforme pode ser verificado na equação (3.53). Desse modo, se a função $\phi(\cdot)$ e os centros \mathbf{c}_i forem conhecidos, os parâmetros ω_i podem ser estimados utilizando o método dos mínimos quadrados.

O acréscimo de termos auto-regressivos lineares, assim como termos adicionais da entrada na representação (3.53) é procedimento bastante comum no contexto de identificação de sistemas. Assim, a equação (3.53) é reescrita da seguinte forma:

¹¹*Radial Basis Functions.*

3.4 Representações Matemáticas Não Lineares de Tempo Discreto

$$y(k) = \omega_0 + \sum_i \omega_i \phi(\|\mathbf{y}(k-1) - \mathbf{c}_i\|) + \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=1}^{n_u} a_i u(k-i) + e(k) \quad (3.55)$$

onde $\mathbf{y}(k-1) = [y(k-1) \dots y(k-n_y) \ u(k-1) \dots u(k-n_u)]^T$ e $e(k)$ é o erro. Caso o sistema apresente atraso puro de tempo, τ_d , os termos lineares de entrada começariam com $u(k - \tau_d - 1)$.

3.4.3 Modelos de Volterra

Pode-se representar a saída de um sistema não linear, $y(t)$, a partir de sua entrada, $u(t)$, através da representação chamada de *Série de Volterra*¹²:

$$y(t) = \sum_{j=1}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_j(\tau_1, \dots, \tau_j) \prod_{i=1}^j u(t - \tau_i) d\tau_i \quad (3.56)$$

onde as funções h_j são denominadas *kernels* ou núcleos de Volterra. Embora o limite de integração, Γ , seja dado genericamente por $[-\infty, \infty]$, as representações mais usuais consideram $\Gamma = [0, \infty]$. Se além disso ainda for considerado que $u(t) = 0$ para $\forall t < 0$, tem-se então o novo limite de integração dado por $\Gamma = [0, t]$.

Tais funções podem ser vistas como generalizações não lineares da resposta ao impulso $h(t)$. Isto fica claro quando se faz uma analogia com um sistema linear ($j = 1$), onde a equação (3.56) reduz-se a uma integral de convolução (AGUIRRE, 2000).

A representação descrita pela equação (3.56) pode ser truncada para produzir (se a série for convergente) uma aproximação de ordem finita, η , em um ponto além do qual os termos de ordens superiores podem ser desprezados (CAMPELLO, 2002):

$$y(t) = \sum_{n=1}^{\eta} \int_{\Gamma} \dots \int_{\Gamma} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n u(t - \tau_i) d\tau_i \quad (3.57)$$

que na forma discreta é dada por:

¹²Estes modelos são uma generalização dos modelos com função de base ortonormal.

$$y(k) = h_0(k) + \sum_{n=1}^{\eta} \sum_{\kappa_1=0}^k \cdots \sum_{\kappa_n=0}^k h_n(\kappa_1, \kappa_2, \dots, \kappa_n) \prod_{i=1}^n u(k - \kappa_i) \quad (3.58)$$

O problema de identificação corresponde a estimar o conjunto de *kernels* do modelo $h_i(\kappa_1, \kappa_2, \dots, \kappa_i)$ com $i = 1, \dots, \eta$.

Uma dificuldade que surge na identificação de sistemas dinâmicos utilizando a série de Volterra, é a necessidade de estimar um elevado número de parâmetros, mesmo para sistemas com não linearidades suaves. Este problema é agravado quando o sistema a ser identificado possui dinâmicas (e conseqüentemente a resposta ao impulso) muito lentas, a exemplo de grande parte dos processos estudados na Engenharia Química. A necessidade desta grande quantidade de parâmetros é conseqüência dos modelos de Volterra descreverem a saída do sistema como função apenas da entrada (modelos com estrutura NFIR¹³)

CAMPELLO (2002) discute detalhadamente os modelos baseados na série de Volterra e os problemas relacionados com a explosão de parâmetros associados aos modelos com estrutura NFIR. Ainda neste trabalho, o autor apresenta alguns resultados relacionados com o desenvolvimento¹⁴ dos kernels de Volterra através de funções ortonormais, em especial as funções de Laguerre.

3.4.4 Modelos Nebulosos (*Fuzzy*)

O conceito de modelagem nebulosa pode ser definido como a representação de sistemas baseado nas técnicas da lógica nebulosa, cujo conceito foi desenvolvido por ZADEH (1965). Os sistemas nebulosos são caracterizados por sua capacidade de determinar relações de entrada-saída sem a necessidade de um modelo matemático explícito que descreva como as saídas dependem das entradas.

Os modelos nebulosos representam um sistema através de um conjunto de regras do tipo:

$$\text{Se } x \text{ é } A \quad \text{Então } y \text{ é } B \quad (3.59)$$

¹³Nonlinear Finite Impulse Response.

¹⁴Esse desenvolvimento resulta em um modelo chamado de Wiener-Volterra.

3.4 Representações Matemáticas Não Lineares de Tempo Discreto

onde a proposição “ x é A ” é o *antecedente* e a proposição “ y é B ” é o *conseqüente* da regra nebulosa.

O conceito de conjuntos nebulosos pode ser visto como uma extensão da noção clássica de conjuntos. Em um conjunto clássico, um objeto pertence ou não pertence a este conjunto. Em um conjunto nebuloso, por outro lado, cada objeto tem um grau de pertinência ao conjunto. Esta comparação é ilustrada na figura 3.1 através de um exemplo bem simples sobre a determinação da temperatura de um líquido. Se considerarmos que esta temperatura pode variar entre 20 e 40°C, duas interpretações são possíveis: Na abordagem clássica, temperaturas abaixo de 30°C seriam consideradas frias e valores acima deste limite seriam consideradas quentes. Na abordagem nebulosa, a transição entre o quente e frio (ou qualquer outra propriedade mensurável) pode ser feita de modo mais suave e classificada de acordo com uma escala de pertinência, onde os valores “quente” e “frio” correspondem às situações limite superior e inferior, com os respectivos graus de pertinência assumido os valores 1 e 0.

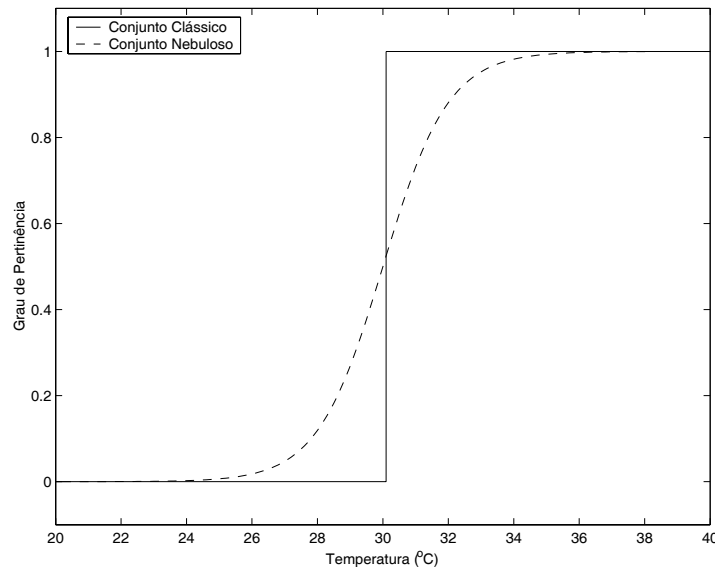


Figura 3.1: Exemplo de conjuntos clássico e nebuloso para expressar as temperaturas “quente” e “fria”.

Os conjuntos nebulosos também podem ser definidos de forma mais elaborada, a partir das operações e dos operadores nebulosos. A seguir, serão apresentadas as principais operações entre conjuntos nebulosos.

3.4.4.1 Operações entre Conjuntos Nebulosos no mesmo Universo de Discurso

Seja $X \in \mathfrak{R}^n$ uma coleção de objetos, também chamada de *Universo de Discurso*. Um conjunto nebuloso A definido em X é caracterizado através de uma função de pertinência $\mu_A : X \rightarrow [0, 1]$, com $\mu_A(x)$ representando o grau de pertinência de $x \in X$ no conjunto nebuloso A . A função de pertinência determina o significado matemático da expressão “ x é A ”, para cada valor de $x \in X$. Um conjunto nebuloso pode ser visto como uma generalização do conceito de conjunto clássico, cuja função de pertinência assume somente os valores 0 e 1 (WANG, 1994).

A figura 3.2 ilustra as funções de pertinência de três conjuntos nebulosos para representar a temperatura de um reator, chamadas “baixa”, “média” e “alta”. Neste caso, o universo de discurso são todas as possíveis temperaturas de um reator, ou seja, $X = [0, T_{max}]$, onde T_{max} é a temperatura máxima do reator. Na temperatura de 40°C , por exemplo, o conjunto nebuloso “baixa” tem um valor de pertinência de 0.5, ou seja, $\mu_{baixa}(40) = 0.5$; o conjunto nebuloso “média” um valor de pertinência de 0.5, $\mu_{media}(40) = 0.5$, e o conjunto nebuloso “alta” tem seu valor de pertinência igual a 0, $\mu_{alta}(40) = 0$.

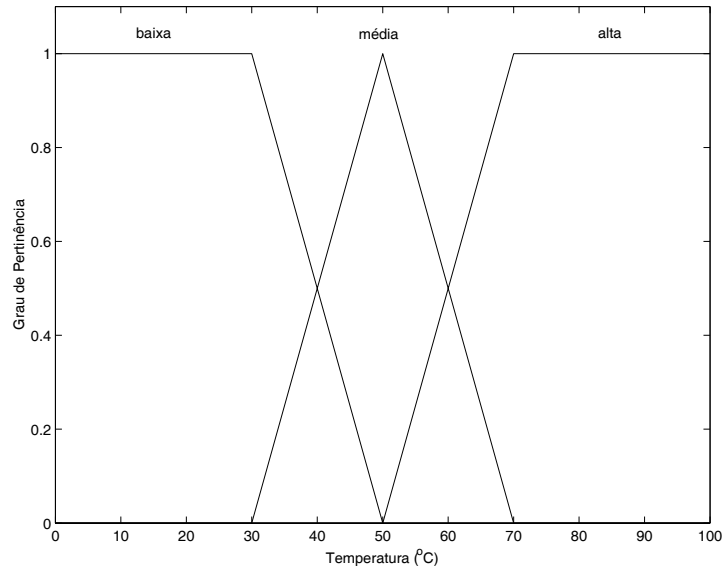


Figura 3.2: Funções de pertinência de três conjuntos nebulosos para a temperatura de um reator.

3.4 Representações Matemáticas Não Lineares de Tempo Discreto

Assim como na teoria de conjuntos tradicional, pode-se definir operações e operadores específicos para os conjuntos nebulosos. Alguns dos operadores mais utilizados são descritos a seguir:

Sejam A e B dois conjuntos nebulosos definidos no universo de discurso X com funções de pertinência dadas por $\mu_A(x)$ e $\mu_B(x)$, respectivamente.

- **União:** A função de pertinência $\mu_{A \cup B}(x)$ da união $A \cup B$ é definida para todo $x \in X$ por:

$$\mu_{A \cup B}(x) = \max \{ \mu_A(x), \mu_B(x) \} \quad (3.60)$$

- **Intersecção:** A função de pertinência $\mu_{A \cap B}(x)$ da intersecção $A \cap B$ é definida para todo $x \in X$ por:

$$\mu_{A \cap B}(x) = \min \{ \mu_A(x), \mu_B(x) \} \quad (3.61)$$

- **Complemento:** A função de pertinência $\mu_{A'}(x)$ do complemento de um conjunto nebuloso A é definida para todo $x \in X$ da seguinte forma:

$$\mu_{A'}(x) = 1 - \mu_A(x) \quad (3.62)$$

3.4.4.2 Operações entre Conjuntos Nebulosos em Universos de Discurso Distintos

As operações entre conjuntos pertencentes a universos de discurso diferentes possibilitam a construção da base de conhecimento de um sistema, ou seja, o mapeamento entre as entradas e as saídas. Em sistemas de modelagem e controle de processos, estes mapeamentos são o foco principal pois ocorrem entre os conjuntos da variável de entrada $A(x) \in X$ e os conjuntos da variável de saída $B(y) \in Y$, através da *expressão condicional de inferência* (SHAW e SIMÕES, 1999):

$$A \Rightarrow B \quad (3.63)$$

ou

$$\text{Se } A(x) \text{ Então } B(y) \quad (3.64)$$

cujo significado é a ligação do conjunto A , chamado de *antecedente* ou condição (e definido por sua função de pertinência $\mu_A(x)$, $x \in X$), ao conjunto B , chamado de *conseqüente*, resultado ou ação e definido por sua função de pertinência $\mu_B(y)$, $y \in Y$.

Os principais operadores para conjuntos nebulosos definidos em universos de discurso diferentes são apresentados a seguir:

- **Produto Cartesiano:** Sejam $A(x)$ e $B(y)$ conjuntos discretos. Juntando-se x , um dos elementos de A e y , um dos elementos de B , de modo a se obter o *par ordenado* (x, y) , define-se o seu *produto cartesiano* da seguinte forma:

$$P(x, y) = A(x) \times B(y) \quad (3.65)$$

onde o símbolo “ \times ” representa o operador de produto cartesiano. Como os conjuntos $A(x)$ e $B(y)$ são caracterizados por suas respectivas funções de pertinência, o seu produto cartesiano será uma matriz de números discretos, onde o produto algébrico pode ser empregado (SHAW e SIMÕES, 1999):

$$P = A \times B = \sum_{X \times Y} \mu_A(x) \mu_B(y) \quad (3.66)$$

Se A_1, A_2, \dots, A_n forem conjuntos discretos nos espaços X_1, X_2, \dots, X_n , respectivamente, então o produto cartesiano de A_1, A_2, \dots, A_n é um conjunto discreto no espaço $X_1 \times X_2 \times \dots \times X_n$ com funções de pertinência definidas pelo operador *min*:

3.4 Representações Matemáticas Não Lineares de Tempo Discreto

$$\mu_{A_1 \times A_2 \times \dots \times A_n}(x_1, x_2, \dots, x_n) = \min[\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n)] \quad (3.67)$$

ou pelo produto algébrico:

$$\mu_{A_1 \times A_2 \times \dots \times A_n}(x_1, x_2, \dots, x_n) = \mu_{A_1}(x_1) \cdot \mu_{A_2}(x_2) \cdot \dots \cdot \mu_{A_n}(x_n) \quad (3.68)$$

Assim, o produto cartesiano pode ser entendido como uma intersecção entre conjuntos pertencentes a universos de discursos diferentes.

- **Relação Nebulosa:** Pode-se definir uma relação nebulosa a partir de um conjunto de pares ordenados representando, por exemplo, uma tabela de valores de entradas e de saídas, $x(k), y(k)$ ($k = 1, 2, \dots, N$), de um sistema discreto com m entradas e p saídas. As m entradas, $x_i(k)$ ($i = 1, 2, \dots, m$), estão no espaço X e as p saídas, $y_j(k)$ ($j = 1, 2, \dots, p$), estão no espaço Y , onde k representa todos os possíveis valores que ocorrem nos respectivos espaços. Se $k < N$, então tal subconjunto, $\mathbf{R} \subset X \times Y$, é chamado de *relação* e pode ser representada da seguinte forma:

$$\mathbf{R} : (x, y) \in X \times Y \longrightarrow \mu_{\mathbf{R}}(x, y) \in [0, 1] \quad (3.69)$$

Em uma relação, a ligação de cada elemento de saída $y \in Y$ com um determinado elemento de entrada $x \in X$ corresponde a uma *regra*. O conjunto de ligações é uma tabela que determina uma correlação fixa entre entrada e saída e satisfaz os seguintes requisitos (SHAW e SIMÕES, 1999):

$$\mathbf{R}\{x_i(k), y_j(k)\} \subset X \times Y \quad (3.70)$$

que, para um caso com três entradas e três saídas, pode ser ilustrada assim:

	y_1	y_2	y_3
x_1	μ_{11}	μ_{12}	μ_{13}
x_2	μ_{21}	μ_{22}	μ_{23}
x_3	μ_{31}	μ_{32}	μ_{33}

- **Regra Composicional de Inferência:** Se \mathbf{R} é uma relação nebulosa em $X \times Y$ e \bar{X} é um conjunto nebuloso em X , então a regra composicional de inferência define que o conjunto nebuloso \bar{Y} em Y , induzido por \bar{X} em \mathbf{R} , é dado por:

$$\bar{Y} = \bar{X} \bullet \mathbf{R} \quad (3.71)$$

onde o símbolo “ \bullet ” representa o *operador composicional*, que pode ser qualquer operador nebuloso como, por exemplo, o operador *máx-mín*:

Sejam

$$\bar{X} = [0.2 \quad 1.0 \quad 0.3] \quad (3.72)$$

$$\mathbf{R} = \begin{bmatrix} 0.8 & 0.9 & 0.2 \\ 0.6 & 1.0 & 0.4 \\ 0.5 & 0.8 & 1.0 \end{bmatrix} \quad (3.73)$$

então, o conjunto nebuloso \bar{Y} é induzido por \bar{X} em \mathbf{R} da seguinte forma:

$$\bar{Y} = [0.2 \quad 1.0 \quad 0.3] \bullet \begin{bmatrix} 0.8 & 0.9 & 0.2 \\ 0.6 & 1.0 & 0.4 \\ 0.5 & 0.8 & 1.0 \end{bmatrix} = [0.6 \quad 1.0 \quad 0.4] \quad (3.74)$$

Em outras palavras, a regra composicional de inferência diz que, se a relação nebulosa que representa um sistema, $\mathbf{R}(x, y)$, é conhecida, então a resposta do sistema, $\bar{Y}(y)$, pode ser determinada a partir de uma entrada conhecida $\bar{X}(x)$.

Os conjuntos nebulosos também podem ser combinados utilizando os conectivos lógicos “e”, “ou” e “não”, que permitem descrever a combinação das funções de pertinência. Considerando que $\mu_A(x)$ e $\mu_B(y)$ são funções de pertinência definidas nos universos de discurso A e B , respectivamente, as proposições “ x é A e y é B ”

3.4 Representações Matemáticas Não Lineares de Tempo Discreto

(representada por A e B) e “ x é A ou y é B ” (representada por A ou B), podem ser definidas em $X \times Y$ de várias maneiras. Assim, cada um desses combinadores de conjuntos nebulosos devem ser definidos formalmente em termos da combinação das funções de pertinência. Várias definições de combinadores têm sido propostas na literatura e algumas das mais utilizadas são as seguintes (SJÖBERG, 1995):

$$\begin{aligned}\mu_{A \text{ e } B}(x, y) &= \min(\mu_A(x), \mu_B(y)) \\ \mu_{A \text{ ou } B}(x, y) &= \max(\mu_A(x), \mu_B(y))\end{aligned}\tag{3.75}$$

cujas variações mais freqüentes para os combinadores “e” e “ou” são dadas por:

$$\begin{aligned}\mu_{A \text{ e } B}(x, y) &= \mu_A(x) \cdot \mu_B(y) \\ \mu_{A \text{ ou } B}(x, y) &= \mu_A(x) + \mu_B(y) - \mu_A(x) \cdot \mu_B(y)\end{aligned}\tag{3.76}$$

O combinador “não”, utilizado em proposições do tipo “ x não é A ”, representada por “não A ”, é definida em X por:

$$\mu_{\text{não } A}(x) = 1 - \mu_A(x)\tag{3.77}$$

Os principais tipos de sistemas nebulosos podem ser classificados em três tipos de estrutura (WANG, 1994): Sistemas Lingüísticos (seção 3.4.4.3); Sistemas Relacionais Nebulosos (seção 3.4.4.4) e Sistemas Nebulosos de Takagi-Sugeno (seção 3.4.4.5) .

3.4.4.3 Sistemas Nebulosos Lingüísticos

Este tipo de modelo é utilizado em estratégias de controle nebuloso, onde o comportamento do processo é descrito por um conjunto de regras do tipo “Se-Então”. Os sistemas nebulosos lingüísticos constituem a parte essencial da lógica nebulosa. Este sistema é uma estrutura geral na qual a informação lingüística de especialistas humanos é quantificada e os princípios da lógica nebulosa são aplicados para fazer uso sistemático desta informação.

Uma base de regras nebulosas lingüísticas é uma coleção de regras escritas na seguinte forma:

$$\begin{array}{llll}
 \text{Se } (x_1 \text{ é } A_{1,1}) & \text{e} & \dots & \text{e } (x_d \text{ é } A_{1,d}) & \text{Então } (y \text{ é } B_1) \\
 \vdots & & & \ddots & \vdots \\
 \text{Se } (x_1 \text{ é } A_{p,1}) & \text{e} & \dots & \text{e } (x_d \text{ é } A_{p,d}) & \text{Então } (y \text{ é } B_p)
 \end{array} \quad (3.78)$$

onde os conjuntos nebulosos $A_{j,i}$ são agora duplamente indexados, com i representando a entrada e j a regra.

A saída do modelo nebuloso dado pela equação (3.78), considerando a entrada $x = [x_1, \dots, x_d]^T$, é obtida através do seguinte procedimento:

1. Calcula-se a função de pertinência do conjunto nebuloso de saída B' (SJÖBERG, 1995):

$$\mu_{B'}(y) = \sum_{j=1}^p \mu_{B_j}(y) \prod_{i=1}^d \mu_{A_{j,i}}(x_i) \quad (3.79)$$

que representa uma função que mapeia os pontos $x_i \in \mathfrak{R}^d$ nos conjuntos nebulosos. Para se obter uma função no conjunto original $\mathfrak{R}^d \mapsto \mathfrak{R}$, deve-se proceder ao segundo passo do procedimento:

2. Efetua-se a *defuzificação* do conjunto nebuloso de saída (consequente) B' , ou seja, substitui-se $\mu_{B'}(y)$ por seu centro de gravidade, considerando a seguinte propriedade:

$$\sum_{j=1}^p \prod_{i=1}^d \mu_{A_{j,i}}(x_i) \equiv 1 \quad (3.80)$$

que finalmente fornece o valor numérico real (não nebuloso) da função de interesse:

$$y = \sum_{j=1}^p b_j \left(\prod_{i=1}^d \mu_{A_{j,i}}(x_i) \right) \quad (3.81)$$

onde b_j é o centro de gravidade do conjunto nebuloso B_j .

3.4 Representações Matemáticas Não Lineares de Tempo Discreto

3.4.4.4 Sistemas Nebulosos Relacionais

Estes sistemas representam uma alternativa aos sistemas puramente lingüísticos, preservando as suas características qualitativas (também podem ser interpretados através de regras do tipo “Se-Então”) sem a necessidade de desenvolvê-las (PEDRYCZ, 1996).

Nos modelos nebulosos relacionais (MNR), as variáveis de entrada e saída são relacionadas através de uma *Matriz Relacional* \mathbf{R} . Esta matriz contém todas as combinações possíveis de condições entrada-saída com um valor, entre zero e um, para cada condição. Cada valor representa o grau de verdade de cada possível relação. Um valor igual a *um* indica que a relação é a mais forte e um valor igual a *zero* indica que a relação é a mais fraca.

Uma relação nebulosa \mathbf{R} é escrita como um conjunto de regras nebulosas com conjuntos nebulosos definidos em cada universo de discurso. Uma equação nebulosa relacional para um sistema nebuloso com múltiplas entradas e uma saída é dada por:

$$\bar{Y} = \bar{X}_1 \bullet \bar{X}_2 \bullet \dots \bullet \bar{X}_n \bullet \mathbf{R} \quad (3.82)$$

onde \bar{Y} é o conjunto nebuloso de saída, $[\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n]$ denotam os conjuntos nebulosos de entrada, \mathbf{R} é a matriz relacional e o símbolo “ \bullet ” representa um operador de composição nebuloso.

Cada elemento da matriz relacional \mathbf{R} é denotado por:

$$\mathbf{R}(i, j) = p_{ij} \quad (3.83)$$

que, por sua vez, podem ser representados por uma regra lingüística do tipo:

$$\text{Se } \bar{X}_i \text{ Então } \bar{Y}_j \text{ com possibilidade } p_{ij} \quad (3.84)$$

O valor *defuzificado* da saída do modelo pode ser obtido de forma análoga ao procedimento descrito na seção 3.4.4.3.

3.4.4.5 Sistemas Nebulosos de Takagi-Sugeno

Estes sistemas apresentam uma estrutura híbrida, ou seja, os dados são representados por conjuntos nebulosos (antecedente ou parte “Se”) e as relações de entrada-saída são expressas através de uma combinação de funções das variáveis de entrada (consequente ou parte “Então”). Assim, a equação (3.78) pode ser reescrita da seguinte forma:

$$\begin{array}{llll} \text{Se } (x_1 \text{ é } A_{1,1}) \text{ e } \dots & \text{e } (x_d \text{ é } A_{1,d}) & \text{Então } y = f_1(x_1, \dots, x_d) \\ \vdots & \ddots & \vdots & \\ \text{Se } (x_1 \text{ é } A_{p,1}) \text{ e } \dots & \text{e } (x_d \text{ é } A_{p,d}) & \text{Então } y = f_p(x_1, \dots, x_d) \end{array} \quad (3.85)$$

Geralmente, os termos consequentes são representados por funções lineares do tipo:

$$f_j(x_1, \dots, x_d) = w_{j,0} + w_{j,1}x_1 + \dots + w_{j,d}x_d, \quad \text{com } j = 1, 2, \dots, p \quad (3.86)$$

A saída do modelo nebuloso de TAKAGI e SUGENO (1985) é definida da seguinte forma:

$$y(x, \theta) = \sum_{j=1}^p (w_{j,0} + w_{j,1}x_1 + \dots + w_{j,d}x_d) \Phi(x; \theta_j) \quad (3.87)$$

onde

$$\Phi(x; \theta_j) = \frac{\mu_j(x; \theta_j)}{\sum_{i=1}^p \mu_i(x; \theta_i)} \quad (3.88)$$

são os coeficientes da combinação convexa dos modelos lineares locais e $\mu_j(x; \theta_j)$ são as funções de pertinência da entrada definidas no espaço das entradas. O vetor θ_j contém os parâmetros da função de pertinência de entrada da j -ésima regra.

O modelo de TAKAGI e SUGENO (1985), dado pela equação (3.87), pode ser interpretado como uma interpolação entre os modelos lineares locais $f_j(x) = w_{j,0} + w_{j,1}x_1 + \dots + w_{j,d}x_d$, com a função $\Phi(x; \theta_j)$ determinando, para cada valor de entrada, o grau de ponderação da j -ésima regra na interpolação (CAVALCANTE JÚNIOR, 2000).

3.4.5 Modelos com Estrutura Hierárquica

Por se tratarem de aproximadores universais¹⁵, os modelos neurais do tipo MLP com arquitetura *feedforward* e os modelos nebulosos são duas das classes de modelos não lineares mais utilizadas na atualidade. Contudo, ambos os modelos apresentam o inconveniente de necessitarem da estimação de um grande número de parâmetros. Este problema fica ainda mais grave quando se trata da identificação de sistemas de grande porte, pois o número de parâmetros a serem estimados cresce exponencialmente com o número de entradas do modelo. Alguns autores (KOSKO, 1997; HAYKIN, 1999) referem-se a este problema como a “Maldição da Dimensionalidade” (*Curse of Dimensionality*).

A utilização de estruturas hierárquicas surgiu como uma alternativa para contornar este problema. A idéia central desta abordagem é substituir um único sistema, nebuloso ou neural, por um conjunto de sub-sistemas (série hierárquica) conectados em cascata. A estrutura resultante exibe a importante propriedade de apresentar uma dependência linear, e não mais exponencial, entre a dimensão do espaço de entrada e o número de parâmetros livres de projeto (CAMPELLO e AMARAL, 1999b; MELEIRO *et al.*, 2001; CAMPELLO, 2002; CAMPELLO *et al.*, 2002; MELEIRO *et al.*, 2002a). A estrutura convencional típica de modelos nebulosos (MN) ou neurais (RN) está representada na figura 3.3-a, enquanto a figura 3.3-b ilustra a estrutura de sub-modelos hierárquicos ligados em cascata.

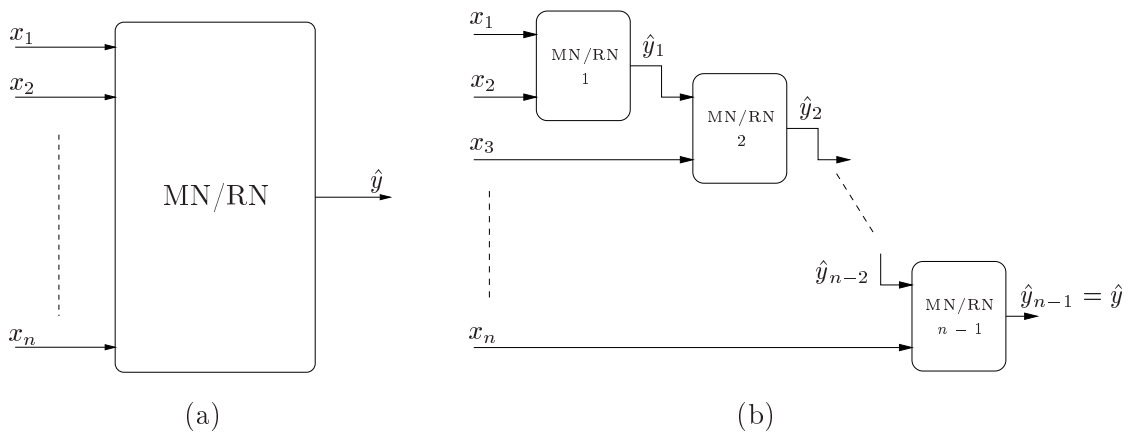


Figura 3.3: (a) Modelo Convencional; (b) Modelo Hierárquico.

¹⁵Modelos com esta característica são capazes de aproximar com precisão arbitrária qualquer mapeamento contínuo definido sobre um domínio compacto (HAYKIN, 1999).

Segundo CAMPELLO (2002), a aplicação da estrutura hierárquica em sistemas nebulosos foi inicialmente proposta por RAJU *et al.* (1991), proporcionando uma redução significativa do número de parâmetros e, por conseguinte, permitindo o seu emprego no projeto de controladores *fuzzy* em sistemas de grande porte. Diferentes modelos foram implementados com sucesso utilizando a estrutura hierárquica. WANG (1998; 1999), por exemplo, implementou sub-sistemas hierárquicos utilizando tipos especiais de modelos nebulosos de Takagi-Sugeno, enquanto CAMPELLO e AMARAL (1999b; 2000b) utilizaram modelos relacionais nebulosos com o mesmo objetivo. Um resultado importante obtido por estes pesquisadores é que os modelos resultantes são aproximadores universais. CAMPELLO (2002) e CAMPELLO e AMARAL (2002) demonstraram ainda que seu modelo permite a extração de regras interpretáveis linguisticamente, capazes de descrever suas relações de entrada-saída.

Embora os modelos hierárquicos sejam aproximadores universais, a sua aplicação com sucesso em problemas reais de grande porte está intimamente relacionada com a disponibilidade de métodos numéricos eficientes para estimar os parâmetros do modelo a partir de um conjunto finito de dados de entrada-saída. Um método de treinamento baseado na técnica da retropropagação do erro (*backpropagation*) para a estimação de modelos hierárquicos com até três entradas foi proposto por WANG (1999). CAMPELLO e AMARAL (1999b) propuseram uma abordagem similar, porém com formulação genérica e, portanto, adequada para qualquer número de entradas (CAMPELLO e AMARAL, 2000b; MELEIRO *et al.*, 2001; CAMPELLO, 2002; MELEIRO *et al.*, 2002a).

A aplicação de modelos não lineares com estrutura hierárquica na identificação e controle de processos não lineares de grande porte é uma das propostas centrais desta tese. A seguir, será apresentada a formulação matemática básica do modelo hierárquico neuro-nebuloso (HNN) utilizado neste trabalho. A determinação da estrutura do modelo HNN, os procedimentos de estimação e validação e os principais resultados de sua aplicação na identificação de um processo fermentativo em escala industrial são descritos nas seções 6.3.4, 6.3.5 e 6.3.6, respectivamente. Os pormenores relacionados com o projeto e os resultados do desempenho de um controlador preditivo não linear baseado no modelo HNN são discutidos na seção 6.4.

3.4 Representações Matemáticas Não Lineares de Tempo Discreto

3.4.5.1 Estrutura Relacional Nebulosa Simplificada

O sistema nebuloso utilizado nos modelos hierárquicos HNN considerados nesta tese é um tipo especial de sistema *fuzzy* chamado de *Estrutura Relacional Simplificada*, proposta por OLIVEIRA e LEMOS (1997). Os modelos relacionais nebulosos (seção 3.4.4.4), ao contrário dos modelos nebulosos clássicos baseados em regra (modelos lingüísticos e de Takagi-Sugeno), permitem que a estimação numérica de seus parâmetros e a extração do conhecimento lingüístico sejam realizadas simultaneamente a partir dos parâmetros obtidos (CAMPELLO e AMARAL, 2000a; CAMPELLO e AMARAL, 2001; CAMPELLO, 2002).

A formulação desta estrutura pode ser desenvolvida a partir da especificação de um sistema MISO¹⁶ genérico dado por $y = \mathcal{F}(x_1, \dots, x_n)$, onde \mathcal{F} é um operador não linear que mapeia as entradas x_i ($i = 1, \dots, n$) na saída y . A modelagem deste sistema através da estrutura relacional nebulosa é dada por:

$$\hat{y} = \Psi^T \Omega \quad (3.89)$$

onde \hat{y} é a saída do modelo, Ω ($m \times 1$) é o vetor de parâmetros e Ψ ($m \times 1$) é o vetor de entrada nebuloso. O vetor Ψ é dado pelo Produto de Kronecker (\otimes) das entradas nebulosas individuais, ou seja:

$$\Psi = X_1 \otimes X_2 \otimes \dots \otimes X_n \quad (3.90)$$

e as entradas X_i ($i = 1, \dots, n$) são formadas a partir das entradas não *fuzzificadas* (*crisp*) x_i da seguinte forma:

$$X_i = [\mathcal{X}_{i_1}(x_i) \quad \mathcal{X}_{i_2}(x_i) \quad \dots \quad \mathcal{X}_{i_{c_i}}(x_i)]^T \quad (3.91)$$

onde $\mathcal{X}_{i_j}(\cdot)$ é o j -ésimo conjunto nebuloso da i -ésima variável de entrada (com c_i conjuntos nebulosos).

¹⁶ Multi-Input Single-Output.

CAMPELLO e AMARAL (1999a) demonstraram que este sistema nebuloso, utilizando conjuntos nebulosos Gaussianos, é completamente equivalente a uma rede neural do tipo RBF com funções de ativação Gaussianas. Contudo, a sua formulação é mais simples, pois no sistema relacional nebuloso, cada variável de entrada é tratada separadamente (CAMPELLO e AMARAL, 2000b; MELEIRO *et al.*, 2001; MELEIRO *et al.*, 2002a; CAMPELLO, 2002; CAMPELLO *et al.*, 2002).

O modelo dado pelas equações (3.89), (3.90) e (3.91) apresenta a estrutura convencional dos modelos nebulosos e neurais, representados esquematicamente na figura 3.3-a. Esta estrutura possui o problema mencionado anteriormente (maldição da dimensionalidade), dada a relação exponencial entre o número de entradas do modelo e o número de parâmetros que devem ser estimados. A fim de ilustrar este problema, considere por simplicidade, e sem perda de generalidade, que o número de conjuntos nebulosos na equação (3.91) seja $c_i = c$ para $i = 1, \dots, n$. A partir da equação (3.90), deduz-se que o número de elementos dos vetores de entradas nebulosas, Ψ , e de parâmetros, Ω , é dado por (CAMPELLO, 2002):

$$m = c^n \quad (3.92)$$

Este número, m , também corresponde ao número de regras nebulosas associadas ao modelo ou, alternativamente, ao número de neurônios da rede neural RBF correspondente. Este também é o número de parâmetros a serem estimados se os conjuntos nebulosos (ou as funções de ativação) forem mantidos constantes. Se, por outro lado, os centros e as aberturas dos conjuntos nebulosos (ou funções de ativação) puderem variar, o número de parâmetros a serem estimados é dado por:

$$p = c^n + 2nc \quad (3.93)$$

Deve-se destacar que a capacidade de aproximação do modelo está diretamente relacionada com o número de conjuntos nebulosos, c . A dependência exponencial entre o número de entradas do modelo, n , e o número de regras nebulosas (ou neurônios), m , fica clara através da equação (3.92).

3.4 Representações Matemáticas Não Lineares de Tempo Discreto

3.4.5.2 O Modelo Hierárquico Neuro-Nebuloso (HNN)

A estrutura hierárquica neuro-nebulosa utilizada nesta tese foi proposta por CAMPELLO e AMARAL (1999b; 2000b), que implementaram o modelo relacional nebuloso na forma de sub-sistemas com espaço de entrada bi-dimensional e conectados em uma arquitetura em cascata, conforme mostrado na figura 3.3-b. Esta estrutura se apresenta como uma alternativa viável para o problema da explosão de parâmetros apresentada pelos modelos com estrutura convencional (ou não hierárquica), ilustrada na figura 3.3-a.

Como pode ser visto na figura 3.3-b, a estrutura hierárquica resultante é composta por $n - 1$ sub-modelos com duas entradas cada, onde cada sub-modelo contém c^2 regras nebulosas (ou neurônios da rede RBF equivalente). Desta forma, o número total de regras nebulosas/neurônios do modelo HNN é dado por:

$$m = (n - 1)c^2, \quad (n \geq 2) \quad (3.94)$$

De forma análoga ao modelo não hierárquico, cujo número de parâmetros é dado pela equação (3.92), a equação (3.94) também fornece o número de parâmetros a serem estimados se os conjuntos nebulosos (ou funções de ativação) do modelo hierárquico permanecerem constantes. Caso contrário, o número de parâmetros é dado por:

$$p = (n - 1)c^2 + 2nc + 2(n - 2)c = (n - 1)c^2 + 4(n - 1)c \quad (3.95)$$

Como pode ser verificado através da equação (3.94), o modelo HNN apresenta a grande vantagem de apresentar uma relação linear, e não mais exponencial, entre o número de entradas do modelo e o número de parâmetros a serem estimados.

A tabela 3.1 ilustra a relação entre a dimensão da entrada e o número de parâmetros a serem ajustados pelos modelos convencional (MN/RN) e HNN, considerando um número típico de conjuntos nebulosos, $c = 5$.

Tabela 3.1: Número de parâmetros, m , dos modelos convencional e HNN em função do número de entradas, n , para $c = 5$.

	Modelo Convencional		Modelo HNN
n	m		m
2	25		25
3	125		50
4	625		75
5	3125		100
6	15625		125
7	78125		150

3.4.5.3 Formulação Matemática do Modelo HNN

O modelo HNN utilizado nesta seção (descrito pela equação (3.89) e ilustrado na figura 3.3), é representado pelas seguintes equações (CAMPELLO e AMARAL, 1999b; CAMPELLO e AMARAL, 2000a; CAMPELLO e AMARAL, 2000b; CAMPELLO e AMARAL, 2001; MELEIRO *et al.*, 2001; CAMPELLO, 2002):

$$\hat{y}_i = \Psi_i^T \Omega_i = \sum_{l=1}^{c^2} \Psi_{i_l} \Omega_{i_l}, \quad i = 1, \dots, n-1 \quad (3.96)$$

$$\Psi_i = [\Psi_{i_1} \quad \dots \quad \Psi_{i_{c^2}}]^T = \begin{cases} X_{i+1} \otimes \hat{Y}_{i-1}, & i = 2, \dots, n-1 \\ X_{i+1} \otimes X_i, & i = 1 \end{cases} \quad (3.97)$$

$$X_j = [\mathcal{X}_{j_1}(x_j) \quad \mathcal{X}_{j_2}(x_j) \quad \dots \quad \mathcal{X}_{j_c}(x_j)]^T, \quad j = 1, \dots, n \quad (3.98)$$

$$\hat{Y}_h = [\mathcal{Y}_{h_1}(\hat{y}_h) \quad \mathcal{Y}_{h_2}(\hat{y}_h) \quad \dots \quad \mathcal{Y}_{h_c}(\hat{y}_h)]^T, \quad h = 1, \dots, n-2 \quad (3.99)$$

onde $\mathcal{X}_{(\cdot)}$ e $\mathcal{Y}_{(\cdot)}$ são os conjuntos nebulosos Gaussianos associados às entradas $x_{(\cdot)}$ e as saídas internas $\hat{y}_{(\cdot)}$:

$$\mathcal{X}_{j_l}(x_j) = \exp \left(-\frac{(x_j - \theta_{j_l})^2}{\sigma_{j_l}^2} \right) \quad (3.100)$$

$$\mathcal{Y}_{h_l}(\hat{y}_h) = \exp \left(-\frac{(\hat{y}_h - \phi_{h_l})^2}{\varphi_{h_l}^2} \right) \quad (3.101)$$

enquanto θ_{j_l} (ϕ_{h_l}) e σ_{j_l} (φ_{h_l}) são os centros e as aberturas do l -ésimo conjunto nebuloso associado à j -ésima entrada x_j (h -ésima saída intermediária \hat{y}_h), respectivamente.

3.4.5.4 A Estimação dos Parâmetros do Modelo HNN

A primeira etapa do processo de estimação dos parâmetros do modelo é a inicialização de seus parâmetros. O primeiro passo consiste em normalizar as variáveis de entrada e saída dentro do intervalo $[-1, 1]$ para evitar problemas numéricos durante o procedimento de otimização (geralmente associados a variáveis com diferentes ordens de grandeza). Os conjuntos nebulosos associados a cada entrada são distribuídos homogeneamente dentro do intervalo de normalização (centros igualmente espaçados e aberturas iguais à distância entre dois centros consecutivos). O vetor de parâmetros, $\Omega(\cdot)$, deve ser inicializado aleatoriamente com média zero e com valores absolutos pequenos o suficiente de modo que os valores iniciais das saídas internas do modelo HNN sejam próximos do intervalo de normalização (CAMPELLO, 2002).

O modelo HNN do sistema pode ser estimado resolvendo o seguinte problema de otimização:

$$\min_{\Gamma} J = \frac{1}{2} \sum_{k=1}^N (y(k) - \hat{y}(k))^2 \quad (3.102)$$

onde Γ representa o conjunto dos parâmetros a serem estimados, N é o conjunto de dados de entrada-saída para treinamento ($\{x_1(k), \dots, x_n(k), y(k)\}_{k=1}^N$) medidos a partir do sistema que se deseja modelar, y é a saída medida do sistema e \hat{y} é a saída estimada pelo modelo.

A técnica de otimização utilizada para resolver o problema de otimização (3.102) foi o algoritmo do gradiente conjugado de Fletcher e Reeves, adequado para problemas de grande porte pois não necessita do cálculo da matriz Hessiana ou de sua inversa. Além disso, este método assegura a convergência do procedimento de otimização com taxas de algoritmos de 2ª ordem (CAMPELLO e AMARAL, 1999b).

A aplicação desta técnica requer o cômputo do vetor gradiente da função de custo J em relação ao conjunto de parâmetros Γ que, no caso do modelo HNN, é formado pelo vetor de parâmetros $\Omega_{(\cdot)}$ (3.96) e pelos centros, $\theta_{(\cdot)}$ e $\phi_{(\cdot)}$, e aberturas, $\sigma_{(\cdot)}$ e $\varphi_{(\cdot)}$, dos conjuntos nebulosos nas equações (3.100) e (3.101). Todas as derivadas necessárias para a aplicação do algoritmo de Fletcher e Reeves estão descritas em CAMPELLO e AMARAL (2000b), MELEIRO *et al.* (2001), CAMPELLO (2002), CAMPELLO *et al.* (2002) e MELEIRO *et al.* (2002a).

3.5 Considerações Finais Sobre o Capítulo

O objetivo deste capítulo foi apresentar as definições e os conceitos fundamentais acerca da modelagem de sistemas lineares e não lineares. Foi dado destaque aos modelos mais citados na literatura e suas principais características foram discutidas e correlacionadas sempre que possível. Outros modelos utilizados com menos frequência, dentre os quais os modelos hierárquicos neuro-nebulosos, foram aqui incluídos por conta de sua importância nas estratégias de identificação e controle de processos não lineares propostas nesta tese, além de seu potencial para amenizar os problemas relacionados com o excesso de parâmetros característico de modelos caixa-preta não lineares.

Capítulo 4

IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS NÃO LINEARES UTILIZANDO MODELOS NEURAI

Preâmbulo

Muito embora aproximações lineares possam ser utilizadas satisfatoriamente em algumas aplicações práticas, os sistemas dinâmicos reais são, em última análise, não lineares. Mesmo com o inerente aumento de complexidade, a utilização de modelos não lineares muitas vezes é inevitável. As exigências relacionadas com o aumento de precisão nem sempre são a única necessidade. Na verdade, uma das razões mais fortes para a escolha de modelos não lineares está relacionada com o fato de que estes modelos podem representar certos regimes dinâmicos que os modelos lineares não são capazes de reproduzir (AGUIRRE, 2000).

Embora os tópicos deste capítulo tenham sido escritos com o objetivo de formalizar o procedimento geral de identificação de sistemas dinâmicos não lineares tendo como modelo básico as redes neurais artificiais, a aplicação deste procedimento a outros modelos não lineares (alguns deles descritos na seção 3.4) é direta.

Apresenta-se a seguir um resumo dos conceitos e da metodologia empregados nesta tese para identificar os processos não lineares tratados nos capítulos 6 e 7.

A seção 4.1 apresenta uma visão geral sobre as principais características das redes neurais artificiais e as motivações para o seu uso na modelagem de processos não lineares. O procedimento padrão de identificação de sistemas não lineares é apresentado na seção 4.2 na forma de um fluxograma, onde são destacadas suas etapas básicas. O projeto e a realização de experimentos são apresentados na seção 4.3. Neste ponto são discutidos os testes para a determinação do grau de não linearidade do sistema; a importância da escolha de um período de amostragem adequado; o problema da explosão de parâmetros apresentado por muitos modelos do tipo caixa-preta; o projeto do sinal de entrada responsável pela correta excitação do sistema e a preparação dos dados amostrados para a etapa de modelagem. A seleção da estrutura de modelos neurais é discutida na seção 4.4. Esta etapa envolve a escolha de um modelo com uma determinada estrutura e a determinação do vetor de regressão adequado. A estimação de modelos neurais e as principais características das técnicas de modelagem direta e inversa são apresentadas na seção 4.5. A etapa de validação do modelo obtido é discutida na seção 4.6, onde os testes de estimação do erro de generalização, correlação de resíduos e inspeção visual, utilizados para a avaliação do desempenho de modelos neurais, são apresentados. Finalmente, a seção 4.7 trata da decisão final do projetista, ou seja, nesta etapa decide-se se o desempenho do modelo obtido é satisfatório ou se é preciso refazer uma ou mais etapas anteriores do procedimento de estimação do modelo.

4.1 Introdução

A habilidade de “aprender” a partir de exemplos é uma das características mais importantes das Redes Neurais Artificiais (RNA) e, por conta disso, as RNA têm sido cada vez mais utilizadas como uma ferramenta para identificação de sistemas não lineares. Contudo, apesar do fato do “aprendizado a partir de exemplos” soar fácil, muitos pesquisadores têm se surpreendido com dificuldades na aplicação de redes neurais para obtenção das soluções desejadas (NØRGAARD *et al.*, 2000).

As redes neurais artificiais têm atraído muita atenção, principalmente quando comparadas a outras técnicas não lineares de identificação de processos (NG, 1997; NØRGAARD *et al.*, 2000). Parte deste sucesso se deve ao fato de que é muito difícil saber *a priori* qual técnica de identificação é a mais adequada para uma dada aplicação sem uma avaliação prévia das técnicas disponíveis. Por isso, é altamente desejável considerar uma única técnica que possa ser empregada para todas as aplicações em vez de avaliar várias técnicas candidatas para cada nova aplicação. A propriedade de ser aplicável a um grande número de situações (versatilidade) é uma das características mais atraentes das RNA pois, além de simplificar o processo de modelagem do sistema, facilita a implementação de ferramentas mais genéricas no projeto de sistemas de controle.

Quando se procura uma técnica que forneça o maior número de vantagens e que funcione bem para a maioria dos problemas de identificação, certos tipos de redes neurais artificiais despontam como uma excelente opção. Particularmente, as redes do tipo *Perceptron Multicamadas* (MLP)¹ têm apresentado um crescente aumento nas aplicações de identificação e controle de processos. De acordo com NG (1997) e NØRGAARD *et al.* (2000), publicações recentes dão conta de que na última década numerosas aplicações práticas evidenciam o fato de que as redes do tipo MLP possuem uma impressionante capacidade, tanto para modelagem de processos não lineares, quanto para aplicação em diversas estratégias de controle de processos. Recentemente surgiram alguns resultados teóricos que tentam explicar as razões deste sucesso e, sem dúvida, um dos resultados mais significativos e importantes foi a demonstração da capacidade de aproximação universal das MLPs (CYBENKO (1989), HORNIK *et al.* (1989), HORNIK *et al.* (1990), BARRON (1993) e JUDITSKY *et al.* (1995)).

¹ *Multi Layer Perceptron.*

Apesar de haver um aumento de complexidade quando comparado com a identificação de sistemas lineares, muitos dos resultados estabelecidos na teoria convencional de identificação de sistemas também se aplicam à identificação de sistemas não lineares. Assim, a identificação de sistemas baseada em redes neurais artificiais pode ser encarada, sob muitos aspectos, como uma extensão natural da abordagem convencional de identificação de sistemas lineares (NØRGAARD *et al.*, 2000).

4.2 O Procedimento Básico da Modelagem Neural

Vários autores (NORTON, 1986; SÖDERSTRÖM e STOICA, 1989; MILLS *et al.*, 1996; LJUNG, 1999; AGUIRRE, 2000; NØRGAARD *et al.*, 2000) destacam algumas etapas básicas necessárias para identificar sistemas dinâmicos não lineares. De uma maneira geral, estas etapas podem ser resumidas da seguinte forma:

1. Realização de Experimentos para Extração de Dados
2. Seleção da Estrutura do Modelo
3. Estimação do Modelo
4. Validação do Modelo

Contudo, deve-se ter em mente que conhecimentos sobre a dinâmica do sistema e/ou informações sobre o uso que se pretende dar ao modelo podem influenciar todos os estágios do procedimento.

Este procedimento, cujas etapas serão discutidas nas seções subseqüentes, pode ser representado na forma de um fluxograma, como mostrado na figura 4.1.

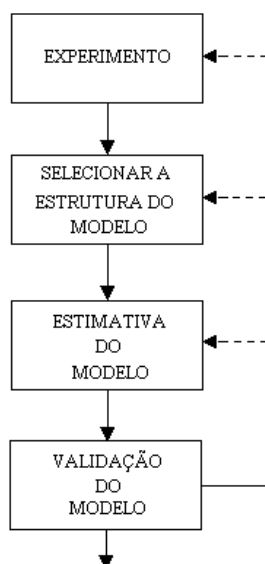


Figura 4.1: Procedimento básico para identificação de sistemas.

4.3 Realização de Experimentos

Este é o primeiro estágio do processo de identificação de modelos neurais e o principal objetivo é projetar e realizar um conjunto de experimentos que descreva como o sistema responde quando submetido a diferentes entradas (ações de controle ou perturbações), assegurando que toda a sua faixa de operação seja contemplada. A importância desta etapa está diretamente associada com a acurácia do modelo que será obtido, uma vez que o conjunto de dados de entrada-saída gerado no experimento será utilizado no processo de treinamento das redes neurais².

A idéia básica é variar a(s) entrada(s) do sistema, u , de modo a cobrir toda a sua faixa de operação e observar o impacto dessas entradas sobre a(s) sua(s) saída(s), y . Deve-se destacar que o efeito das perturbações pode influenciar fortemente o comportamento do sistema e, tratando-se de perturbações mensuráveis, o seu registro é de fundamental importância na etapa de estimação do modelo do processo.

Considerando que os dados experimentais que descrevem o comportamento do processo em toda a sua faixa de operação tenham sido obtidos com uma frequência de amostragem adequada, então $\{u(k)\}$ não é mais apenas um conjunto de entradas mas um *signal*, o *signal de controle*. Da mesma forma, $\{y(k)\}$ representa agora o *signal da saída medida*, enquanto k especifica o instante de amostragem. O conjunto de dados de entrada-saída correspondente $Z^N = \{[u(k), y(k)], k = 1, \dots, N\}$ será utilizado mais tarde na determinação de um modelo do sistema. Se o sistema a ser identificado for instável ou apresentar dinâmicas levemente amortecidas, pode ser necessário conduzir os experimentos em malha fechada³.

Deve-se, contudo, estar atento ao fato de que o projeto de experimentos é mais complexo do que simplesmente gerar dados de entrada-saída para o treinamento das redes neurais. Tópicos de grande importância dentro deste contexto são: *i*) Testes para decidir se a abordagem de identificação não linear é relevante; *ii*) Projeto de sinais de entrada que levem a um conjunto de dados com informações adequadas sobre

²De um modo geral, isto se aplica a todo tipo de modelagem caixa-preta não linear.

³A identificação em malha fechada pode também ser parte intrínseca da estratégia de controle a ser implementada. CAJUEIRO (2000) empregou esta abordagem utilizando redes neurais artificiais para identificar a dinâmica não modelada do sistema.

4.3 Realização de Experimentos

a dinâmica do sistema; e *iii*) Técnicas de preparação dos dados para o treinamento das redes neurais. Estes e outros aspectos sobre a realização de experimentos serão contemplados a seguir.

4.3.1 Determinação do Grau de Não Linearidade do Sistema

Antes de decidir sobre a utilização de modelos neurais (ou outros tipos de modelos não lineares) para realizar o processo de identificação de um sistema, deve-se realizar uma série de testes para verificar a sua não linearidade e avaliar a relevância da utilização de tais modelos. Esta etapa é fortemente recomendada quando há pouco ou nenhum conhecimento sobre o comportamento dinâmico do sistema a ser identificado. TONG (1990) descreve vários testes para verificar a não linearidade de um sistema enquanto NØRGAARD *et al.* (2000) indicam dois dos mais simples e utilizados:

1. Teste da Superposição: Duas características marcantes dos sistemas não lineares são que as propriedades de superposição (eq. 4.1) e homogeneidade (eq. 4.2) não são satisfeitas:

$$y(t) = g[\varphi_1(t) + \varphi_2(t)] = g[\varphi_1(t)] + g[\varphi_2(t)] \quad (4.1)$$

$$y(t) = g[\alpha\varphi(t)] = \alpha g[\varphi(t)] \quad (4.2)$$

2. Teste da Resposta em Frequência: Uma propriedade bem estabelecida para sistemas lineares é que sua resposta em frequência é única, independente da amplitude do sinal de entrada. Um teste de linearidade relativamente simples é submeter o sistema a diferentes entradas senoidais com frequências e amplitudes distintas. Se o sistema for linear, sua saída estacionária deve ser uma senóide de mesma frequência e amplitude proporcional ao sinal de entrada.

Caso os testes de não linearidade e/ou algum conhecimento prévio sobre o sistema tenham conduzido à abordagem não linear, deve-se estar atento para alguns pormenores relacionados à aquisição dos dados que serão destinados à estimação do modelo. Os pontos de maior destaque são brevemente comentados a seguir.

4.3.2 Escolha do Período de Amostragem

Se a frequência de amostragem escolhida for muito alta em relação à dinâmica do sistema, podem ocorrer problemas de mal condicionamento numérico no processo de identificação do modelo. Por outro lado, períodos de amostragem muito longos podem implicar na perda de dinâmicas mais rápidas (MIDDLETON e GOODWIN, 1990). Caso a identificação seja realizada para fins de projeto de sistemas de controle, a escolha da frequência de amostragem torna-se ainda mais delicada pois, neste caso, esta deve ser escolhida de acordo com a dinâmica do sistema como um todo (planta e controlador). Uma alta frequência de amostragem permite mais rapidez de resposta para problemas servo, possibilitando ações de controle mais suaves. Por outro lado, os problemas de mal condicionamento numérico tendem a ser mais pronunciados, o que indica que a escolha do período de amostragem deve atender um compromisso entre beneficiar a identificação e o projeto do controlador (NØRGAARD *et al.*, 2000). MELEIRO *et al.* (2001) discutiram alguns aspectos relacionados com a determinação de um período de amostragem adequado quando modelos não lineares são utilizados para identificar processos de grande porte não lineares, ressaltando a importância de estudos preliminares acerca do comportamento dinâmico do processo frente a diferentes tipos de entradas e perturbações.

4.3.3 A “Maldição da Dimensionalidade”

Uma vez que os sistemas não lineares não apresentam as propriedades de superposição e homogeneidade, existe a necessidade de excitar o sistema adequadamente através de um sinal de entrada rico o suficiente em frequências e amplitudes. Isto significa que o sistema deve ser excitado com todas as combinações de frequências e amplitudes passíveis de ocorrer dentro de sua faixa operacional. A geração de uma base de dados como esta implica, invariavelmente, em um número muito maior de dados necessários para identificar um sistema não linear quando comparado aos sistemas lineares. Como consequência, o número de parâmetros e o tamanho do conjunto de dados necessários para identificar um sistema não linear com uma dada acurácia cresce exponencialmente com a dimensão do espaço de entradas do mapeamento a ser aproximado. Este é um problema bem conhecido e chamado de *Maldição da Dimensionalidade*

4.3 Realização de Experimentos

(KOSKO, 1997; HAYKIN, 1999). Este problema configura um ponto fraco da abordagem caixa-preta não linear, cuja gravidade fica pronunciada quando se trata de sistemas de grande porte. Contudo, alguns resultados relativamente recentes envolvendo estruturas hierárquicas vem sendo reportadas na literatura especializada como uma solução viável para este tipo de problema (RAJU *et al.*, 1991; WANG, 1998; CHEN e WANG, 2000). CAMPELLO e AMARAL (1999b; 2000b; 2002) apresentam resultados bastante significativos nesta linha de pesquisa e alguns destes resultados são utilizados no presente trabalho como uma das alternativas para a identificação de sistemas não lineares de grande porte.

4.3.4 Projeto do Sinal de Entrada

A seleção de uma seqüência adequada de sinais de entrada está intimamente relacionada com a correta verificação da faixa de operação do processo. Além disso, deve-se evitar excitar dinâmicas indesejadas, o que na prática significa que o sinal de entrada deve ser mantido em freqüências suficientemente baixas. Contudo, quando se trabalha com sistemas não lineares, é fundamental que todas as amplitudes e freqüências estejam representadas. Portanto, é importante aplicar uma seqüência de sinais de entrada que realmente excitem o sistema de modo que toda a sua faixa operacional seja coberta, garantindo assim um conjunto de dados bem amplo. Todavia, deve-se evitar ao máximo coletar informações redundantes.

Alguns autores propuseram metodologias capazes de gerar sinais que atendessem a esta demanda, dentre eles destacam-se os trabalhos de SÖDERSTRÖM e STOICA (1989) e mais recentemente NØRGAARD *et al.* (2000). Duas das estratégias mais utilizadas relacionadas com esta metodologia são apresentadas a seguir.

I) Uma seqüência de sinais de entrada com mudança de nível de ativação (amplitude) variando em intervalos (freqüência) constantes pode ser gerada da seguinte forma:

Seja $e(t)$ um sinal correspondente a um ruído branco com variância σ_e^2 . O sinal de entrada definido de acordo com a equação (4.3) mudará para um novo nível de ativação a cada novo instante de amostragem N .

$$u(t) = e \left(\text{int} \left[\frac{t-1}{N} \right] + 1 \right), \quad t = 1, 2, \dots \quad (4.3)$$

onde *int* significa a parte inteira da divisão. A função de correlação do sinal de entrada é dada por:

$$R_u(\tau) = \frac{N-\tau}{N} \sigma_e^2 \quad (4.4)$$

correspondendo à seguinte densidade espectral

$$\phi(\omega) = \frac{\sigma_e^2}{2\pi N} \frac{1 - \cos(N\omega)}{1 - \cos(\omega)} \quad (4.5)$$

Um exemplo de sinal de entrada gerado desta forma é mostrado na figura 4.2.

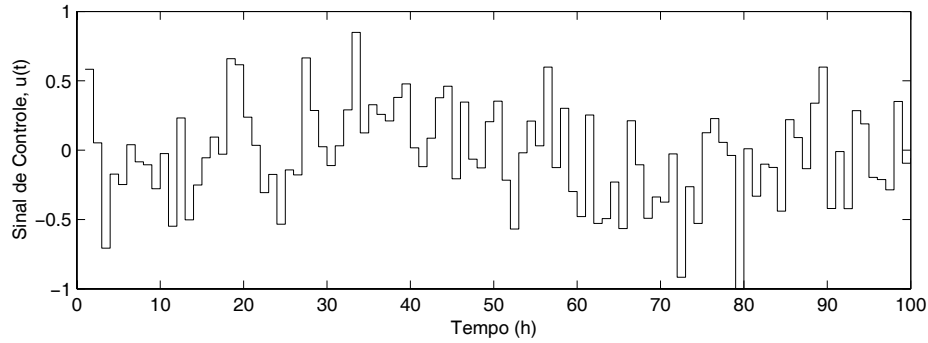


Figura 4.2: Sinal de entrada com nível de ativação variando em intervalos constantes.

II) Uma extensão da metodologia descrita no item **I**, pode ser obtida introduzindo-se uma variável aleatória adicional que determine *quando* o sinal de entrada será alterado:

$$u(t) = \begin{cases} u(t-1) & \text{com probabilidade } \alpha \\ e(t) & \text{com probabilidade } 1 - \alpha \end{cases} \quad (4.6)$$

Este sinal tem a seguinte função de covariância

4.3 Realização de Experimentos

$$R_u(\tau) = \alpha^\tau \sigma_e^2 \quad (4.7)$$

correspondendo à densidade espectral

$$\phi(\omega) = \frac{\sigma_e^2}{2\pi} \frac{1 - \alpha}{1 + \alpha^2 - 2\alpha \cos(\omega)} \quad (4.8)$$

Exemplos de sinais de entrada gerados desta forma são mostrados na figura 4.3.

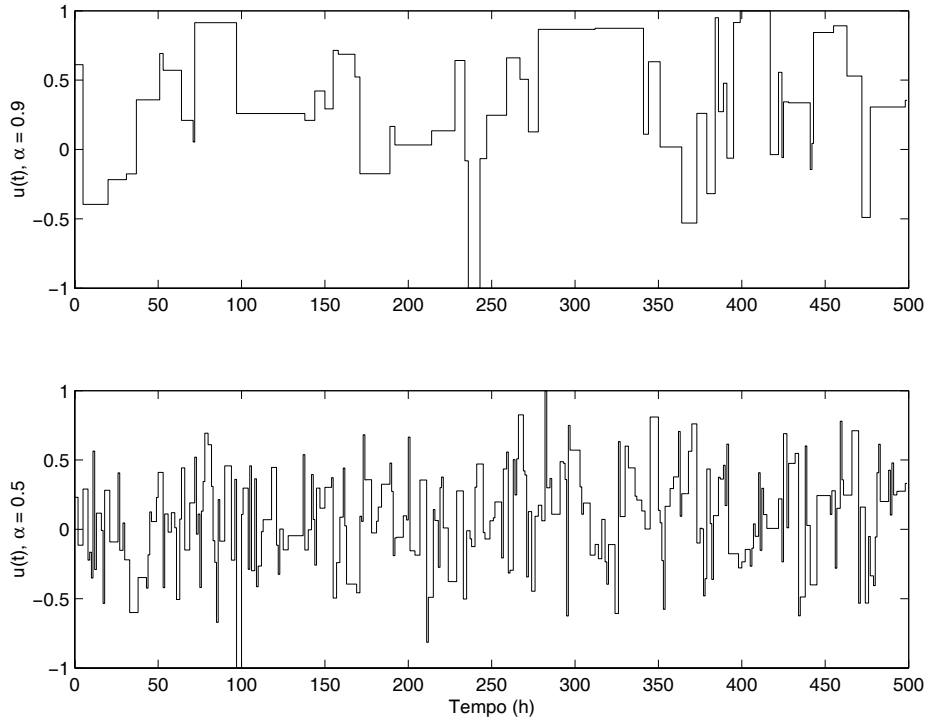


Figura 4.3: Sinal de entrada com nível de ativação variando em intervalos aleatórios.

4.3.5 Preparação dos Dados para Modelagem

Uma análise prévia dos dados é extremamente importante e geralmente evita ou, no mínimo, reduz a extenuante tarefa de testar um grande número de diferentes estruturas de modelos e esquemas de treinamento. O arquivo de dados destinado para a

identificação do processo pode conter erros de medidas e outras características tais como ruídos e dados fora da faixa de interesse, que não devem ser incorporados ao modelo. Este tipo de problema deve ser eliminado antes da tarefa de identificação ser iniciada. Os procedimentos utilizados com mais frequência são (NØRGAARD *et al.*, 2000): *i*) Filtragem; *ii*) Remoção de redundâncias; e *iii*) Escalonamento.

i) Filtragem: Este é um procedimento amplamente utilizado para remover ruídos, perturbações periódicas e efeitos de dinâmicas indesejáveis dos dados medidos. Alguns tipos de ruídos de alta frequência devem ser removidos com filtros adequados, do tipo “passa baixa”, antes da amostragem. As perturbações de baixa frequência podem ser removidas por filtros discretos após a amostragem dos dados.

ii) Remoção de redundâncias: Normalmente, uma simples “inspeção visual” através de gráficos pode revelar algumas imperfeições e tendências no conjunto de dados. Geralmente os problemas relacionados com a quebra ou falhas em sensores são facilmente identificáveis. Outros problemas, contudo, são mais sutis e a sua detecção exige um pouco mais de experiência do projetista. Um exemplo clássico é quando o conjunto de dados apresenta um número elevado de pares de entrada-saída concentrados em uma pequena faixa de operação do sistema. Quando treinado com um conjunto de dados como este, o modelo apresentará resultados muito mais acurados na faixa de operação que contiver mais informação. Neste caso em particular, significa que o modelo fornecerá as melhores previsões em uma faixa operacional muito restrita, o que obviamente não é interessante. A eliminação de dados redundantes correspondentes a esta faixa de operação assegura uma melhor ponderação da informação fornecida ao modelo. Outro benefício advindo da eliminação de dados redundantes é a redução do tempo de treinamento.

Por outro lado, deve-se ressaltar que a eliminação de parte do conjunto de dados, ainda que redundantes, pode acarretar em problemas de identificação quando se utiliza redes neurais com estrutura recorrente⁴. As conexões “feedback” características destes modelos dão origem a regimes transientes sempre que ocorrem mudanças abruptas no sinal de saída. Como tais transientes levam algum tempo para desaparecer, a eliminação aleatória de informações sobre a dinâmica do sistema pode acarretar em problemas na etapa de treinamento (NØRGAARD *et al.*, 2000).

⁴Redes com este tipo de estrutura não serão utilizadas nesta tese.

4.3 Realização de Experimentos

iii) **Escalaonamento:** Escalonar todos os dados dentro de um mesmo intervalo é um procedimento altamente recomendável pois, não raro, as variáveis de interesse dentro de um mesmo processo são medidas em diferentes unidades físicas. Sem o escalonamento há uma tendência de que o sinal de maior magnitude seja muito dominante, o que pode prejudicar o processo de identificação do modelo. Utilizar dados escalonados também favorece a robustez numérica do algoritmo de treinamento, além de acelerar a sua convergência (LE CUN *et al.*, 1991).

4.4 Seleção da Estrutura de Modelos Neurais

A estrutura de um modelo matemático é o “esqueleto” do modelo que, quando associada aos parâmetros, formam o próprio modelo (AGUIRRE, 2000). Em uma função de transferência, por exemplo, o número de pólos e zeros definem completamente a estrutura de tal modelo. Determinar a *Estrutura do Modelo* significa, em linhas gerais, escolher um modelo candidato dentro de uma classe de modelos previamente selecionada. De uma maneira geral, o problema relacionado com a seleção da estrutura do modelo pode ser dividido em duas etapas:

- 1 Selecionar uma “família” de modelos cuja estrutura seja considerada apropriada para descrever o sistema de interesse, por exemplo, modelos lineares (FIR, ARX, MA, ARMAX, etc.), modelos neurais (MLP, RBF, etc.), modelos nebulosos (Lingüístico, Relacional, Takagi-Sugeno, etc), entre outros.
- 2 Escolher um subconjunto da família de modelos com a estrutura selecionada. Dentro da família dos modelos com estrutura linear, pode-se escolher, por exemplo, uma estrutura do tipo ARX(2,3,1), onde (2,3,1) significa que a saída atual depende de duas saídas passadas e três entradas passadas, além de considerar um atraso de um período de amostragem.

4.4.1 Os Regressores

Na identificação de *sistemas lineares*, a determinação da estrutura consiste basicamente da determinação da ordem do modelo. Já no caso de *sistemas não lineares*, o problema torna-se significativamente mais complexo e, considerando que a identificação do processo será feita com redes neurais artificiais, a seleção da estrutura do modelo consiste em escolher não somente as entradas da rede mas também um conjunto adequado de *regressores*, além da própria arquitetura da rede neural. Por regressores, entende-se o conjunto de termos passados relevantes, associado às variáveis de entrada e saída do processo, que serão utilizados como entradas do modelo a ser estimado. A necessidade de escolher um valor adequado para a ordem de um sistema pode ser entendida verificando-se que, se a ordem utilizada for muito menor do que a ordem efetiva

4.4 Seleção da Estrutura de Modelos Neurais

do sistema real, o modelo não possuirá a complexidade estrutural necessária para reproduzir a dinâmica do sistema. Por outro lado, se a ordem do modelo for muito maior do que o necessário, a estimação dos parâmetros provavelmente será mal condicionada (apresentará problemas numéricos). Para o caso de modelos baseados em redes neurais artificiais, a estratégia utilizada com mais frequência é selecionar os regressores baseado nos conceitos advindos da teoria de identificação de sistemas lineares e então determinar a melhor arquitetura possível da RNA tomando os regressores como entradas da rede neural. A topologia de uma rede neural, por sua vez, é equivalente à estrutura do respectivo modelo matemático linear. NØRGAARD *et al.* (2000) destacam as seguintes vantagens nesta abordagem:

- É uma extensão natural da (bem estabelecida) abordagem linear.
- A arquitetura interna da rede pode ser expandida para possibilitar mapeamentos não lineares mais complexos.
- Nível de complexidade não muito alto associado à determinação da estrutura do modelo.
- Adequado para projeto de controladores.

O equivalente não linear das estruturas de modelos lineares apresentadas na seção 3.3 pode ser obtido genericamente por

$$y(t) = g[\boldsymbol{\varphi}(t, \boldsymbol{\theta}), \boldsymbol{\theta}] + e(t) \quad (4.9)$$

que escrito na forma de preditor fica

$$\hat{y}(t | \boldsymbol{\theta}) = g[\boldsymbol{\varphi}(t, \boldsymbol{\theta}), \boldsymbol{\theta}] \quad (4.10)$$

onde $\boldsymbol{\varphi}(t, \boldsymbol{\theta})$ é o vetor de regressão, $\boldsymbol{\theta}$ é o vetor de parâmetros ajustáveis da rede neural, também conhecidos como *pesos*, e g é a função que realiza o mapeamento não linear, que aqui é representada por uma rede neural com estrutura *feedforward*.

Várias estruturas de modelos não lineares podem ser construídas dependendo da escolha dos regressores. Escolhendo-se o vetor de regressão de acordo com a estrutura ARX, a estrutura do modelo não linear correspondente é denominado de NNARX, que é o acrônimo para “*Neural Network ARX*”. Da mesma forma, as estruturas NNFIR, NNARMAX e NNOE são geradas a partir das estruturas lineares correspondentes. A seguir, as estruturas de modelos neurais (não lineares) correspondentes às estruturas lineares apresentadas na seção 3.3 serão apresentadas.

4.4.2 Estruturas Neurais do tipo NNFIR e NNARX

Assim como seus correspondentes lineares, os preditores associados a estas estruturas neurais sempre serão estáveis pelo fato de haver apenas uma relação algébrica entre as predições e as entradas e saídas passadas (NØRGAARD *et al.*, 2000). Esta característica é de particular importância em sistemas não lineares pois as questões relacionadas com a estabilidade assumem uma importância ainda maior neste caso.

As redes neurais com estruturas NNFIR e NNARX estão representadas graficamente nas figuras 4.4 e 4.5, respectivamente. A ausência de problemas relacionados com a estabilidade dos modelos faz com que estas estruturas, em especial a NNARX, sejam utilizadas com mais frequência quando o nível de ruído é baixo.

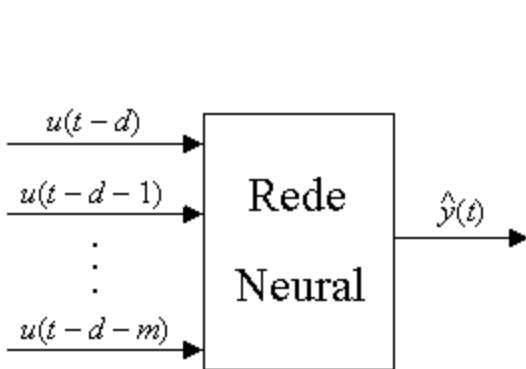


Figura 4.4: Rede neural com estrutura NNFIR.

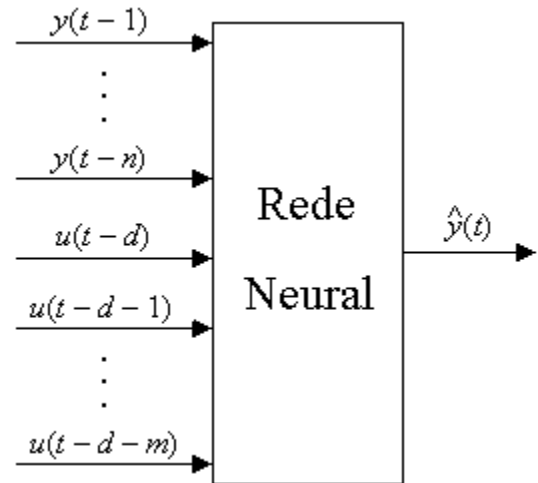


Figura 4.5: Rede neural com estrutura NNARX.

4.4 Seleção da Estrutura de Modelos Neurais

4.4.3 Estrutura Neural do tipo NNARMAX

Embora a função g na equação (4.10) seja representada por uma rede neural do tipo *feedforward*, o preditor apresentará realimentação quando os regressores forem selecionados de acordo com a estrutura dos modelos ARMAX, ou seja

$$\boldsymbol{\varphi}(t, \boldsymbol{\theta}) = [y(t) \ \dots \ y(t-n), \ u(t-d) \ \dots \ u(t-d-m), \ \epsilon(t, \boldsymbol{\theta}) \ \dots \ \epsilon(t-k, \boldsymbol{\theta})]^T \quad (4.11)$$

Os erros de predição passados dependem da saída do modelo e, conseqüentemente, fica estabelecida uma realimentação. A estrutura neural NNARMAX está representada na figura 4.6.

A estabilidade do preditor para o modelo ARMAX linear (equação (3.40)), é determinada avaliando as raízes do polinômio C . Já a análise das propriedades de estabilidade do modelo NNARMAX é mais difícil. Normalmente considera-se a estabilidade destes modelos como uma propriedade local, o que significa dizer que a estabilidade do modelo NNARMAX dependerá de seu ponto de operação.

4.4.4 Estrutura Neural do tipo NNOE

Alguns dos regressores associados à estrutura neural NNOE (figura 4.7) são predições de saídas passadas e, portanto, esta estrutura está sujeita aos mesmos problemas apresentados pela estrutura neural NNARMAX.

Das estruturas neurais apresentadas acima, a NNARX é a empregada com mais frequência pois é aquela cujo preditor não apresenta realimentação⁵ ou *feedback*. Os outros modelos (NNARMAX e NNOE) apresentam realimentações através da escolha dos regressores, o que na terminologia das redes neurais significa dizer que a rede se tornou recorrente, ou seja, as futuras entradas da rede dependerão de suas saídas passadas

⁵Embora o preditor associado à estrutura NNFIR também não apresente realimentação, o seu emprego na identificação de sistemas não lineares não é usual, uma vez que a ausência do vetor de saídas atrasadas como parâmetros de entrada para a rede dificulta a identificação da dinâmica do processo.

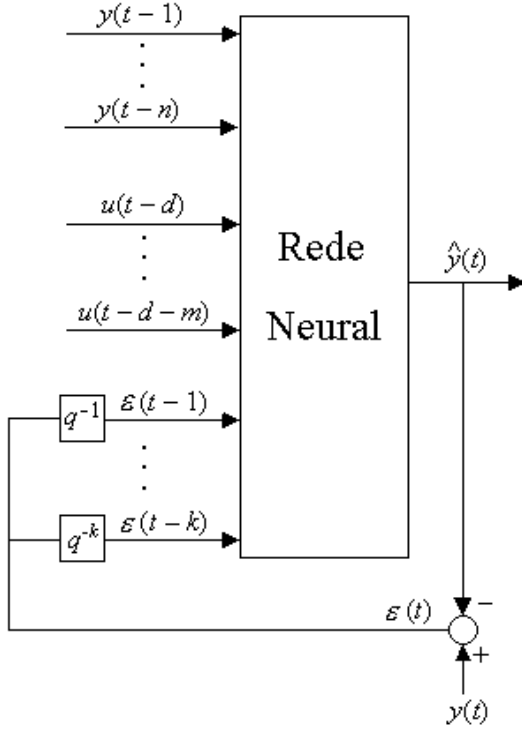


Figura 4.6: Rede neural com estrutura NNARMAX.

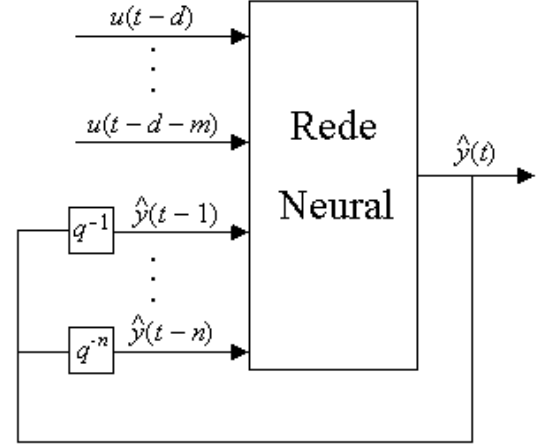


Figura 4.7: Rede neural com estrutura NNOE.

e presente. Isto pode levar à instabilidade em certos regimes de operação e pode ser muito difícil determinar se o preditor é ou não estável. Existem algumas estruturas neurais que são variações da NNARMAX onde um filtro linear do tipo média móvel é empregado para filtrar os resíduos. Contudo, a regra básica é utilizar a estrutura NNARX sempre que possível (NØRGAARD *et al.*, 2000).

Existem ainda algumas variações das estruturas neurais descritas acima, além da abordagem híbrida que mescla estruturas de modelos lineares e não lineares. Estas últimas são implementadas introduzindo-se conexões diretas entre as entradas e saídas do modelo. Assim, as partes lineares passam a ser subconjuntos das estruturas neurais. Alguns exemplos interessantes foram propostos por NARENDRA e PARTHASARATHY (1990), onde duas redes neurais com estrutura NNARX são utilizadas para identificar e controlar sistemas dinâmicos. As estruturas gerais desses modelos são representadas da seguinte forma:

4.4 Seleção da Estrutura de Modelos Neurais

$$\hat{y}(t | \boldsymbol{\theta}) = g_y[\boldsymbol{\varphi}_y(t), \boldsymbol{\theta}_y] + g_u[\boldsymbol{\varphi}_u(t), \boldsymbol{\theta}_u] \quad (4.12)$$

$$\hat{y}(t | \boldsymbol{\theta}) = g[\boldsymbol{\varphi}_y(t), \boldsymbol{\theta}_y] + q^{-d}B(q^{-1})u(t) \quad (4.13)$$

$$\hat{y}(t | \boldsymbol{\theta}) = [1 - A(q^{-1})]y(t) + g_u[\boldsymbol{\varphi}_u(t), \boldsymbol{\theta}_u] \quad (4.14)$$

onde

$$\boldsymbol{\varphi}_y(t) = [y(t-1) \ \dots \ y(t-n)]^T \quad (4.15)$$

$$\boldsymbol{\varphi}_u(t) = [u(t-d) \ \dots \ u(t-d-m)]^T \quad (4.16)$$

De modo similar, NØRGAARD *et al.* (2000) descrevem algumas variações nas estruturas de modelos do tipo NNOE e NNARMAX. Uma modificação particularmente interessante é desenvolvida a partir da equação (4.14) para eliminar os problemas relacionados com a estabilidade dos modelos do tipo OE:

$$\hat{y}(t | \boldsymbol{\theta}) = [1 - A(q^{-1})]\hat{y}(t | \boldsymbol{\theta}) + g[\boldsymbol{\varphi}_u(t), \boldsymbol{\theta}_u] \quad (4.17)$$

Seguindo o mesmo procedimento, uma variação do modelo NNARMAX pode ser obtido quando se introduz a perturbação:

$$\hat{y}(t | \boldsymbol{\theta}) = g[\boldsymbol{\varphi}_{uy}(t), \boldsymbol{\theta}_{uy}] + C(q^{-1})\epsilon(t, \boldsymbol{\theta}) \quad (4.18)$$

onde

$$\boldsymbol{\varphi}_{uy}(t) = [y(t-1) \ \dots \ y(t-n), \ u(t-d) \ \dots \ u(t-d-m)]^T \quad (4.19)$$

Um ponto muito importante que deve ser destacado dentro da discussão relacionada com a seleção da melhor estrutura do modelo é que já existem algumas técnicas disponíveis para automatizar esta tarefa. As duas abordagens possíveis são: *i) Métodos*

*de Poda*⁶, onde uma estrutura considerada grande o suficiente para representar o sistema é determinada *a priori* e então é feita a redução gradual desta estrutura até que uma estrutura ótima seja alcançada; *ii) Métodos Construtivos* onde a estrutura da rede neural é construída iterativamente, a partir de um único neurônio, até que a adição de mais neurônios à estrutura da rede não melhore a sua capacidade de aproximação. Uma discussão mais aprofundada sobre as vantagens e desvantagens de cada um dos métodos foi feita na seção 2.2.

⁶*Pruning.*

4.5 Estimação de Modelos Neurais

O processo de selecionar um modelo a partir de uma estrutura previamente determinada é conhecido na literatura estatística como *estimação*. No entanto, o mesmo processo na comunidade de redes neurais artificiais é tratado como *treinamento* ou *aprendizado* (NØRGAARD *et al.*, 2000).

Uma vez que um conjunto de modelos candidatos tenham sido escolhidos, o passo seguinte é escolher um modelo em particular deste conjunto. Normalmente tal escolha é feita baseada em algum tipo de critério de desempenho. Este critério pode ser formulado de várias maneiras, contudo, ele deve estar relacionado com a futura aplicação do modelo. A estratégia mais freqüente é escolher o modelo que exhibe o melhor desempenho em predições *um passo adiante* em termos do menor erro quadrático médio entre as saídas observadas e as predições do modelo. Todavia, este critério⁷ pode não ser adequado para aplicações em estratégias de controle avançado, onde o desempenho desejado é não apenas a melhor predição um passo adiante, mas também predições o mais acuradas possível em um horizonte relativamente grande (predições k passos adiante). A necessidade de modelos que forneçam bons resultados em simulações recursivas está intimamente relacionada com a filosofia dos controladores preditivos.

Mesmo que o critério para caracterizar o melhor modelo seja simples, não é fácil garantir que o algoritmo de treinamento convergirá para o ponto ótimo. O problema está relacionado ao fato de que freqüentemente existem muitos mínimos locais no mapeamento a ser aproximado e, como consequência, a tarefa de alcançar o mínimo global torna-se muito difícil.

A seguir são apresentadas algumas das abordagens mais utilizadas para o treinamento de redes neurais artificiais, dando ênfase no seu emprego para a modelagem de sistemas dinâmicos.

⁷Uma discussão mais aprofundada sobre a escolha adequada de critérios de desempenho é feita na seção 4.6.

4.5.1 Modelagem Direta

Seja uma planta dinâmica discreta, causal, invariante no tempo cuja entrada, $u(\cdot)$, é uma função do tempo uniformemente limitada e cuja saída é representada por $y_p(\cdot)$. A planta é considerada estável, com parametrização conhecida mas com os valores dos parâmetros desconhecidos. O objetivo da identificação é construir um modelo neural adequado de modo que, quando submetido à mesma entrada u da planta, forneça uma saída y_m que se aproxime de y_p segundo um determinado critério. Este processo de identificação está representado graficamente na figura 4.8.

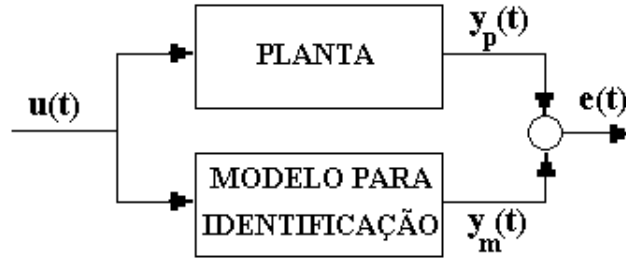


Figura 4.8: Identificação de uma planta utilizando RNA.

O procedimento de treinamento de uma rede neural para representar a dinâmica de uma planta é conhecido como modelagem direta (*“forward modelling”*). O modelo neural é colocado em paralelo com a planta e os erros entre as saídas do sistema e as da rede neural (os erros de predição) são utilizados como sinal para o treinamento da rede. Este procedimento é detalhado na seqüência.

Considere que a planta é descrita pela seguinte equação de diferenças não linear:

$$y_p(t+1) = F[y_p(t), \dots, y_p(t-n+1); u(t), \dots, u(t-m+1)] \quad (4.20)$$

A saída da planta y_p no tempo $t+1$ depende dos n valores passados da própria saída e dos m valores passados da entrada u . Neste ponto, somente a parte dinâmica da resposta da planta será considerada; o modelo não representa explicitamente as possíveis perturbações agindo sobre a planta (CHEN *et al.* (1990), por exemplo, descrevem um método que considera perturbações). NARENDRA e PARTHASARATHY (1990) fazem

4.5 Estimação de Modelos Neurais

uma descrição pormenorizada de métodos de identificação de sistemas utilizando redes neurais, além de analisarem casos especiais do modelo representado pela equação (4.20).

A abordagem mais comum para a modelagem de sistemas é escolher a estrutura de entrada-saída da rede neural idêntica à do sistema que se deseja modelar. Denotando a saída da rede neural por y_m , obtém-se a seguinte representação:

$$y_m(t+1) = \hat{F}[y_p(t), \dots, y_p(t-n+1); u(t), \dots, u(t-m+1)] \quad (4.21)$$

Na equação acima, $\hat{F}(\cdot)$ representa o mapeamento não linear entrada-saída realizado pela rede neural para aproximar o mapeamento $F(\cdot)$ da planta. Deve-se notar que as entradas da rede neural incluem os valores passados da saída da *planta* e não os valores passados da saída da *rede*, ou seja, a rede não apresenta realimentação (*feedback*). Neste caso, o algoritmo de aprendizado estático *backpropagation* pode ser utilizado para encontrar os valores ótimos dos pesos da rede neural. A estrutura do modelo representada pela equação (4.21) é chamada *série-paralelo* (NARENDRA e PARTHASARATHY, 1990). A estrutura de identificação resultante é ilustrada na figura 4.9, onde a sigla “LDA” é a abreviação para *Linha de Derivação de Atrasos*.

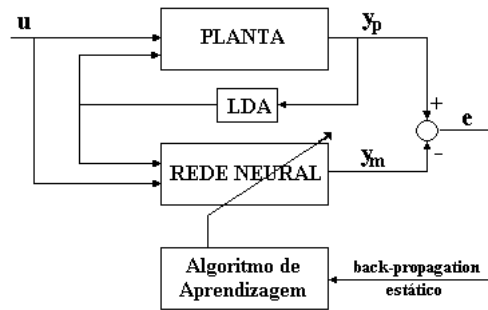


Figura 4.9: Estrutura de identificação série-paralelo.

Se considerarmos que após um tempo adequado de treinamento a rede neural é capaz de fornecer uma boa representação da planta, ou seja, $y_m \approx y_p$, então a própria saída da rede neural e seus valores atrasados podem ser realimentados e, neste caso, a rede neural pode ser utilizada independentemente da planta. Tal rede neural é descrita por:

$$y_m(t+1) = \hat{F}[y_m(t), \dots, y_m(t-n+1); u(t), \dots, u(t-m+1)] \quad (4.22)$$

Esta estrutura (4.22) também pode ser utilizada desde o início, ou seja, durante todo o processo de aprendizado e, neste caso, é chamada de *estrutura paralela* (NARENDRA e PARTHASARATHY, 1990).

Pode-se ainda considerar um caso especial do sistema descrito pela equação (4.20) onde os efeitos das entradas e saídas são aditivos:

$$y_p(t+1) = F[y_p(t), \dots, y_p(t-n+1)] + G[u(t), \dots, u(t-m+1)] \quad (4.23)$$

A estrutura deste modelo é ilustrada na figura 4.10.

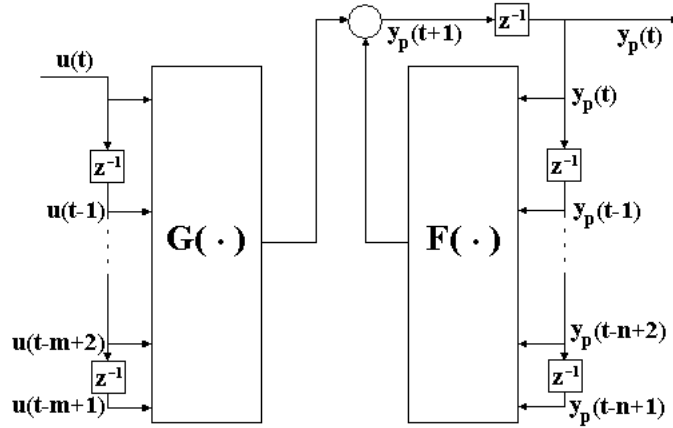


Figura 4.10: Estrutura da dinâmica da planta.

Embora a abordagem geral descrita anteriormente também possa ser aplicada, é mais razoável utilizar a característica aditiva da estrutura da planta. Portanto, o modelo é descrito pela seguinte equação série-paralelo:

$$y_m(t+1) = \hat{F}[y_m(t), \dots, y_m(t-n+1)] + \hat{G}[u(t), \dots, u(t-m+1)] \quad (4.24)$$

onde os mapeamentos \hat{F} e \hat{G} são implementados através de duas redes neurais independentes. Da mesma forma, os pesos de ambas as redes neurais, \mathbf{W}_F e \mathbf{W}_G são ajustados de forma independente utilizando o algoritmo *backpropagation* estático.

4.5 Estimação de Modelos Neurais

A importância da classe de entradas a ser utilizada no treinamento de sistemas de aprendizado já é bem conhecida. O conjunto de treinamento tem que ser representativo de toda a classe de entradas a que o sistema pode estar sujeito. Isto garantirá que o sistema responderá da maneira desejada, mesmo quando uma entrada não pertencente ao conjunto de treinamento for aplicada.

4.5.2 Modelagem Inversa

Nesta abordagem, o modelo do sistema dinâmico fornece uma entrada para que a saída desejada seja obtida. Tais modelos desempenham um papel fundamental em estruturas neurais com propósitos de controle. Contudo, a obtenção de modelos inversos envolve questões muito importantes relacionadas à existência da inversa da planta. Conceitualmente, a abordagem mais simples é a *modelagem direta inversa*, mostrada na figura 4.11, onde um sinal de treinamento sintético (a entrada da planta) é introduzido no sistema. A saída da planta é utilizada como entrada para a rede neural e a saída desta é então comparada com o sinal de treinamento (a entrada do sistema). O erro gerado é utilizado para ajustar os parâmetros da rede neural através de um algoritmo de treinamento. Esta estrutura força a rede neural a representar a inversa da planta, fato que normalmente gera alguns problemas:

- O procedimento de aprendizagem não é direcionado ao objetivo (*goal directed*); o sinal de treinamento deve ser escolhido de modo a cobrir uma ampla faixa de entradas do sistema, porém as entradas correspondentes às faixas reais de operação podem ser difíceis de definir *a priori*. O objetivo real no contexto de controle é fazer com que a saída do sistema se comporte da maneira desejada, assim, o sinal de treinamento na modelagem direta inversa não corresponde explicitamente ao objetivo final. Este tópico está fortemente relacionado com o conceito geral de excitação persistente;
- Uma inversa incorreta pode ser gerada caso o sistema não linear não seja do tipo *one-to-one* (uma entrada fornece uma, e apenas uma, saída).

Uma segunda abordagem de modelagem inversa que visa superar estes problemas é conhecida como *aprendizado inverso especializado* (PSALTIS *et al.*, 1988; NG,

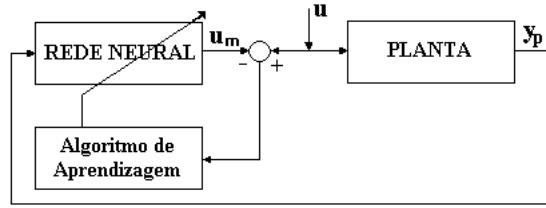


Figura 4.11: Modelagem neural direta inversa.

1997; NØRGAARD *et al.*, 2000); sua estrutura está representada na figura 4.12. Nesta abordagem, o modelo neural inverso precede o sistema e recebe como entrada um sinal de treinamento que cobre o espaço operacional desejado de saída do sistema controlado (isto é, corresponde ao sinal de referência do sistema). Esta estrutura de aprendizado também conta com um modelo neural direto (*forward*) do sistema colocado em paralelo com a planta. Neste caso, o sinal de erro para o algoritmo de treinamento é a diferença entre o sinal de treinamento e a saída do sistema (se o sistema for muito ruidoso, o erro também pode ser definido como sendo a diferença entre o sinal de treinamento e a saída do modelo neural *forward*). Pode-se mostrar que usando a saída da planta é possível produzir uma inversa adequada, mesmo quando o modelo neural direto não é preciso (BRDYŚ, 1995; NG e COOK, 1998; HAGAN e DEMUTH, 1999); o mesmo resultado não é obtido quando se usa a saída do modelo neural direto. O erro pode ser retropropagado através do modelo neural direto e então através do modelo neural inverso. Durante este procedimento, somente os pesos no modelo neural inverso são ajustados.

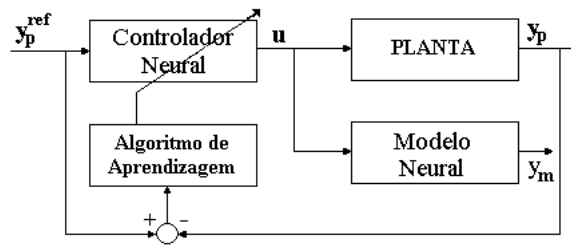


Figura 4.12: Modelagem inversa especializada.

Em comparação com a modelagem direta inversa, o aprendizado especializado inverso apresenta as seguintes características:

- O procedimento é direcionado ao objetivo (*goal directed*) uma vez que é baseado no erro entre as saídas desejadas do sistema e as saídas reais, ou seja, durante

4.5 Estimação de Modelos Neurais

o treinamento o sistema recebe entradas que correspondem às entradas reais que ele receberá posteriormente.

- Quando o mapeamento do sistema não for do tipo *one-to-one*, uma inversa particular (pseudo-inversa) será determinada.

Procedendo uma análise da estrutura entrada-saída da modelagem neural inversa, verifica-se que a determinação da inversa F^{-1} que leva à obtenção de $u(t)$ a partir da equação (4.20), requer o conhecimento do valor futuro $y_p(t+1)$. Para superar este problema, pode-se substituir este valor por $y_p^{ref}(t+1)$ que admite-se estar disponível no tempo t . Esta consideração é bastante razoável uma vez que y_p^{ref} está relacionado com o sinal de referência que, por sua vez, normalmente é conhecido de antemão. Assim, o mapeamento não linear entrada-saída da modelagem neural inversa é dado por:

$$u(t) = F^{-1}[y_p(t), \dots, y_p(t-n+1); y_p^{ref}(t+1); u(t-1), \dots, u(t-m+1)] \quad (4.25)$$

4.6 Validação de Modelos Neurais

Após o modelo ter sido estimado, deve-se proceder a avaliação do mesmo para que se determine se o modelo selecionado atende ou não às necessidades requeridas. A metodologia empregada na validação do modelo identificado deve estar intimamente relacionada com a futura aplicação do modelo. Esta é geralmente a etapa mais imprecisa do procedimento de identificação, uma vez que os critérios de aceitabilidade do modelo nem sempre são muito claros ou fáceis de definir matematicamente.

Existem na literatura, uma série de testes padrões que podem ser realizados com relativa facilidade quando comparados com os testes específicos necessários para avaliar o desempenho do modelo para fins de controle. Estes testes permitem obter uma validação preliminar, onde os piores modelos podem ser descartados rapidamente.

A maioria destes testes utiliza um conjunto de dados de validação que, além de ser diferente do conjunto de dados utilizado durante o treinamento, deve conter dados representativos de toda a faixa operacional do sistema. Este procedimento, conhecido genericamente por “validação cruzada” ou *cross-validation* (HANSEN e SALAMON, 1990), é uma medida direta da capacidade de generalização da rede neural e, se aplicado simultaneamente ao processo de aprendizado, pode servir não só para avaliar continuamente o desempenho do modelo que está sendo identificado como também como um critério de parada para o algoritmo de treinamento. Este critério de parada é baseado no conceito de que se o erro associado ao conjunto de validação começar a crescer continuamente (a despeito do erro de treinamento continuar decrescendo), significa que a rede neural está ficando muito especializada no conjunto de dados de treinamento e, conseqüentemente, perdendo a capacidade de generalização.

Existem ainda ao menos outros três métodos clássicos de validação para avaliar o desempenho do modelo neural:

- Testes de *Estimação do Erro de Generalização*;
- Testes de *Correlação* ou *Análise de Resíduos*;
- Testes de *Inspeção Visual das Predições*.

4.6 Validação de Modelos Neurais

4.6.1 Testes de Estimação do Erro de Generalização

Embora tais testes não sejam muito confiáveis, principalmente quando se trata de sistemas não lineares, as estimativas dos erros médios de generalização podem fornecer uma idéia geral sobre a capacidade de predição do modelo. Entretanto, a principal aplicação deste tipo de testes é para a seleção da estrutura do modelo. Os resultados dos testes podem ser úteis para uma rápida comparação entre diferentes estruturas de modelos, auxiliando na análise sobre qual dos modelos tem o melhor potencial.

O teste mais aplicado para este tipo análise é o *Critério de Informação de Akaike* (CIA) ou simplesmente *Critério de Akaike*:

$$CIA(n_\theta) = N \ln[\sigma_{erro}^2(n_\theta)] + 2n_\theta \quad (4.26)$$

onde N é o número de dados, $\sigma_{erro}^2(n_\theta)$ é a variância do erro de modelagem ou resíduos (erro de predição um passo adiante) e $n_\theta = \dim[\hat{\boldsymbol{\theta}}]$ é o número de parâmetros do modelo.

O critério de Akaike (4.26) pode ser entendido da seguinte forma (AGUIRRE, 2000): À medida que novos termos são incluídos no modelo, o número de graus de liberdade aumenta permitindo um ajuste de dados mais preciso. Assim, $\sigma_{erro}^2(n_\theta)$ diminui na medida em que n_θ aumenta e, para o caso em que n_θ for igual ao número de restrições obtidas a partir do modelo, a matriz de regressores, Ψ , é quadrada e o vetor de parâmetros pode ser determinado exatamente:

$$\hat{\boldsymbol{\theta}} = \Psi^{-1}\mathbf{y} \quad (4.27)$$

Nesse caso, $\sigma_{erro}^2(n_\theta) = 0$.

O fato da variância dos resíduos ser nula não implica necessariamente que o modelo seja bom pois, a partir de determinado ponto, a diminuição da variância dos resíduos devido à inclusão de um novo termo passa a ser insignificante. Isso pode ser melhor entendido considerando a equação (4.26), cuja primeira parcela mede a diminuição da variância dos resíduos devido à inclusão de um novo termo, enquanto que a segunda parcela é responsável pela penalização da inclusão de cada termo. Assim,

se a redução da variância dos resíduos, $\ln[\sigma_{erro}^2(n_\theta)]$, for menor do que o custo de incluí-lo, $2n_\theta$, este termo não deve ser incluído no modelo.

Existem outros critérios de informação que são bastante semelhantes ao critério de Akaike, dentre os quais destacam-se o *Critério de Informação de Bayes* (CIB) e o *Critério do Erro Final de Predição* (EFP):

$$CIB(n_\theta) = N \ln[\sigma_{erro}^2(n_\theta)] + n_\theta \ln N \quad (4.28)$$

$$EFP(n_\theta) = N \ln[\sigma_{erro}^2(n_\theta)] + N \ln \left[\frac{N + n_\theta}{N - n_\theta} \right] \quad (4.29)$$

O critério de Akaike geralmente atinge um mínimo para um determinado conjunto de parâmetros que, dependendo do critério utilizado, pode ser ótimo. Este critério é especialmente útil para a determinação da ordem de modelos lineares. Considerando um sistema SISO, isto seria feito através do aumento gradativo da ordem do modelo, ou seja, testa-se $y(k-1), u(k-1)$, depois um incremento para $y(k-2), u(k-2)$, e assim por diante, seguido da verificação correspondente do $CIA(n_\theta)$ referente a cada aumento de ordem do modelo. A passagem por um mínimo é um indicativo de que o procedimento deve ser terminado.

4.6.2 Testes de Correlação ou Análise de Resíduos

Embora os testes de inspeção visual forneçam uma boa idéia da qualidade do modelo identificado, eles não indicam se o modelo tem falhas que possam ser corrigidas. Ou seja, se houver erros significativos entre as predições do modelo e os dados reais, não é possível saber se o resultado insatisfatório é fruto de problemas na estimação dos parâmetros ou se o modelo identificado não foi capaz de representar satisfatoriamente todas as dinâmicas do processo em questão. Em contrapartida, o teste de análise dos resíduos é um procedimento que fornece uma indicação direta se os parâmetros do modelo identificado foram ou não estimados corretamente.

Considere o vetor de resíduos definido da seguinte maneira:

$$\boldsymbol{\xi} = \mathbf{y} - \Psi \hat{\boldsymbol{\theta}} \quad (4.30)$$

onde \mathbf{y} é o vetor de saídas medidas (reais), Ψ é a matriz de regressores do modelo e $\hat{\boldsymbol{\theta}}$ é o vetor de parâmetros estimados.

Pode-se perceber que, no que se refere ao modelo, o vetor de resíduos depende tanto dos parâmetros estimados, $\hat{\boldsymbol{\theta}}$, quanto dos regressores que compõem o modelo, Ψ . Se o estimador for baseado no critério dos mínimos quadrados, então a soma dos quadrados dos resíduos é mínima para um dado vetor de regressores, o que significa que a variância dos resíduos não é um indicador adequado para validar modelos dinâmicos.

Tendo em vista a tarefa de validação dos modelos identificados, deve-se ter em mente que os resíduos são a parte do sistema que o modelo não consegue representar. Portanto, se a análise do vetor de resíduos indicar que esta variável apresenta uma distribuição branca, isto significa que não há informação útil nos resíduos, ou seja, os dados estão totalmente não correlacionados e que o modelo gerou a melhor representação possível do sistema. Contudo, este resultado não garante que a simulação livre do modelo (infinitos passos adiante) apresentará bons resultados, mas apenas que a simulação um passo adiante será boa. Caso a avaliação dos resíduos não apresente uma distribuição característica de ruído branco, isso indica que ainda há informação no conjunto de dados que o modelo não foi capaz de representar. Se o estimador utilizado for do tipo mínimos quadrados, resultados como este indicam que o estimador está polarizado e que deve ser escolhido um novo conjunto de regressores para solucionar o problema.

Por outro lado, deve-se destacar que uma das principais condições para que o estimador baseado nos mínimos quadrados não apresente polarização (tendência) na determinação de $\hat{\boldsymbol{\theta}}$ é que o erro na equação de regressão seja do tipo ruído branco (AGUIRRE, 2000). Dado que o modelo identificado, $y(k) = \boldsymbol{\psi}^T \hat{\boldsymbol{\theta}} + \xi(k)$, é obtido diretamente da equação de regressão após a estimação dos parâmetros, fica claro que, para que a esperança matemática da estimativa dos parâmetros seja tal que $E[\hat{\boldsymbol{\theta}}_{MQ}] = \boldsymbol{\theta}$, é necessário que os resíduos sejam brancos (AGUIRRE, 2000). Assim, o método de validação denominado *Análise de Resíduos* refere-se a um conjunto de testes efetuados para verificar se os resíduos são brancos ou não.

Um vetor de valores residuais, $\xi(k)$, é dito *linearmente branco*⁸ se a sua função de auto-correlação for nula para todos os valores de atraso maiores ou iguais a um (SÖDERSTRÖM e STOICA, 1989; AGUIRRE, 2000):

$$r_{\xi\xi}(k) = 0, \quad \forall k \neq 0 \quad (4.31)$$

Dado que o vetor descrito pela equação (4.30) é resultado de um ajuste dos regressores do modelo (estrutura matemática) a um conjunto de dados e, admitindo que o vetor de resíduos atenda a equação (4.31), o próximo passo é determinar se o modelo identificado apresenta um bom desempenho para um conjunto de dados diferentes, ou seja, determinar a capacidade de generalização do modelo. Uma maneira de verificar o quão genérica é a capacidade de predição do modelo é calculando a função de correlação cruzada entre o vetor de resíduos e o sinal de entrada, ou seja, $r_{u\xi}(k)$. A verificação desta propriedade é dada se

$$r_{u\xi}(k) = 0, \quad \forall k \neq 0 \quad (4.32)$$

o que significa dizer que os erros de predição do modelo independem das entradas do sistema utilizadas para identificá-lo.

Se toda a informação sobre a dinâmica do sistema estiver incorporada no modelo, então os erros de predição (resíduos) devem ser independentes dos sinais de entrada e de saída (SÖDERSTRÖM e STOICA, 1989; NØRGAARD *et al.*, 2000). Em outras palavras, se os resíduos não contiverem informações sobre a dinâmica do sistema ou sobre os resíduos passados, é provável que toda a informação do sistema (representado pelo conjunto de dados de treinamento) tenha sido extraída e que o modelo seja capaz de descrever bem o comportamento do sistema.

Para o caso de sistemas não lineares, é possível que um modelo linear seja capaz de modelar toda a informação linear contida nos dados, contudo, as dinâmicas não lineares não serão modeladas. Em casos como este, a análise dos resíduos atenderá

⁸O termo *linearmente branco* é empregado para indicar que não há auto-correlação *linear* no vetor $\xi(k)$.

4.6 Validação de Modelos Neurais

a ambos os critérios descritos acima (4.31 e 4.32), entretanto, os resíduos apresentarão correlações não lineares. Para que tais correlações sejam detectadas é preciso utilizar um conjunto de funções adequadas. A idéia central deste tipo de teste, objeto de pesquisa de vários autores como por exemplo BILLINGS e colaboradores (BILLINGS e VOON, 1986; BILLINGS *et al.*, 1992; BILLINGS e ZHU, 1994), é calcular e visualizar tais funções de correlação, que convergem para uma distribuição Gaussiana com média zero e variância $1/N$, caso o sistema real tenha sido bem identificado. A investigação dessas propriedades envolve a verificação da descorrelação entre os resíduos e todas as combinações lineares e não lineares com os dados passados. Obviamente um teste como este é infactível na prática e o que se faz normalmente é considerar apenas as funções de autocorrelação e correlação cruzada mais significativas.

BILLINGS e VOON (1986) apresentam um conjunto de funções de correlação que tem sido largamente utilizado na validação de modelos lineares e não lineares, dentre as quais destacam-se as seguintes (SÖDERSTRÖM e STOICA, 1989; AGUIRRE, 2000; NØRGAARD *et al.*, 2000):

$$\begin{aligned} r_{\xi\xi}(\tau) &= E\{\xi(k-\tau)\xi(k)\} = \delta(0) \\ r_{u\xi}(\tau) &= E\{u(k-\tau)\xi(k)\} = 0, \quad \forall \tau \\ r_{u^2\xi}(\tau) &= E\{(u^2(k-\tau) - \overline{u^2(k)})\xi(k)\} = 0, \quad \forall \tau \\ r_{u^2\xi^2}(\tau) &= E\{(u^2(k-\tau) - \overline{u^2(k)})\xi^2(k)\} = 0, \quad \forall \tau \\ r_{\xi(\xi u)} &= E\{\xi(k)\xi(k-1-\tau)u(k-1-\tau)\} = 0, \quad \tau \geq 0 \end{aligned} \tag{4.33}$$

onde $\delta(0)$ é o impulso unitário e a barra sobre a variável indica o seu valor médio.

Considerando o conjunto de equações (4.33), as duas primeiras correspondem às funções de autocorrelação e correlação cruzada lineares, respectivamente, enquanto que as demais são as suas correspondentes não lineares. Existe ainda a possibilidade de derivar outras funções caso as equações (4.33) não sejam capazes de detectar correlações específicas nos resíduos.

Deve-se destacar ainda que todos os testes se aplicam igualmente a sistemas multi-variáveis (MIMO). Neste caso, as funções de correlação devem ser calculadas para cada combinação entrada/saída.

4.6.3 Testes de Inspeção Visual

Embora os testes de estimação do erro de generalização e de análise de resíduos forneçam uma boa idéia da magnitude numérica do erro de identificação, eles não são capazes de dar informações sobre a variação de acurácia do modelo em diferentes pontos de operação do sistema. Este tipo de informação pode ser facilmente obtida através de uma simples visualização gráfica dos dados reais e simulados.

O teste de inspeção visual é baseado na comparação visual entre as saídas verdadeiras e as predições fornecidas pelo modelo. Embora a comparação realizada com mais freqüência seja a análise das predições do modelo um passo adiante, cumpre destacar a importância da avaliação do modelo em predições de longo alcance (infinitos passos adiante, simulação livre, simulação recursiva, ou ainda série sintética) pois, não raro, modelos que apresentam ótimo desempenho em predições um passo adiante, exibem um comportamento inaceitável em simulações livres.

A título de ilustração, será apresentado o desempenho de um modelo linear na simulação do comportamento dinâmico de uma das variáveis de interesse do “Estudo de Caso 1” (Capítulo 6). A identificação foi feita utilizando-se um modelo linear com estrutura ARX. Uma avaliação tomando como base somente as predições do modelo um passo adiante poderia gerar a falsa idéia de que o modelo linear foi capaz de representar bastante bem a dinâmica do sistema. A razão para este desempenho aparentemente bom é que, quando se trata de sistemas dinâmicos, a validação do modelo não é tão trivial quanto a de um sistema estático, cujo objetivo é minimizar o erro quadrático médio (EQM) da aproximação de uma dada superfície. Assim, bons resultados em predições um passo adiante não significam necessariamente que o modelo exibirá predições dinâmicas (simulação livre) aceitáveis, simplesmente porque o critério utilizado para validação (em predições um passo adiante) é o mesmo critério adotado para a estimação do modelo, isto é, a minimização do EQM referente à previsão um passo adiante da saída (erro de previsão).

A figura 4.13 ressalta a importância da avaliação do desempenho de modelos dinâmicos em previsões recursivas (simulação livre), além de ilustrar a utilidade do teste de inspeção visual.

Para esclarecer melhor os conceitos de *predição um passo adiante* e *simula-*

4.6 Validação de Modelos Neurais

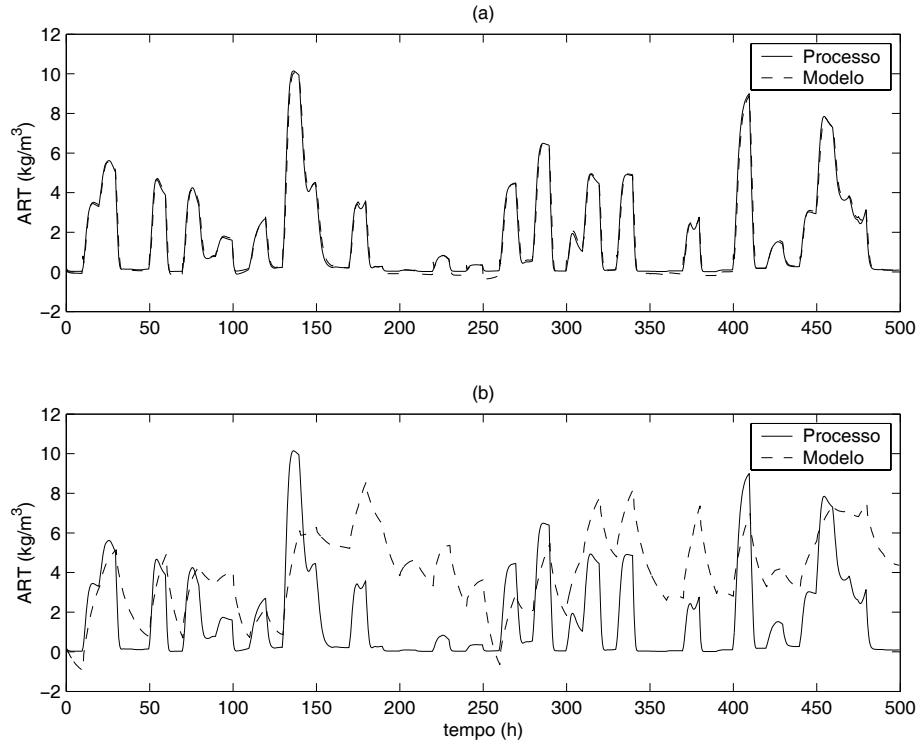


Figura 4.13: Predições um passo adiante (a) e simulação recursiva (b).

ção recursiva, considere o seguinte modelo cujo conjunto de regressores é composto de variáveis tomadas até o instante $(k - 1)$:

$$y(k) = \boldsymbol{\psi}^T(k - 1)\hat{\boldsymbol{\theta}} + \xi(k) \quad (4.34)$$

A *predição um passo adiante* é obtida montando-se o vetor de regressão, $\boldsymbol{\psi}$, com as *observações* (dados reais) obtidas do conjunto de dados até o instante $(k - 1)$ e então calculando a predição da variável de interesse no instante k (um passo adiante):

$$\hat{y}(k) = \boldsymbol{\psi}^T(k - 1)\hat{\boldsymbol{\theta}} \quad (4.35)$$

Para obter a predição no instante $(k + 1)$, tomam-se as *observações* do conjunto de dados (ou da própria planta) até o instante k , ou seja, $\hat{y}(k + 1) = \boldsymbol{\psi}^T(k)\hat{\boldsymbol{\theta}}$, e assim sucessivamente.

Deve-se destacar que, em simulações deste tipo, a predição um passo adiante gerada pelo modelo, $\hat{y}(k)$, não é utilizada na obtenção da próxima predição, $\hat{y}(k+1)$. O erro de predição, $y(k) - \hat{y}(k)$, é definido como o resíduo no instante k que, por sua vez, é utilizado pela maioria dos algoritmos de estimação de parâmetros como argumento da função de custo (minimização do quadrado dos resíduos). Considerando tais algoritmos, torna-se claro que, para um determinado conjunto de regressores, os erros de predição um passo serão minimizados. Desse modo, predições deste tipo não são um bom teste para validar modelos dinâmicos, dado que mesmo modelos ruins costumam apresentar bons resultados em simulações um passo adiante (AGUIRRE, 2000).

Para ilustrar predições do tipo *simulação recursiva*, considere o caso em que o modelo a ser simulado tem o seguinte vetor de regressores:

$$\boldsymbol{\psi}^T(k-1) = [y(k-1) \ y(k-2) \ u(k-1) \ u(k-2)] \quad (4.36)$$

Para iniciar a simulação é preciso iniciar o modelo com valores medidos (reais). Sejam $y(1)$ e $y(2)$ os dois primeiros valores da saída do conjunto de dados e, similarmente $u(1)$ e $u(2)$ correspondem aos seus respectivos valores de entrada. Tem-se então que

$$\hat{y}(3) = [y(2) \ y(1) \ u(2) \ u(1)]\hat{\boldsymbol{\theta}} \quad (4.37)$$

As predições seguintes são feitas utilizando os valores preditos pelo próprio modelo em um procedimento recursivo:

$$\begin{aligned} \hat{y}(4) &= [\hat{y}(3) \ y(2) \ u(3) \ u(2)]\hat{\boldsymbol{\theta}} \\ \hat{y}(5) &= [\hat{y}(4) \ \hat{y}(3) \ u(4) \ u(3)]\hat{\boldsymbol{\theta}} \end{aligned} \quad (4.38)$$

Este procedimento, ao contrário da predição um passo adiante, fornece uma boa medida da capacidade do modelo de representar o comportamento dinâmico do sistema.

Existe ainda um preditor intermediário, o *preditor k passos adiante*, que utiliza o modelo como preditor recursivo apenas durante k intervalos de amostragem, depois dos quais o modelo é reiniciado com dados medidos.

4.7 Reiniciando o Procedimento

Os passos indicados na figura 4.1 sugerem que o procedimento de identificação de sistemas não lineares utilizando redes neurais artificiais deve ser executado de modo iterativo. Por isso, pode ser necessário voltar uma ou mais etapas, especificar vários modelos diferentes, testar estruturas diferentes e, no pior caso, refazer os experimentos até que se consiga um modelo que forneça um resultado adequado. No entanto, os problemas mais comuns apresentados pelas redes neurais artificiais com arquitetura MLP quando utilizadas na identificação de sistemas dinâmicos estão relacionados com a sua sensibilidade aos mínimos locais e a dificuldade na determinação da arquitetura ótima. Estes problemas e algumas de suas soluções serão discutidos na sequência e estão apresentados da seguinte forma:

- Retreinamento da Rede Neural
- Determinação da Arquitetura Neural Ótima
- Realização de Novos Experimentos

4.7.1 Retreinamento da Rede Neural

O algoritmo de treinamento de uma rede neural é na verdade um método iterativo de otimização irrestrita. A condição inicial do método iterativo pode ser interpretada como um ponto na superfície do espaço de solução, a partir do qual a busca pelo mínimo da função objetivo será iniciada. A necessidade de uma condição inicial adequada pode ser entendida através da visualização de uma superfície de erro, caracterizada pela função a ser minimizada, que geralmente apresenta uma grande quantidade de mínimos locais. A solução mais trivial que costuma ser recomendada é que a rede neural seja treinada mais de uma vez com diferentes inicializações para os pesos sinápticos de modo que o modelo com melhor desempenho seja escolhido. O objetivo é tentar partir de um ponto inicial no espaço de busca (determinado aleatoriamente) que conduza, na melhor das hipóteses, ao mínimo global.

O problema da sensibilidade dos algoritmos de treinamento a mínimos locais pode ser melhor contornado através de uma definição mais criteriosa do conjunto inicial

de pesos na etapa de treinamento. DE CASTRO (1998) descreve detalhadamente dois paradigmas mais elaborados para a determinação do melhor conjunto inicial de pesos via treinamento supervisionado: o *Paradigma do caminho mais fácil* e o *Paradigma do caminho mais curto*.

DE CASTRO e VON ZUBEN (1998b) apresentam uma abordagem alternativa, que provou ser simples e robusta, para a solução do mesmo problema. Neste trabalho, os autores propõem um método de inicialização onde os dados de treinamento são utilizados para inicializar os pesos da rede de forma que os neurônios estejam operando inicialmente nas proximidades de sua região mais linear. Os resultados obtidos foram superiores, em média, a todos os demais resultados apresentados na literatura até aquele momento (DE CASTRO, 1998).

IYODA (2000) destacou a aplicação de algoritmos genéticos na extração de características importantes de um conjunto de pesos iniciais adequado para o treinamento. Neste trabalho, o processo evolutivo mostrou que os melhores e os piores conjuntos de pesos iniciais correspondiam aos casos em que os valores absolutos médios eram elevados, enquanto que os desempenhos intermediários foram apresentados pelos conjuntos de pesos cujos valores absolutos médios apresentavam valores reduzidos. Desse modo, verificou-se que existe uma grande possibilidade de produzir conjuntos iniciais de pesos ruins caso seus valores absolutos médios sejam elevados.

Além da sensibilidade a mínimos locais, a capacidade de generalização da rede neural também depende fortemente da magnitude do decaimento dos pesos (*weight decay*). Este conceito pode ser melhor entendido através da definição de alguns conceitos relacionados como a *Teoria da Regularização* que, de modo resumido, pode ser descrita como um procedimento para a obtenção de um problema de aproximação bem comportado através da imposição de restrições de suavidade ao modelo de aproximação (VON ZUBEN, 1996). Os *reguladores* são funcionais de custo adicionados ao funcional de erro que está sendo minimizado e o *decaimento de pesos* é provavelmente o regulador mais conhecido.

A título de ilustração, considere uma rede neural com arquitetura MLP cujo algoritmo de treinamento minimiza a diferença entre as saídas da rede e as saídas desejadas em relação ao valor do vetor de parâmetros θ :

4.7 Reiniciando o Procedimento

$$J(\boldsymbol{\theta}) = \sum_{i=1}^N \sum_{j=1}^m (g_{ij}(\mathbf{x}) - \hat{g}_{ij}(\mathbf{x}, \boldsymbol{\theta}))^2 \quad (4.39)$$
$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^P} J(\boldsymbol{\theta})$$

onde $g(\cdot)$ é a função a ser aproximada, $\hat{g}(\cdot)$ é a aproximação fornecida pela rede, N é o número de amostras e m é o número de saídas.

Os reguladores são termos adicionais cujo objetivo é reduzir a flexibilidade da rede neural no sentido de atender aos critérios de generalização. Tais critérios determinam que o funcional de erro descrito pela equação (4.39) seja minimizado ao mesmo tempo em que se minimiza, por exemplo, a soma do quadrado dos pesos da rede neural. Este procedimento evita um aumento exagerado dos valores dos pesos das unidades intermediárias (DE CASTRO, 1998). Definindo o custo relacionado aos pesos da rede como

$$C_{\text{pesos}}(\boldsymbol{\theta}) = \sum_{i=1}^P \boldsymbol{\theta}_i^2, \quad (4.40)$$

a função de custo total passa a ser definida assim

$$C_{\text{total}} = J(\boldsymbol{\theta}) + C_{\text{pesos}}(\boldsymbol{\theta}) \quad (4.41)$$

Tomando a derivada da equação (4.41), obtém-se uma lei para o ajuste do vetor de parâmetros (que considera os dois termos da equação) na minimização do funcional de custo total. Assim, para uma dada estrutura, determina-se o decaimento de peso e a minimização do erro de aproximação que tendem a produzir uma rede neural com melhor capacidade de generalização.

Finalmente, deve-se estar atento para o fato de que o mínimo global não é necessariamente o melhor mínimo do ponto de vista de capacidade de generalização da rede. Este problema é evidenciado quando a rede está sobre-parametrizada, pois a informação disponível para o treinamento é finita. Com isso, onde não há informação

disponível, o excesso de flexibilidade da rede neural pode produzir estimativas distorcidas. O problema de sobre-parametrização pode ser eliminado ou reduzido através da determinação de uma arquitetura mais adequada para a rede neural.

4.7.2 Determinação da Arquitetura Neural Ótima

A determinação da estrutura ótima de uma rede neural⁹ envolve a especificação de: *i*) O conteúdo do vetor de regressão e *ii*) O número de neurônios da camada interna (número de parâmetros do modelo neural). A prática mais comum é determinar em primeiro lugar uma estrutura para o modelo neural (NNARX, NNARMAX, etc.) e o número de entradas e saídas passadas, ou seja, os vetores de regressão. Uma vez determinada a estrutura dos regressores (o que não é uma tarefa fácil), a etapa seguinte é escolher a melhor arquitetura possível para a rede neural, i.e., o número de neurônios da camada interna. O procedimento mais simples para determinar o número adequado de nós na camada escondida é aumentar gradativamente o número de neurônios e avaliar relação entre as capacidades de aproximação e generalização do modelo neural gerado. Quando o procedimento chegar a um ponto onde o aumento do número de neurônios não corresponder a uma melhora significativa do modelo, considera-se que a arquitetura da rede está determinada. Um ponto importante que deve ser destacado é a relação entre a quantidade de dados disponíveis para o treinamento e o número máximo de neurônios (parâmetros) que deve ser utilizado. Algumas heurísticas sugerem que o ideal é ter algo em torno de dez a quinze dados de treinamento para cada parâmetro ajustável do modelo neural (NØRGAARD *et al.*, 2000).

Existem diferentes tentativas reportadas na literatura para automatizar o processo de seleção da arquitetura neural para um dado conjunto de regressores previamente determinado. As duas principais abordagens são os métodos de poda (*pruning algorithms*) e os métodos construtivos, sendo que este último é empregado no presente trabalho e discutido mais profundamente na seção 2.2. Comparando os métodos construtivo e de poda, fica evidenciada a superioridade do primeiro, porém ambos são superados pelos métodos híbridos que são predominantemente construtivos mas apresentam etapas de construção e poda (NØRGAARD *et al.*, 2000). Existem muitas referências

⁹Assumindo que a rede neural possui apenas uma única camada intermediária.

4.8 Considerações Finais Sobre o Capítulo

que discutem cada uma destas abordagens isoladamente, a comparação entre elas, assim com a abordagem híbrida, dentre as quais: VON ZUBEN e NETTO (1995; 1997), VON ZUBEN (1996), PRECHELT (1996; 1997), DE CASTRO (1998), HENRIQUE (1999), HENRIQUE *et al.* (2000), NØRGAARD *et al.* (2000) e LEUNG *et al.* (2001).

4.7.3 Realização de Novos Experimentos

Um indicativo de que o conjunto de dados é insuficiente para representar adequadamente o comportamento de um sistema é a aparente impossibilidade de estimar um modelo que atenda os requisitos desejados para uma dada aplicação, mesmo quando várias estruturas tenham sido testadas. Normalmente esta impossibilidade está relacionada com a falta de informação no conjunto de dados disponível, ou seja, certos regimes de operação podem não estar representados no conjunto de dados. Neste caso, faz-se necessário realizar experimentos adicionais de modo a adquirir mais informações sobre os regimes e dinâmicas ausentes. Além disso, uma vez que as redes neurais artificiais típicas possuem muitos parâmetros, deve-se estar atento para o problema da “Maldição da Dimensionalidade” (seção 4.3) e coletar o máximo de dados possível.

4.8 Considerações Finais Sobre o Capítulo

O objetivo deste capítulo foi apresentar os conceitos e formalizar o procedimento de identificação de sistemas dinâmicos não lineares utilizando redes neurais artificiais. A abordagem neural foi escolhida como exemplo de modelo não linear pelo fato de apresentar uma série de vantagens em relação a outros modelos não lineares do tipo caixa-preta, sendo a versatilidade de aplicação em diferentes situações uma de suas melhores características. A eficiência na modelagem de sistemas não lineares é outra das razões pelas quais esta técnica de identificação desponta como uma das mais populares. Por outro lado, a obtenção de bons resultados está intimamente relacionada com a correta utilização desta ferramenta. Por este motivo, foram apresentadas ao longo deste capítulo as diversas etapas que devem ser realizadas para que se possa extrair das redes neurais e, por extensão, de muitos outros modelos não lineares, todo o seu potencial para modelagem de sistemas não lineares.

Capítulo 5

CONTROLE PREDITIVO BASEADO EM MODELO

Preâmbulo

A estratégia de controle preditivo baseado em modelo (MPC) é, segundo vários autores, a estratégia de controle multi-variável mais empregada na indústria de processos. Devido à sua importância e grande aplicabilidade em processos da indústria química, esta estratégia de controle foi utilizada para controlar os processos estudados nesta tese.

Apresenta-se na seção 5.1.1 uma breve revisão sobre algumas das várias versões disponíveis de algoritmos MPC, incluindo desde as versões mais simples, baseadas em modelos de resposta ao degrau do processo que não consideram as restrições operacionais, às mais complexas que utilizam modelos não lineares, incorporam as restrições diretamente no projeto do controlador e resolvem o problema de otimização não linear, sujeito a restrições, iterativamente a cada instante de amostragem. Na seção 5.1.2 são apresentados o conceito da estratégia de controle preditivo, seus elementos básicos e a idéia do horizonte móvel. A seção 5.2 discute o projeto de controladores preditivos baseados em modelo lineares, dando ênfase ao *Controle por Matriz Dinâmica* - DMC - que ainda é o algoritmo de controle preditivo mais utilizado na indústria. As formulações mono-variáveis (SISO) e multi-variáveis (MIMO) para este controlador são apresentadas na seção 5.2.2. O controle preditivo baseado em modelos não lineares (NMPC) é

apresentado na seção 5.3. Após uma breve introdução justificando a abordagem não linear para este tipo de controlador, as três classes básicas de NMPCs são apresentadas na seção 5.3.2. Finalmente, são apresentadas a estrutura geral da estratégia NMPC adotada nesta tese (figura 5.5) e a formulação básica do algoritmo de controle não linear.

5.1 Introdução

O conceito básico da estratégia de controle preditivo 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 (*set-point*). A abordagem de controle MPC foi desenvolvida para tratar de problemas de controle multi-variáveis complexos onde existam interações significativas entre as entradas (variáveis manipuladas) e as saídas (variáveis controladas). Uma grande vantagem da estratégia MPC é a sua capacidade de incorporar restrições nas variáveis de processo diretamente no projeto do controlador (BEQUETTE, 1991; QIN e BADGWELL, 1997; SEBORG, 1999).

Nos algoritmos MPCs que consideram restrições, o cálculo das ações de controle consiste na resolução, em tempo real, de um problema de programação linear ou quadrática em cada instante de amostragem, assim que novas medidas do processo estejam disponíveis. Deve-se atentar para o fato de que o método de otimização deve ser robusto o suficiente para fornecer uma solução no intervalo de tempo entre duas amostragens. Segundo QIN e BADGWELL (1997), o ponto de operação de uma planta química típica que leva ao ótimo econômico, geralmente faz intersecção com as restrições operacionais. Assim sendo, o algoritmo preditivo também deve considerar as restrições das variáveis operacionais. Um controlador industrial de sucesso é aquele que mantém o sistema o mais próximo possível das restrições, sem que elas sejam violadas. Além dos problemas relacionados com as restrições, os processos químicos geralmente são não lineares, e multi-variáveis, configurando-se em sistemas cujo comportamento dinâmico varia com o tempo. Este tipo de cenário levou ao desenvolvimento de uma metodologia genérica

5.1 Introdução

capaz de projetar um controlador baseado em modelo para processos multi-variáveis e com restrições operacionais. O ponto central desta metodologia é que a ação de controle é determinada através da solução, em tempo real, de um problema de otimização dinâmico em cada instante de amostragem.

A identificação do modelo é, segundo SEBORG (1999), um passo fundamental para a implementação da estratégia MPC. Geralmente, os modelos são desenvolvidos a partir de testes experimentais em malha aberta. Em princípio, a abordagem MPC pode ser implementada a partir de uma grande variedade de modelos de processo: fenomenológicos ou empíricos, lineares ou não lineares, estáticos ou dinâmicos, etc. A grande maioria das aplicações industriais utilizam modelos dinâmicos de tempo discreto na forma de resposta ao degrau ou impulso. No entanto, modelos na forma de função de transferência e na representação de espaço de estados também podem ser empregados. Os controladores MPC baseados em modelos na forma de função de transferência, conhecidos como *Controle Preditivo Generalizado - GPC*, é um dos algoritmos que têm recebido grande atenção em aplicações industriais.

5.1.1 Histórico da Estratégia MPC

O *Controle Preditivo Baseado em Modelo* (MPC) é, segundo SEBORG (1999), a estratégia de controle multi-variável mais aplicada na indústria de processos. Os primeiros algoritmos do tipo MPC foram desenvolvidos por dois grupos (RICHALET *et al.*, 1976; CUTLER e RAMAKER, 1979), independentemente, no final da década de 1970. A técnica desenvolvida por RICHALET *et al.* (1976) foi comercializada com o nome *IDCOM* e a idéia de CUTLER e RAMAKER (1979) deu origem a um dos algoritmos de controle preditivo mais empregados na indústria química, o *DMC* ou Controle por Matriz Dinâmica¹ (QIN e BADGWELL, 1997). Estes algoritmos, que são MPCs baseados em modelos lineares, foram muitos usados durante os últimos 20 anos na solução de problemas de controle multi-variável, principalmente em refinarias de petróleo e plantas petroquímicas (SEBORG, 1999). No final da década de 1980, CLARKE *et al.* (1987a; 1987b) desenvolveram o algoritmo do Controle Preditivo Generalizado (GPC), cuja base teórica parece, a primeira vista, ser bastante similar aos conceitos do DMC. Contudo,

¹*Dynamic Matrix Control.*

existem diferenças significativas entre eles. Enquanto o DMC foi concebido para tratar de problemas de controle multi-variáveis típicos das refinarias e indústrias químicas, o desenvolvimento do GPC teve como objetivo oferecer uma nova alternativa para o controle estocástico pois, em sua maioria, este tipo de controlador utilizava modelos do tipo entrada/saída (funções de transferência) (MORARI e LEE, 1999). Enquanto no GPC, os aspectos estocásticos desempenham um papel fundamental, a formulação original do DMC era completamente determinística, baseada em um modelo no domínio do tempo (modelo de resposta ao impulso ou ao degrau) e não incluía nenhum tipo de modelo para a perturbação. Segundo MORARI e LEE (1999), o GPC não é indicado para problemas multi-variáveis com restrições que, por sua vez, são encontrados com muita frequência nas indústrias químicas.

Atualmente, boa parte dos algoritmos MPC são formulados na forma de espaço de estados (MORARI e LEE, 1999). Assim, o sistema a ser controlado é descrito por um modelo linear discreto do tipo:

$$x(k+1) = Ax(k) + Bu(k), \quad x(0) = x_0 \quad (5.1)$$

onde $x(k) \in \mathbb{R}^n$ e $u(k) \in \mathbb{R}^m$ denotam o estado e o sinal de controle do sistema, respectivamente. A implementação da estratégia chamada de *Horizonte Móvel*² é formulada através da introdução do seguinte problema de otimização em malha aberta (GARCIA *et al.*, 1989):

$$J_{(N_y, N_u)}(x_0) = \min_{u(\cdot)} \left[\sum_{i=0}^{N_y-1} x^T(i)x(i) + \sum_{i=1}^{N_u-1} u^T(i)u(i) \right] \quad (5.2)$$

sujeito a

$$Ex + Fu \leq \psi \quad (5.3)$$

²*Receding Horizon.*

5.1 Introdução

onde N_y denota o horizonte de predição e N_u representa o horizonte de controle. As equações (5.1) a (5.3) definem um problema quadrático que pode ser resolvido numericamente por vários algoritmos de otimização.

Seja $u_{(N_y, N_u)}^*(i \mid x(k))$ com $i = 0, \dots, N_u - 1$, a seqüência minimizante de controle para $J_{(N_y, N_u)}(x(k))$, sujeita à dinâmica do sistema (equação (5.1)) e às restrições do sistema (equação (5.3)). A estratégia do horizonte móvel é aplicada de forma que somente a primeira ação de controle, $u_{(N_y, N_u)}^*(0 \mid x(k))$, é implementada para obter $x(k+1) = Ax(k) + Bu_{(N_y, N_u)}^*(0 \mid x(k))$. O resto da seqüência de controle, $u_{(N_y, N_u)}^*(i \mid x(k))$ é descartada e $x(k+1)$ é utilizado para atualizar o problema de otimização (equação (5.2)) como uma nova condição inicial. Este processo é repetido a cada instante de amostragem, utilizando somente a primeira ação de controle para obter uma nova condição inicial. A função de custo é então deslocada um passo de amostragem adiante e o processo é reiniciado. Este procedimento deu origem ao termo *horizonte móvel*.

Os controladores preditivos não lineares (NMPC) são baseados na mesma idéia do horizonte móvel, com a diferença de que os modelos utilizados para descrever o comportamento dinâmico do sistema são não lineares. Vários modelos não lineares têm sido testados no projeto de NMPC, dentre os quais destacam-se as redes neurais artificiais, modelos *fuzzy*, modelos de Wiener e de Volterra, equações diferenciais, equações algébrico-diferenciais, etc. (MORARI e LEE, 1999). BEQUETTE (1991) apresenta uma extensa revisão sobre diferentes métodos de solução para os problemas não lineares dos NMPCs e suas aplicações a processos da indústria química. QIN e BADGWELL (1998) também descrevem alguns dos NMPCs mais utilizados na indústria e discutem alguns aspectos relacionados com a modelagem, controle, otimização e implementação destes controladores. MORARI e LEE (1999) fazem uma análise da evolução da estratégia de controle preditivo, desde a sua origem até as perspectivas e necessidades futuras dos MPC.

Existe uma vasta literatura, entre artigos e livros, dedicada à teoria e à aplicação das várias vertentes relacionadas ao controle preditivo baseado em modelo. Os artigos fundamentais de RICHalet *et al.* (1976) e CUTLER e RAMAKER (1979) são, sem dúvida, o melhor ponto de partida. A estratégia do controle preditivo generalizado é apresentada de forma bastante elucidativa nos artigos originais de CLARKE *et al.*

(1987a; 1987b). Existem algumas publicações mais abrangentes, do tipo revisão (*review*), que são bastante úteis para situar todas as tendências e o estado da arte do MPC, dentre as quais, destacam-se os trabalhos de GARCIA *et al.* (1989), QIN e BADGWELL (1997; 1998), HENSON (1998), MORARI e LEE (1999), SEBORG (1999) e RAWLINGS (2000). Para discussões mais aprofundadas, os livros de SOETERBOEK (1992), CLARKE (1994), MORARI *et al.* (1994) e CAMACHO e BORDONS (1999) constituem excelentes referências.

5.1.2 A Estratégia de Controle Preditivo

A figura 5.1 ilustra o princípio geral do controle preditivo. No instante de amostragem k , o algoritmo de controle recebe informações sobre o estado atual do sistema e, baseado nesta informação e no modelo do processo, calcula as ações de controle futuras para que a saída do processo siga as referências. A trajetória de entradas futuras é calculada otimizando algum critério e, no algoritmo preditivo, somente a primeira ação de controle é implementada até a próxima amostragem, quando todo o processo é repetido baseado nas novas informações medidas.

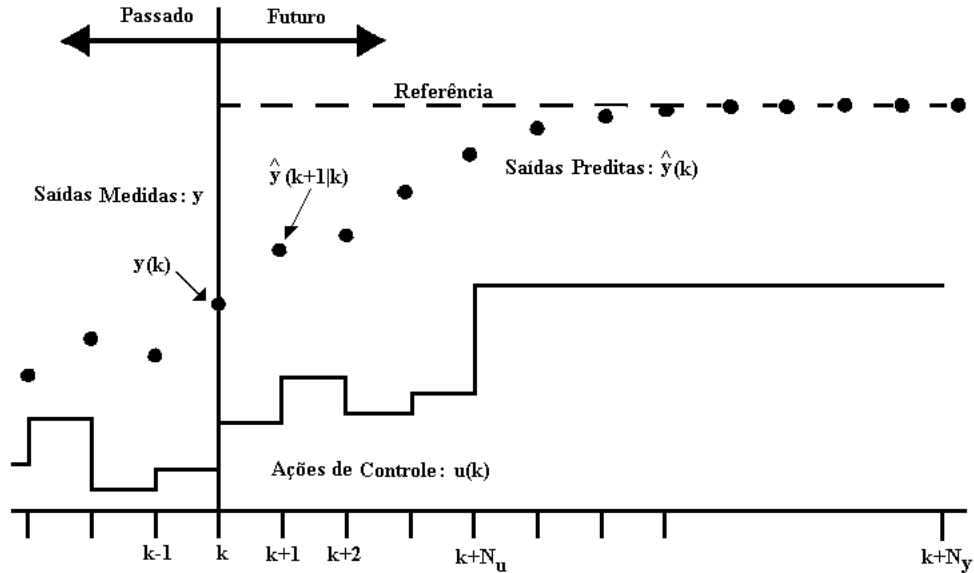


Figura 5.1: Representação esquemática dos elementos básicos do controle preditivo.

Embora a figura 5.1 represente a essência da estratégia de controle preditivo,

5.1 Introdução

esta abordagem pode ser melhor compreendida separando-a em duas partes. As figuras 5.2 e 5.3 representam o modo como os controladores preditivos atuam sobre um sistema SISO. As escalas de tempo localizadas no fundo destas figuras representam a escala de tempo absoluto, enquanto as escalas de tempo correspondentes às abscissas de cada gráfico, representam escalas de tempo relativas ao instante de amostragem, k , que denota o presente (SOETERBOEK, 1992). Desse modo, as figuras 5.2 e 5.3 estão dispostas em ordem cronológica.

As variáveis $u(k)$, $y(k)$ e $w(k)$ são definidas como sendo a saída do controlador (ação de controle), saída do processo e saída desejada do processo (*set-point*) no instante de amostragem k , respectivamente. Os seguintes vetores podem então ser definidos:

$$\begin{aligned}\mathbf{u} &= [u(k), \dots, u(k + N_y - 1)]^T \\ \hat{\mathbf{y}} &= [\hat{y}(k + 1), \dots, \hat{y}(k + N_y)]^T \\ \mathbf{w} &= [w(k + 1), \dots, w(k + N_y)]^T\end{aligned}$$

onde N_y é o horizonte de predição e o símbolo $\hat{}$ denota uma variável estimada.

Considere inicialmente a figura 5.2, onde o tempo de amostragem atual, k , corresponde ao tempo absoluto t . O controlador preditivo calcula a seqüência de ações de controle futura, \mathbf{u} , para que a saída predita do processo, $\hat{\mathbf{y}}$, se aproxime da saída desejada³ \mathbf{w} . Contudo, ao invés de se utilizar toda a seqüência de ações de controle, \mathbf{u} , calculada pelo algoritmo de otimização para controlar o processo nos próximos N_y instantes de amostragem, somente o primeiro elemento desta seqüência, $u(k)$, é utilizado para controlar o processo. No próximo instante de amostragem (figura 5.3), todo o procedimento é repetido utilizando as últimas informações medidas do processo. Esta estratégia é conhecida como *horizonte móvel*. Assumindo que não há perturbações nem erros de modelagem, a saída do processo, $\hat{y}(k + 1)$, predita no tempo t é exatamente igual à saída do processo, $y(k)$, medida no tempo $t + 1$. Outra seqüência de ações de controle é então calculada de modo que a saída predita do processo se aproxime da saída de referência e assim sucessivamente (SOETERBOEK, 1992).

A principal razão para a utilização do horizonte móvel é a possibilidade que esta

³A saída desejada do processo, ou trajetória de referência, pode ser uma seqüência arbitrária de pontos ou, mais freqüentemente, a resposta de um modelo de primeira ordem.

estratégia oferece para a correção de erros de modelagem e/ou devido a perturbações futuras. Assim, se a saída *predita* do processo no tempo t , $\hat{y}(k+1)$, não for igual à saída *medida* do processo no tempo $t+1$, $y(k)$, esta diferença será corrigida através de um mecanismo *feedback*. Além disso, no tempo $t+1$, as próximas predições serão feitas a partir da saída medida do processo e não a partir da saída predita no instante anterior. O resultado da aplicação da abordagem do horizonte móvel é que o horizonte sobre o qual a saída do processo é predita é deslocado em um instante de amostragem na direção do futuro, a cada instante de amostragem (SOETERBOEK, 1992).

Uma maneira de se determinar a qualidade da predição fornecida pelo modelo do processo, ou seja, o quanto a saída predita se aproxima da saída desejada, é através de um critério que seja função de $\hat{\mathbf{y}}$, \mathbf{w} e \mathbf{u} . Uma função geralmente utilizada é do tipo:

$$J = \sum_{i=1}^{N_y} (\hat{y}(k+i) - w(k+i))^2 \quad (5.4)$$

e a seqüência ótima de ações de controle, \mathbf{u}_{opt} , sobre o horizonte de predição é obtida através da minimização de J em relação a \mathbf{u} :

$$\mathbf{u}_{opt} = \arg \min_{\mathbf{u}} J \quad (5.5)$$

Assim, a seqüência \mathbf{u}_{opt} é ótima em relação à função de custo que é minimizada e, conseqüentemente, o erro de predição é minimizado (SOETERBOEK, 1992).

O cálculo da seqüência de ações de controle é um problema de otimização ou, mais especificamente, de minimização, cuja solução geralmente requer um procedimento iterativo. Contudo, quando o critério é quadrático, o modelo é linear e não existem restrições, a solução é analítica.

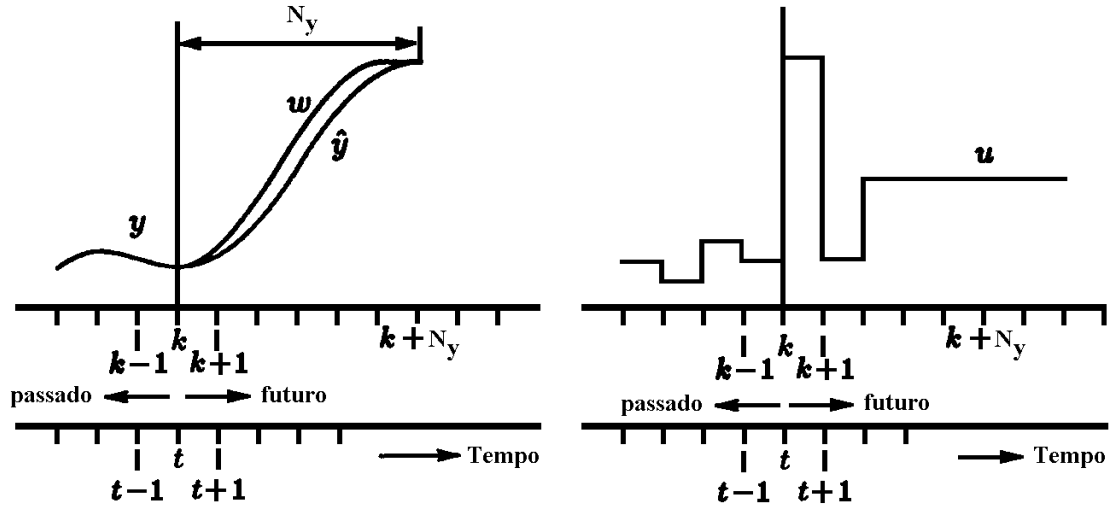


Figura 5.2: Representação do princípio do horizonte móvel: Situação no tempo t .

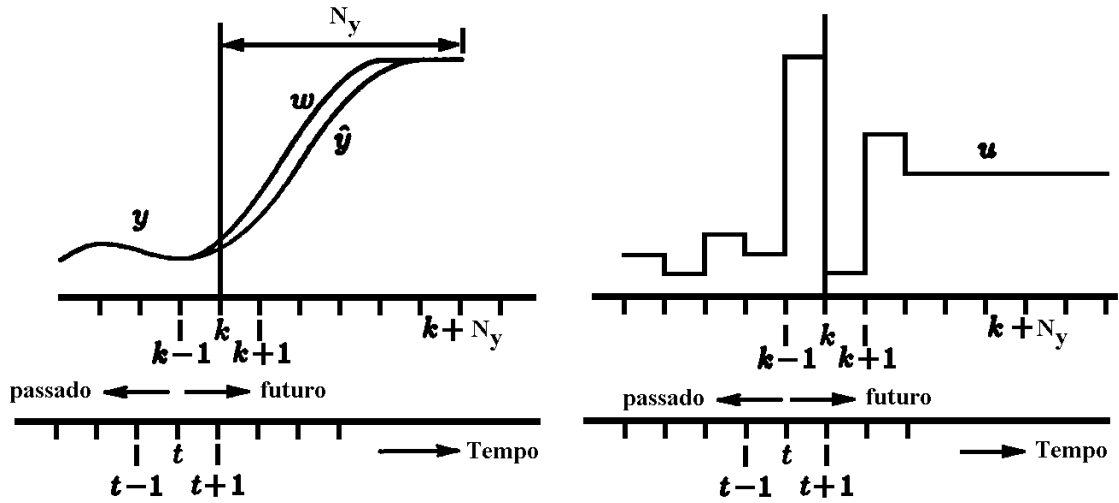


Figura 5.3: Representação do princípio do horizonte móvel: Situação no tempo $t + 1$.

5.2 Controle Preditivo Linear

5.2.1 Introdução

A maioria dos controladores MPCs utilizados na indústria são baseados em modelos lineares do tipo entrada-saída que, por sua vez, são identificados utilizando dados de testes feitos diretamente sobre a planta. O procedimento de identificação e, particularmente, o projeto dos testes realizados nas plantas são etapas cruciais para o bom desempenho dos algoritmos MPC (QIN e BADGWELL, 1997).

5.2.1.1 Modelos Lineares

A essência da estratégia MPC é otimizar o desempenho da planta através da manipulação de suas entradas, tendo como base as previsões do comportamento dinâmico do processo fornecido pelo modelo. Assim, o modelo é um elemento essencial de um controlador MPC e, embora a ação *feedback* possa corrigir alguns erros de modelagem, o uso de modelos que forneçam previsões futuras inadequadas pode comprometer o desempenho do controlador MPC (RAWLINGS, 2000).

Historicamente, no início das aplicações industriais dos MPCs, utilizavam-se modelos no domínio do tempo, do tipo entrada-saída e resposta ao impulso ou degrau. Parte do apelo inicial dos MPCs, principalmente no ambiente industrial, foi devido ao fácil entendimento deste tipo de modelos. Já no ambiente acadêmico, a formulação dos modelos lineares no espaço de estados tornou-se mais comum. A representação das plantas através de modelos com representação no espaço de estados apresenta várias vantagens, tais como a facilidade para a generalização para sistemas multi-variáveis e para a análise de suas propriedades em malha fechada. Além disso, vários resultados associados à teoria dos sistemas lineares, como por exemplo o Regulador Quadrático Linear (LQR), Filtro de Kalman (KF), Controle com Modelo Interno (IMC), etc., tornam-se imediatamente acessíveis para uso no projeto do MPC (RAWLINGS, 2000).

Um modelo linear de um processo sujeito a restrições pode ser representado na forma de espaço de estados como segue:

5.2 Controle Preditivo Linear

$$x(k+1) = Ax(k) + Bu(k), \quad x(0) = x_0 \quad (5.6)$$

$$y(k) = Cx(k) \quad (5.7)$$

$$Du(k) \leq d \quad (5.8)$$

$$Hx(k) \leq h \quad (5.9)$$

onde D e H são as matrizes de restrições e d e h são vetores com componentes positivos.

5.2.2 MPCs Baseados em Modelos Lineares

Conforme mencionado anteriormente, o controlador DMC foi um dos algoritmos preditivo implantados com muito sucesso em vários processos da indústria química e é, ainda hoje, um dos controladores preditivos mais utilizados industrialmente. Por este motivo, decidiu-se utilizá-lo nesta tese como referência para a avaliação do desempenho dos controladores preditivos não lineares propostos.

A estratégia empregada pelo *Controle por Matriz Dinâmica* - DMC, assim como a sua formulação básica, serão apresentados a seguir.

5.2.2.1 O Controle Por Matriz Dinâmica - DMC

Engenheiros da Cia. de Petróleo Shell desenvolveram, de forma independente, sua própria estratégia de controle MPC, cuja aplicação inicial data de 1973. CUTLER e RAMAKER (1979) apresentaram os detalhes de um algoritmo de controle multi-variável sem restrições, denominado *Dynamic Matrix Control* (DMC) durante o encontro anual da sociedade americana de engenharia química - *AIChE* - em 1979 e posteriormente

durante o *Joint Automatic Control Conference*, em 1980 (CUTLER e RAMAKER, 1980). Desde então, várias modificações foram feitas no algoritmo original para lhe conferir habilidades para lidar principalmente com restrições e não linearidades.

As principais características do algoritmo de controle DMC incluem (QIN e BADGWELL, 1997):

- Modelo linear da planta baseado na resposta ao degrau;
- Minimização de uma função objetivo de desempenho quadrático sobre um horizonte de predição finito;
- Comportamento futuro da saída da planta especificado de modo a seguir a saída de referência o mais próximo possível;

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. Matematicamente, a resposta ao degrau pode ser definida como a integral da resposta ao impulso, assim, dada uma das formas, a outra pode ser facilmente obtida. Através do modelo de resposta ao degrau, pode-se prever as mudanças futuras das saídas do processo como uma combinação linear das variações nas entradas. Este procedimento é ilustrado na figura 5.4 para o caso SISO.

Em cada instante de amostragem, o valor da resposta ao degrau da variável de saída é medido e é chamado de coeficiente de resposta ao degrau a_i . A diferença entre dois coeficientes de resposta ao degrau sucessivos é chamado de coeficiente de resposta ao impulso e é dado por (NEWELL e LEE, 1989):

$$h_i = a_i - a_{i-1} \quad (5.10)$$

O valor predito da variável de saída, \hat{y} , no instante de amostragem k , pode ser expresso da seguinte forma:

5.2 Controle Preditivo Linear

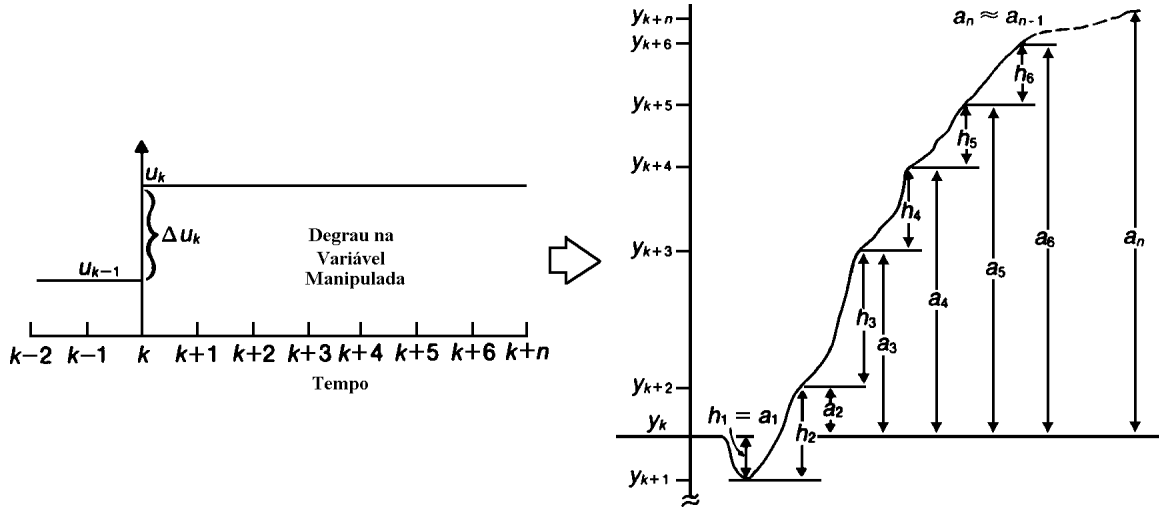


Figura 5.4: Representação esquemática do modelo de convolução e seus coeficientes: Resposta do sistema a um degrau na variável de entrada.

$$\hat{y}_k = \sum_{i=1}^{N_m} h_i u_{k-i} \quad (5.11)$$

onde N_m é o horizonte de modelo, que corresponde ao número de coeficientes da resposta ao impulso necessários para representar 90-95% da resposta estacionária do sistema.

5.2.2.2 O Algoritmo DMC para Sistemas SISO

O algoritmo DMC para sistemas mono-variáveis utiliza o modelo descrito pela equação (5.11). Reescrevendo-a para o instante de amostragem $k + 1$, tem-se:

$$\hat{y}_{k+1} = \sum_{i=1}^{N_m} h_i u_{k+1-i} \quad (5.12)$$

Subtraindo a equação (5.11) da equação (5.12), obtém-se:

$$\hat{y}_{k+1} - \hat{y}_k = \sum_{i=1}^{N_m} h_i (u_{k+1-i} - u_{k-i}) \quad (5.13)$$

Definindo $\Delta u_k = u_k - u_{k-1}$, a equação (5.13) pode ser reescrita:

$$\hat{y}_{k+1} = \hat{y}_k + \sum_{i=1}^{N_m} h_i \Delta u_{k+1-i} \quad (5.14)$$

O modelo de convolução (equação (5.14)) pode ser estendido para considerar um horizonte de predição de N_y instantes futuros:

$$\hat{y}_{k+j} = \hat{y}_{k+j-1} + \sum_{i=1}^{N_m} h_i \Delta u_{k+1-i} \quad \text{para } j = 1, \dots, N_y \quad (5.15)$$

O algoritmo DMC utiliza uma estratégia de realimentação para corrigir os erros de predição do modelo dado pela equação (5.15). O valor predito da saída do processo no instante de amostragem k , é comparado com o valor medido desta mesma variável. O desvio calculado entre os dois valores é utilizado para corrigir as futuras predições. Tomando como exemplo o instante de amostragem $k + 1$, tem-se:

$$y_{k+1}^c = \hat{y}_{k+1} + (y_k - \hat{y}_k) \quad (5.16)$$

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

Generalizando para os N_y instantes futuros:

$$y_{k+j}^c = \hat{y}_{k+j} + (y_{k+j-1}^c - \hat{y}_{k+j-1}) \quad \text{para } j = 1, \dots, N_y \quad (5.17)$$

Para o instante $k+j$ com $j = 1$, verifica-se que o termo y_{k+j-1}^c da equação (5.17) corresponde ao termo y_k da equação (5.16), ou seja, o valor medido da saída. Para $j = 2$, a equação (5.17) fica:

$$\begin{aligned} y_{k+2}^c &= \hat{y}_{k+2} + (y_{k+1}^c - \hat{y}_{k+1}) \\ &= \hat{y}_{k+2} + (\hat{y}_{k+1} + (y_k - \hat{y}_k) - \hat{y}_{k+1}) \\ &= \hat{y}_{k+2} + (y_k - \hat{y}_k) \end{aligned} \quad (5.18)$$

5.2 Controle Preditivo Linear

Para o instante $k + j$ com $j = N_y$, obtém-se:

$$y_{k+N_y}^c = \hat{y}_{k+N_y} + (y_k - \hat{y}_k) \quad (5.19)$$

Pode-se observar que as predições para os instantes futuros de $k + 1$ a $k + N_y$, feitas no instante atual k , são corrigidas com o 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 (5.15) e (5.17), chega-se a:

$$y_{k+j}^c = y_{k+j-1}^c + \sum_{i=1}^{N_m} h_i \Delta u_{k+j-i} \quad \text{para } j = 1, \dots, N_y \quad (5.20)$$

Para o instante $k + j$ com $j = 1$, a equação (5.20) fica:

$$\begin{aligned} y_{k+1}^c &= y_k^c + \sum_{i=1}^{N_m} h_i \Delta u_{k+1-i} \\ &= y_k + \sum_{i=1}^{N_m} h_i \Delta u_{k+1-i} \\ &= y_k + h_1 \Delta u_k + h_2 \Delta u_{k-1} + \dots + h_{N_m} \Delta u_{k+1-N_m} \end{aligned} \quad (5.21)$$

Uma vez que os valores de $\Delta u_{k-1}, \Delta u_{k-2}, \dots, \Delta u_{k-N_m}$ são conhecidos, pode-se agrupá-los em um único termo da seguinte forma:

$$S_1 = \sum_{i=2}^{N_m} h_i \Delta u_{k+1-i} \quad (5.22)$$

que, por sua vez, pode ser utilizado para reescrever a equação (5.21):

$$y_{k+1}^c = y_k + h_1 \Delta u_k + S_1 \quad (5.23)$$

Repetindo-se o processo para o instante $k + j$ com $j = 2$, obtém-se:

$$y_{k+2}^c = y_{k+1}^c + h_1 \Delta u_{k+1} + h_2 \Delta u_k + \dots + h_{N_m} \Delta u_{k+2-N_m} \quad (5.24)$$

Agrupando novamente os valores conhecidos em um único termo:

$$y_{k+2}^c = y_{k+1}^c + h_1 \Delta u_{k+1} + h_2 \Delta u_k + S_2 \quad (5.25)$$

onde $S_2 = \sum_{i=3}^{N_m} h_i \Delta u_{k+2-i}$

Substituindo a equação (5.23) na equação (5.25), obtém-se:

$$y_{k+2}^c = y_k + (h_1 + h_2) \Delta u_k + h_1 \Delta u_{k+1} + S_1 + S_2 \quad (5.26)$$

Definindo-se:

$$a_i = \sum_{j=1}^i h_j \quad \text{para } i = 1, \dots, N_y \quad (5.27)$$

$$S_m = \sum_{i=m+1}^{N_m} h_i \Delta u_{k+m-i} \quad \text{para } m = 1, \dots, N_y \quad (5.28)$$

$$P_i = \sum_{m=1}^i S_m \quad \text{para } i = 1, \dots, N_y \quad (5.29)$$

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

$$y_{k+j}^c = y_k + a_j \Delta u_k + a_{j-1} \Delta u_{k+1} + \dots + a_1 \Delta u_{k+j-1} + P_j \quad (5.30)$$

A equação (5.30) pode ser escrita na forma matricial:

5.2 Controle Preditivo Linear

$$\begin{aligned}
 \begin{bmatrix} y_{k+1}^c \\ y_{k+2}^c \\ \vdots \\ y_{k+N_y-1}^c \\ y_{k+N_y}^c \end{bmatrix} &= \begin{bmatrix} a_1 & 0 & \cdots & 0 & 0 \\ a_2 & a_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{N_y-1} & a_{N_y-2} & \cdots & a_1 & 0 \\ a_{N_y} & a_{N_y-1} & \cdots & a_2 & a_1 \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+N_y-2} \\ \Delta u_{k+N_y-1} \end{bmatrix} + \\
 &+ \begin{bmatrix} y_k + P_1 \\ y_k + P_2 \\ \vdots \\ y_k + P_{N_y-1} \\ y_k + P_{N_y} \end{bmatrix}
 \end{aligned} \tag{5.31}$$

ou, de modo mais compacto:

$$(\mathbf{y}^c)_{N_y \times 1} = (\mathbf{M})_{N_y \times N_y} (\Delta \mathbf{u})_{N_y \times 1} + (\mathbf{y}^P)_{N_y \times 1} \tag{5.32}$$

onde $\mathbf{M}(N_y \times N_y)$ é a matriz triangular e $\mathbf{y}^P(N_y \times 1)$ é o vetor de predições dados pela equação (5.31). Tais predições são feitas sem considerar ações de controle nos intervalos de amostragem de 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 (*resposta livre*).

Se definirmos a trajetória desejada, \mathbf{y}^d , igual ao valor do *set-point*, \mathbf{y}_{sp} , obtém-se:

$$\begin{bmatrix} y_{k+1}^d \\ y_{k+2}^d \\ \vdots \\ y_{k+N_y-1}^d \\ y_{k+N_y}^d \end{bmatrix} = \begin{bmatrix} y_{sp_k} \\ y_{sp_k} \\ \vdots \\ y_{sp_k} \\ y_{sp_k} \end{bmatrix} \tag{5.33}$$

Subtraindo a equação (5.31) da equação (5.33) e definindo $e_k = y_{sp_k} - y_k$, obtém-se:

$$\mathbf{E} = -\mathbf{M}\Delta\mathbf{u} + \mathbf{E}' \quad (5.34)$$

onde

$$\mathbf{E} = \begin{bmatrix} y_{k+1}^d - y_{k+1}^c \\ y_{k+2}^d - y_{k+2}^c \\ \vdots \\ y_{k+N_y-1}^d - y_{k+N_y-1}^c \\ y_{k+N_y}^d - y_{k+N_y}^c \end{bmatrix} \quad \mathbf{E}' = \begin{bmatrix} e_k - P_1 \\ e_k - P_2 \\ \vdots \\ e_k - P_{N_y-1} \\ e_k - P_{N_y} \end{bmatrix} \quad (5.35)$$

Se é requerido que a saída predita seja igual à desejada, então $\mathbf{E} = 0$ e, conseqüentemente

$$\Delta\mathbf{u} = \mathbf{M}^{-1}\mathbf{E}' \quad (5.36)$$

A solução fornecida pela equação (5.36) 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. A estratégia DMC consiste em obter um sistema indeterminado, reduzindo arbitrariamente a dimensão do vetor $\Delta\mathbf{u}$ de acordo com o tamanho do horizonte de controle. Para tanto, considera-se que $\Delta u_{k+j} = 0$ para $j \geq N_u$, sendo que $N_u < N_y$ (PINTO, 1990).

Assim, a equação (5.31) é escrita da seguinte forma:

5.2 Controle Preditivo Linear

$$\begin{bmatrix} y_{k+1}^c \\ y_{k+2}^c \\ \vdots \\ y_{k+N_y-1}^c \\ y_{k+N_y}^c \end{bmatrix} = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{N_y-1} & a_{N_y-2} & \cdots & a_{N_y-N_u} \\ a_{N_y} & a_{N_y-1} & \cdots & a_{N_y-N_u+1} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+N_u-2} \\ \Delta u_{k+N_u-1} \end{bmatrix} + \begin{bmatrix} y_k + P_1 \\ y_k + P_2 \\ \vdots \\ y_k + P_{N_y-1} \\ y_k + P_{N_y} \end{bmatrix} \quad (5.37)$$

ou, na forma compacta:

$$(\mathbf{y}^c)_{N_y \times 1} = (\mathbf{A})_{N_y \times N_u} (\Delta \mathbf{u})_{N_u \times 1} + (\mathbf{y}^P)_{N_y \times 1} \quad (5.38)$$

A matriz $\mathbf{A}(N_y \times N_u)$, chamada de *Matriz Dinâmica*, relaciona as entradas com as saídas do processo e através dela pode-se calcular analiticamente (através da solução de um problema de mínimos quadrados) o vetor ótimo de ações de controle (QIN e BADGWELL, 1997). Note-se que a matriz dinâmica \mathbf{A} é formada pelas primeiras N_u colunas da matriz \mathbf{M} .

O sistema, que não apresenta uma solução única, deve ser resolvido segundo um critério de otimização. O algoritmo DMC calcula o vetor $\Delta \mathbf{u}$ minimizando o seguinte critério de desempenho:

$$J = \mathbf{E}^T \mathbf{E} \quad (5.39)$$

cujas solução é dada por:

$$\Delta \mathbf{u} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{E}' \quad (5.40)$$

A solução da equação (5.40) configura um problema de otimização sem restrições e, como consequência, a lei de controle pode resultar em ações de controle muito bruscas. Para evitar este tipo de problema, define-se um índice de desempenho modificado que incorpora restrições nos movimentos da variável manipulada, resultando em:

$$J = (\mathbf{E}^T \mathbf{E}) + (\mathbf{\Lambda} \Delta \mathbf{u})^T (\mathbf{\Lambda} \Delta \mathbf{u}) \quad (5.41)$$

onde $\mathbf{\Lambda}(N_u \times N_u)$ é uma matriz diagonal (*Matriz de Ponderação*) que permite introduzir penalizações nas ações de controle. Os elementos da matriz de ponderação são dados por:

$$\begin{cases} \lambda_{ij} = f & \text{se } i = j \\ \lambda_{ij} = 0 & \text{se } i \neq j \end{cases} \quad (5.42)$$

onde f é o fator de ponderação.

A lei de controle resultante da função objetivo descrita pela equação (5.41) é dada por (CUTLER e RAMAKER, 1979):

$$\Delta \mathbf{u} = (\mathbf{A}^T \mathbf{A} + \mathbf{\Lambda}^T \mathbf{\Lambda})^{-1} \mathbf{A}^T \mathbf{E}' \quad (5.43)$$

Embora a equação (5.43) forneça as ações de controle para todo o horizonte de controle (de k a $k + N_u$), somente a primeira ação de controle, Δu_k , é implementada. Assim, o valor da variável manipulada no instante k é calculada da seguinte forma:

$$u_k = u_{k-1} + \Delta u_k \quad (5.44)$$

A cada instante de amostragem, a solução fornecida pela equação (5.43) é calculada novamente e um novo valor para a variável manipulada (5.44) é então determinado.

5.2 Controle Preditivo Linear

5.2.2.3 O Algoritmo DMC para Sistemas MIMO

Apresenta-se a seguir a representação do algoritmo DMC para o caso multi-variável. Neste caso, todas as entradas influenciam todas as saídas do processo, resultando em sistemas com muitas interações.

Para um processo com V_m variáveis manipuladas e V_c variáveis controladas, a representação do modelo de convolução é dada por:

$$\begin{aligned}
 \hat{y}_{1,k} &= \sum_{i=1}^{N_m} h_{1,1,i} u_{1,k-i} + \sum_{i=1}^{N_m} h_{1,2,i} u_{2,k-i} + \cdots + \sum_{i=1}^{N_m} h_{1,V_m,i} u_{V_m,k-i} \\
 \hat{y}_{2,k} &= \sum_{i=1}^{N_m} h_{2,1,i} u_{1,k-i} + \sum_{i=1}^{N_m} h_{2,2,i} u_{2,k-i} + \cdots + \sum_{i=1}^{N_m} h_{2,V_m,i} u_{V_m,k-i} \\
 \vdots &= \vdots \\
 \hat{y}_{V_c,k} &= \sum_{i=1}^{N_m} h_{V_c,1,i} u_{1,k-i} + \sum_{i=1}^{N_m} h_{V_c,2,i} u_{2,k-i} + \cdots + \sum_{i=1}^{N_m} h_{V_c,V_m,i} u_{V_m,k-i}
 \end{aligned} \tag{5.45}$$

que em notação compacta fica:

$$\hat{\mathbf{y}}_k = \sum_{i=1}^{N_m} \mathbf{H}_i \mathbf{u}_{k-i} \tag{5.46}$$

De forma análoga à equação (5.14):

$$\hat{\mathbf{y}}_{k+1} = \hat{\mathbf{y}}_k + \sum_{i=1}^{N_m} \mathbf{H}_i \mathbf{u}_{k+1-i} \tag{5.47}$$

O modelo de convolução dado pela equação (5.47) pode ser estendido para N_y instantes futuros:

$$\hat{\mathbf{y}}_{k+j} = \hat{\mathbf{y}}_{k+j-1} + \sum_{i=1}^{N_m} \mathbf{H}_i \mathbf{u}_{k+j-i} \quad \text{para } j = 1, \dots, N_y \tag{5.48}$$

Corrigindo as predições de forma análoga ao procedimento descrito pela equação (5.17):

$$\mathbf{y}_{k+j}^c = \hat{\mathbf{y}}_{k+j} + (\mathbf{y}_{k+j-1}^c - \hat{\mathbf{y}}_{k+j-1}) \quad \text{para } j = 1, \dots, N_y \quad (5.49)$$

onde, no instante $k + 1$, \mathbf{y}_{k+j-1}^c é igual ao valor medido \mathbf{y}_k .

Substituindo a equação (5.45) na equação (5.46), obtém-se:

$$\mathbf{y}_{k+j}^c = \mathbf{y}_{k+j-1}^c + \sum_{i=1}^{N_m} \mathbf{H}_i \Delta \mathbf{u}_{k+j-i} \quad \text{para } j = 1, \dots, N_y \quad (5.50)$$

Repetindo o procedimento descrito para os sistemas mono-variáveis, obtém-se a forma multi-variável da equação (5.37):

$$\begin{aligned} \begin{bmatrix} \mathbf{y}_{k+1}^c \\ \mathbf{y}_{k+2}^c \\ \vdots \\ \mathbf{y}_{k+N_y-1}^c \\ \mathbf{y}_{k+N_y}^c \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_1 & 0 & \cdots & 0 \\ \mathbf{A}_2 & \mathbf{A}_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{N_y-1} & \mathbf{A}_{N_y-2} & \cdots & \mathbf{A}_{N_y-N_u} \\ \mathbf{A}_{N_y} & \mathbf{A}_{N_y-1} & \cdots & \mathbf{A}_{N_y-N_u+1} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_k \\ \Delta \mathbf{u}_{k+1} \\ \vdots \\ \Delta \mathbf{u}_{k+N_u-2} \\ \Delta \mathbf{u}_{k+N_u-1} \end{bmatrix} + \\ &+ \begin{bmatrix} \mathbf{y}_k + \mathbf{P}_1 \\ \mathbf{y}_k + \mathbf{P}_2 \\ \vdots \\ \mathbf{y}_k + \mathbf{P}_{N_y-1} \\ \mathbf{y}_k + \mathbf{P}_{N_y} \end{bmatrix} \end{aligned} \quad (5.51)$$

onde

$$\mathbf{A}_i = \sum_{j=1}^i \mathbf{H}_j \quad \text{para } i = 1, \dots, N_y \quad (5.52)$$

5.2 Controle Preditivo Linear

$$\mathbf{S}_m = \sum_{i=m+1}^{N_m} \mathbf{H}_i \Delta \mathbf{u}_{k+m-i} \quad \text{para } m = 1, \dots, N_y \quad (5.53)$$

$$\mathbf{P}_i = \sum_{m=1}^i \mathbf{S}_m \quad \text{para } i = 1, \dots, N_y \quad (5.54)$$

Reescrevendo a equação (5.51) na forma compacta:

$$(\mathbf{y}^c)_{N_y \cdot V_c \times 1} = (\mathbf{M})_{N_y \cdot V_c \times N_u \cdot V_m} (\Delta \mathbf{u})_{N_u \cdot V_m \times 1} + (\mathbf{y}^P)_{N_y \cdot V_c \times 1} \quad (5.55)$$

A versão multi-variável da equação (5.34) é escrita de forma similar:

$$\mathbf{E} = -\mathbf{M} \Delta \mathbf{u} + \mathbf{E}' \quad (5.56)$$

contudo, \mathbf{M} e $\Delta \mathbf{u}$ são definidos na equação (5.55) e \mathbf{E} e \mathbf{E}' são dados por:

$$\mathbf{E} = \begin{bmatrix} \mathbf{y}_{k+1}^d - \mathbf{y}_{k+1}^c \\ \mathbf{y}_{k+2}^d - \mathbf{y}_{k+2}^c \\ \vdots \\ \mathbf{y}_{k+N_y-1}^d - \mathbf{y}_{k+N_y-1}^c \\ \mathbf{y}_{k+N_y}^d - \mathbf{y}_{k+N_y}^c \end{bmatrix} \quad \mathbf{E}' = \begin{bmatrix} \mathbf{e}_k - \mathbf{P}_1 \\ \mathbf{e}_k - \mathbf{P}_2 \\ \vdots \\ \mathbf{e}_k - \mathbf{P}_{N_y-1} \\ \mathbf{e}_k - \mathbf{P}_{N_y} \end{bmatrix} \quad (5.57)$$

onde $\mathbf{e}_k = \mathbf{y}_{sp_k} - \mathbf{y}_k$.

Como no caso SISO, o algoritmo DMC calcula o vetor $\Delta \mathbf{u}$ para minimizar o seguinte critério de desempenho:

$$J = (\mathbf{E}^T \mathbf{E}) + (\Lambda \Delta \mathbf{u})^T (\Lambda \Delta \mathbf{u}) \quad (5.58)$$

cuja lei de controle resultante da minimização da função objetivo descrita pela equação (5.58) é :

$$\Delta \mathbf{u} = (\mathbf{A}^T \mathbf{A} + \mathbf{\Lambda}^T \mathbf{\Lambda})^{-1} \mathbf{A}^T \mathbf{E}' \quad (5.59)$$

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

$$\mathbf{u}_k = \mathbf{u}_{k-1} + \Delta \mathbf{u}_k \quad (5.60)$$

5.3 Controle Preditivo Não Linear

5.3.1 Introdução

Segundo HENSON e SEBORG (1997), o controle de sistemas não lineares vem recebendo grande atenção, tanto da indústria como da universidade, desde de o início da década de 90. Este interesse pelo projeto de controladores não lineares deve-se a vários fatores e, segundo estes autores, o principal é o desempenho inadequado apresentado pelos controladores lineares quando utilizados em sistemas fortemente não lineares ou mesmo a sistemas moderadamente não lineares que operam em uma faixa ampla de condições. Por outro lado, tem havido um progresso significativo no desenvolvimento de novas estratégias de identificação e de projeto de controladores baseados em modelo para sistemas não lineares. Estas técnicas empregam o modelo não linear diretamente no algoritmo de cálculo do controlador, sem que seja necessário realizar uma linearização local em torno do ponto operacional.

O desenvolvimento de estratégias de identificação e controle voltadas para sistemas não lineares também tem sido motivada pelas seguintes razões (HENSON e SEBORG, 1997):

- Avanços na teoria de sistemas não lineares que possibilitaram o projeto de controladores aplicáveis a uma grande classe de problemas de controle não lineares.
- O desenvolvimento de métodos de identificação eficientes para modelos não lineares empíricos.
- A contínua melhoria na capacidade dos *softwares* e *hardwares* voltados para controle através de computador, tornando possível incorporar modelos não lineares complexos nos sistemas de controle dos processos.

O sucesso dos controladores preditivos baseados em modelos lineares motivou a extensão desta metodologia para os problemas de controle não lineares. A 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. As ações de controle são calculadas através da solução de um problema de programação não linear

em cada instante de amostragem. Esta abordagem é genericamente chamada de *Controle Preditivo Baseado em Modelo Não Linear* - NMPC - ou simplesmente *Controle Preditivo Não Linear*. Contudo, deve-se estar atento para o fato de que os potenciais benefícios propiciados pela abordagem não linear devem justificar o inerente aumento de complexidade (computacional, inclusive) quando comparado aos métodos lineares tradicionais.

Um aspecto muito importante na estratégia NMPC é o tipo de modelo do processo que deve ser usado. Os modelos determinísticos têm um grande apelo por causa da sua capacidade de interpretação física e por sua aplicabilidade em grandes faixas de condições operacionais. Contudo, estes modelos nem sempre estão disponíveis por conta do grande esforço de modelagem envolvido, além da necessidade de um profundo conhecimento sobre o processo. Outra desvantagem desta abordagem é que tais modelos freqüentemente contêm parâmetros e/ou variáveis de processo que não podem ser medidos. Nesses casos, deve-se dispor de um método de estimação de estados eficiente.

Uma abordagem alternativa, cuja importância vem crescendo sistematicamente, é o projeto de sistemas de controle não linear utilizando modelos empíricos ou semi-empíricos, desenvolvidos a partir de dados experimentais. Recentemente, o desenvolvimento de modelos dinâmicos não lineares a partir de dados entrada-saída tem sido foco de vários estudos, onde várias técnicas de modelagem não linear têm sido propostas. A técnica predominante para a obtenção de tais modelos e mesmo para o desenvolvimento de estratégias de controle não lineares tem sido a rede neural artificial (HENSON e SEBORG, 1997).

5.3.2 MPCs Baseados em Modelos Não Lineares

O uso de modelos não lineares na estratégia de controle preditivo tem atraído um grande número de pesquisadores nos últimos anos (RAWLINGS, 2000). O principal obstáculo na abordagem NMPC é a necessidade de resolver um problema de otimização dinâmica não linear em cada instante de amostragem.

Existem basicamente três diferentes classes de NMPCs, que podem ser agrupadas como segue:

5.3 Controle Preditivo Não Linear

1. *Extensões Não Lineares do Controle por Matriz Dinâmica - DMC e do Controle Quadrático por Matriz Dinâmica - QDMC*: Existem várias contribuições nesta área, cujas abordagens diferem umas das outras basicamente em função do modelo ou do método de otimização utilizado. Algumas das abordagens mais representativas são listadas a seguir. GARCIA e MORSHEDI (1986) propuseram o Controle Quadrático Não Linear por Matriz Dinâmica (NLQDMC) que utiliza um modelo não linear para calcular o efeito das variáveis manipuladas passadas sobre as saídas preditas. Em cada instante de amostragem, obtém-se um modelo linear do processo através da linearização do modelo não linear em torno daquele ponto de operação e este modelo é então utilizado para calcular as ações de controle. Este método apresenta a vantagem que somente um programa quadrático deve ser resolvido em cada instante de amostragem. GATTU e ZAFIRIOU (1992; 1995) propuseram uma extensão do algoritmo NLQDMC incorporando a estimação de estados utilizando o ganho estacionário do filtro de Kalman a cada instante de amostragem. PETERSON *et al.* (1992) modificaram o algoritmo DMC clássico estendendo o vetor de perturbações para considerar o efeito de não linearidades no horizonte de predição. Esta modificação foi feita utilizando um modelo não linear com formulação no espaço de estados. As ações de controle futuras são determinadas a partir de um procedimento iterativo onde as ações de controle são calculadas através do algoritmo DMC clássico e um novo vetor de perturbações é calculado utilizando-se o modelo não linear. A matriz dinâmica utilizada pelo algoritmo DMC pode ser calculada através da linearização do modelo não linear em torno do estado estacionário ou através da aplicação de um degrau no próprio modelo não linear. Este algoritmo apresenta como vantagens a facilidade de adaptação para a versão NLQDMC (onde as ações de controle podem ser calculadas como solução de um problema quadrático a cada instante de amostragem) e a prova das condições de convergência do procedimento iterativo de otimização.
2. *Algoritmos NMPC com Critérios de Otimização Baseados no Método de Newton*: As abordagens mais representativas desta classe de algoritmos foram propostas por BIEGLER e colaboradores (LI e BIEGLER, 1989; OLIVEIRA e BIEGLER, 1995), cuja idéia principal é linearizar o modelo não linear, na representação de espaço de estados, em torno de uma trajetória de referência determinada pela seqüência

de entradas calculada no instante de amostragem anterior. Obtêm-se modelos de resposta ao impulso ou degrau, a partir do modelo no espaço de estados linearizado e variante no tempo. Uma nova seqüência de entradas é calculada sobre o horizonte de predição através da solução de um novo problema quadrático (resolvido em cada instante de amostragem), seguido de uma busca uni-dimensional onde o critério de otimização quadrático é calculado baseado no modelo não linear. O método de otimização proposto por estes pesquisadores pode ser considerado como uma variação da estratégia de programação quadrática sucessiva (SQP)⁴.

3. *Algoritmos NMPC com Critérios de Otimização Baseados em Técnicas de Programação Não Linear*: Esta abordagem combina técnicas de programação não linear com modelos não lineares, geralmente descritos na forma de espaço de estados, para a resolução do problema de otimização. As principais diferenças são o modo como as equações do modelo são resolvidas e o método de otimização utilizado. Os algoritmos SQP são os mais utilizados para resolver este tipo de problema de programação não linear e, embora estes algoritmos exijam mais esforço computacional quando comparado às outras técnicas não lineares descritas anteriormente, o modelo não linear pode ser utilizado diretamente no algoritmo de otimização, sem que haja a necessidade de linearização (QIN e BADGWELL, 1998).

A experiência (positiva) acumulada no projeto de controladores preditivos baseados em modelos lineares do tipo entrada-saída, sugerem que a versão não linear destes modelos são fortes candidatos para o desenvolvimento de NMPCs. Os tipos de modelos entrada-saída não lineares estudados com mais freqüência para aplicações em estratégias de controle preditivo não linear são:

- Redes neurais artificiais;
- Modelos nebulosos;
- Modelos de Volterra;
- Modelos polinomiais do tipo NARX.

Embora existam diferenças nas representações dos modelos não lineares, todos os NMPCs desta classe são muito similares em relação à formulação e à solução do problema de otimização.

⁴*Successive Quadratic Programming.*

5.3 Controle Preditivo Não Linear

Os controladores preditivos não lineares propostos nesta tese foram projetados baseados na abordagem NMPC correspondente ao item 3. Foram considerados três tipos diferentes de modelos neurais⁵ (ou modelos matematicamente equivalentes aos modelos neurais) e a solução do problema de otimização - formulada de forma idêntica para os três casos - foi obtida através do algoritmo SQP (EDGAR e HIMMELBLAU, 1988). A estrutura básica deste tipo de controlador está representada na figura 5.5.

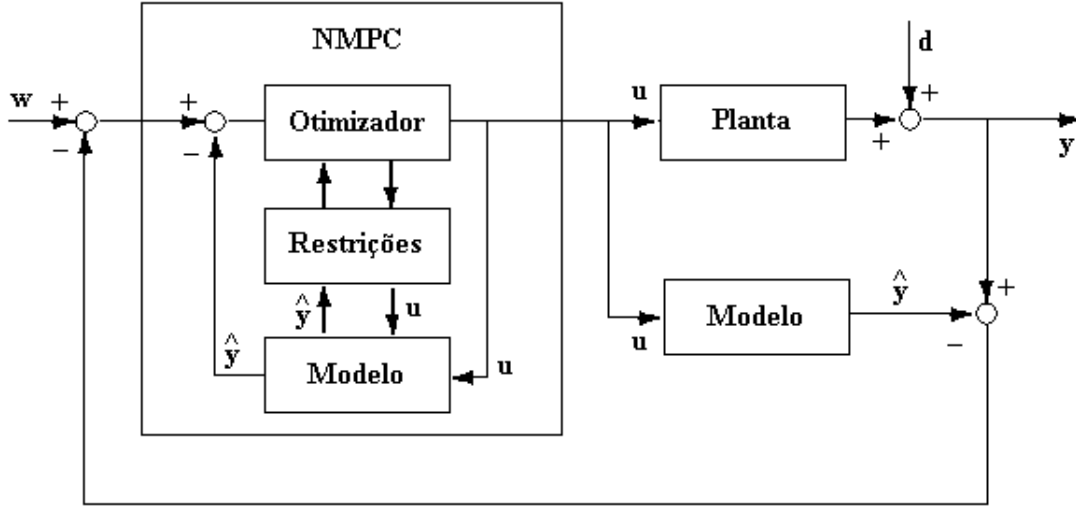


Figura 5.5: Estrutura do controlador preditivo não linear utilizado nos capítulos 6 e 7.

5.3.2.1 Formulação Geral do Controlador Preditivo Não Linear

Considere um processo multi-variável com V_c variáveis controladas e V_m variáveis manipuladas. O objetivo de um controlador do tipo NMPC, é calcular um conjunto de ações de controle futuras de forma a minimizar a seguinte função objetivo:

$$J = \sum_{j=1}^{V_c} \sum_{i=N_1}^{N_y} (\hat{y}_{j,k+i}^c - w_{j,k+i})^2 + \sum_{j=1}^{V_m} \lambda_j \sum_{i=1}^{N_u} (\Delta u_{j,k+i-1})^2 \quad (5.61)$$

onde λ é o vetor com os fatores de ponderação (parâmetros de ajuste que penalizam as ações de controle), $\Delta \mathbf{u}$ é o vetor contendo os incrementos nas variáveis manipuladas,

⁵Vale lembrar que a generalização para outros tipos de modelos não lineares é obtida facilmente.

N_1 é o horizonte inicial, N_y é o horizonte de predição, N_u é o horizonte de controle, \mathbf{w} é o vetor com as trajetórias de referência (*set-points*) para cada variável de saída e $\hat{\mathbf{y}}^c$ é o vetor contendo as previsões corrigidas dos modelos não lineares. Estas previsões são obtidas recursivamente até um total de N_y previsões futuras para cada variável controlada, V_c , e corrigidas da seguinte forma:

$$\hat{\mathbf{y}}_{k+i}^c = \hat{\mathbf{y}}_{k+i} + \mathbf{d}_k, \quad (i = 1, \dots, N_y) \quad (5.62)$$

Na equação (5.62), $\hat{\mathbf{y}}_{k+i}$ são as previsões realizadas utilizando o modelo não linear para as saídas controladas do processo no instante de amostragem futuro, $k + i$, e \mathbf{d}_k são os termos de correção dados por:

$$\mathbf{d}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k \quad (5.63)$$

onde \mathbf{y}_k são as saídas medidas do processo no instante de amostragem presente e $\hat{\mathbf{y}}_k$ são as respectivas previsões do modelo (calculadas no instante de amostragem anterior).

O algoritmo de otimização calcula os incrementos, $\Delta \mathbf{u}$, para as variáveis manipuladas minimizando a função objetivo (5.61). As ações de controle futuras são então derivadas a partir dos incrementos ótimos de controle:

$$\mathbf{u}_{k+i} = \mathbf{u}_{k+i-1} + \Delta \mathbf{u}_{k+i}, \quad (i = 0, \dots, N_u - 1) \quad (5.64)$$

É importante observar que somente as primeiras N_u ações de controle são otimizadas, enquanto que as demais são mantidas constantes, ou seja:

$$\mathbf{u}_{k+i} = \mathbf{u}_{k+N_u-1}, \quad (i = N_u, \dots, N_y - 1) \quad (5.65)$$

O problema de otimização também está sujeito às seguintes restrições:

$$\mathbf{y}_{min} \leq \hat{\mathbf{y}}_{k+i} \leq \mathbf{y}_{max}, \quad (i = 1, \dots, N_y) \quad (5.66)$$

5.4 Considerações Finais Sobre o Capítulo

$$\mathbf{u}_{min} \leq \mathbf{u}_{k+i-1} \leq \mathbf{u}_{max}, \quad (i = 1, \dots, N_u) \quad (5.67)$$

$$|\mathbf{u}_{k+i-1} - \mathbf{u}_{k+i-2}| \leq \Delta \mathbf{u}_{max}, \quad (i = 1, \dots, N_u) \quad (5.68)$$

Finalmente, aplica-se a estratégia do “Horizonte Móvel”⁶ (GARCIA *et al.*, 1989; SOETERBOEK, 1992; CAMACHO e BORDONS, 1999), onde somente a primeira ação de controle (para cada variável manipulada), \mathbf{u}_k , é implementada e o problema de otimização é resolvido novamente em cada instante de amostragem.

5.4 Considerações Finais Sobre o Capítulo

O objetivo deste capítulo foi realizar uma breve descrição sobre os controladores preditivos baseados em modelo e apresentar a estratégia de controle utilizada nesta tese. O controle preditivo baseado em modelo é a abordagem empregada com maior frequência e sucesso nos processos da indústria química. Os principais motivos que fizeram deste tipo de controlador o preferido na maioria das aplicações industriais, foram discutidos e justificados ao longo deste capítulo. A literatura relata que, apesar da disponibilidade de modelos não lineares e da comprovada capacidade de algumas técnicas de otimização para lidar com problemas de programação não linear, os algoritmos preditivos baseados em modelos e/ou problemas de otimização lineares ainda são os mais utilizados. A proposta desta tese de utilizar a abordagem de identificação e controle não lineares, pretende contribuir para a análise que deve ser feita sobre o custo do desenvolvimento deste tipo de controlador e o ganho em desempenho que esta abordagem pode oferecer.

⁶*Receding Horizon.*

Capítulo 6

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

Preâmbulo

Neste capítulo serão apresentados alguns resultados da aplicação das técnicas não lineares de identificação e controle propostas nesta tese. A seção 6.1 destaca a importância dos processos biotecnológicos na sociedade contemporânea. Também são apresentados nesta seção um breve histórico sobre a evolução do processo de produção do etanol associado ao contexto sócio-econômico característico de cada década, desde sua origem nos velhos engenhos de açúcar até os dias atuais. Alguns aspectos relacionados com o desenvolvimento de novas técnicas de produção também são destacados, assim como os fatos mais importantes que fizeram com que o etanol ora fosse encarado como um produto estratégico para o desenvolvimento nacional e ora fosse relegado à condição de sub-produto da indústria açucareira. O processo fermentativo em questão, representado por um simulador desenvolvido a partir da modelagem fenomenológica de uma planta real em escala industrial, é apresentado na seção 6.2. Nesta seção é feita a descrição detalhada da planta, das variáveis de interesse e suas faixas de operação e a formulação do problema de controle (mono-variável). Ainda na seção 6.2 é apresentado o equacionamento matemático que descreve o comportamento dinâmico do processo.

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

As estratégias de identificação empregadas neste caso de estudo são discutidas na seção 6.3, onde os procedimentos que envolvem a determinação das variáveis fundamentais do modelo, a geração dos dados para identificação do processo, a escolha do período de amostragem adequado, a determinação da estrutura dos modelos utilizados para identificar o processo e as etapas de estimação e validação dos modelos selecionados são explorados. O procedimento de identificação do processo fermentativo é ilustrado utilizando-se três modelos não lineares (Hierárquico Neuro Nebuloso - HNN, Neural - MLP e Neural Construtivo - UGL) e um modelo linear do tipo ARX. Finalmente o controle do processo fermentativo é apresentado na seção 6.4. A estratégia de controle preditivo baseado em modelo (MPC) foi escolhida devido ao sucesso destes algoritmos em problemas de grande porte, especialmente em processos da indústria química. Os modelos HNN, MLP e UGL identificados na seção 6.3, foram utilizados no projeto de controladores MPC não lineares. Os controladores resultantes (MPC-HNN, MPC-MLP e MPC-UGL) foram testados em várias condições operacionais e os seus desempenhos foram comparados entre si e com o desempenho de um controlador bastante utilizado na indústria química, o controlador DMC.

6.1 Introdução

A Biotecnologia vem se tornando cada vez mais presente e importante nas atividades da sociedade contemporânea por ser uma tecnologia “limpa” e segura quando comparada aos processos químicos tradicionais. Além disso, os seus produtos são caracterizados por apresentarem alto valor agregado e serem extremamente úteis para vários segmentos produtivos tais como as indústrias de fármacos, alimentos e combustíveis. Os processos biotecnológicos são caracterizados por apresentarem comportamento dinâmico complexo como, por exemplo, resposta inversa, tempo morto, atraso nos instrumentos de medida, parâmetros que variam ao longo do tempo e fortes não linearidades. Estas características se devem principalmente ao fato dos microorganismos (que são a principal força motriz destes processos) serem extremamente sensíveis a variações nas condições do meio fermentativo tais como temperatura, concentrações de substrato e produto, variações de pH, etc. Por motivos como estes, a modelagem, simulação e controle de processos biotecnológicos configuram um desafio ainda não totalmente resolvido e um tema de pesquisa bastante oportuno.

Os processos de fermentação alcoólica brasileiros surgiram nos velhos engenhos de açúcar como processos (rudimentares) de produção de aguardente a partir do caldo de cana-de-açúcar. A produção de etanol por via fermentativa em escala industrial teve início no século XIX, onde as primeiras indústrias surgiram vinculadas às usinas de fabricação de açúcar. O desafio na época era transformar o melaço, um sub-produto sem grande valor comercial, em outro produto com maior valor agregado, o etanol (ANDRIETTA, 1994).

Já na década de 1930, o processo fermentação alcoólica foi aprimorado na França através da produção de álcool em regime de batelada alimentada (processo MELLE-BOINOT), contudo o processo clássico de produção - em regime de batelada - continuou a vigorar no Brasil até o início dos anos 60. Os principais fatores para este atraso tecnológico, segundo ANDRIETTA (1994), foram a falta de incentivos por parte do governo, a dificuldade na obtenção de equipamentos adequados e o pequeno interesse pela produção de etanol por parte dos produtores, fruto de um mercado interno pequeno e da forte concorrência internacional. O processo de modernização da indústria alcooleira teve início na década de 1960 mas somente em meados da década de 1970 é

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

que a maior parte das destilarias brasileiras passaram a operar em regime de batelada alimentada. Este salto tecnológico proporcionou grandes ganhos no rendimento e na produtividade do processo fermentativo que, através de estudos de otimização, alcançou seu máximo de eficiência nos anos 80.

O grande impulso para a utilização do etanol como combustível alternativo ocorreu durante a crise do petróleo, no início da década de 1970. Nesta época, o etanol despontava como uma alternativa promissora para os países carentes em reservas de petróleo (ou capacidade de refino) e com grandes extensões territoriais (ANDRIETTA e STUPIELLO, 1990a). O primeiro programa de grande porte destinado a desenvolver a tecnologia necessária para a utilização do etanol como uma alternativa viável para o combustível fóssil teve início no Brasil, com a criação do Programa Nacional do Álcool, o ProÁlcool, em 1975. Programas similares foram criados nos Estados Unidos em 1978 e posteriormente no Canadá (WHEALS *et al.*, 1999).

O aumento da produção e do uso de etanol como combustível no Brasil foi possível graças a três ações governamentais: A primeira foi a garantia de compra de quantidades pré-determinadas de etanol pela companhia estatal nacional de petróleo, a PETROBRAS; a segunda foi a determinação de incentivos econômicos para projetos agro-industriais para a produção de etanol, através de baixas taxas de juros no período 1980/1985 e aproximadamente US\$ 2 bilhões (valores históricos) em empréstimos, representando 29% do total dos investimentos necessários para a instalação da atual capacidade de produção; e a terceira foi a atuação diretamente no mercado consumidor, através da venda de álcool combustível nos postos de abastecimento por 59% do preço da gasolina. Isto só foi possível porque o preço da gasolina no Brasil era estabelecido pelo Governo, a um valor aproximadamente igual ao dobro do preço comercializado nos Estados Unidos da América (MOREIRA e GOLDEMBERG, 2002).

A abertura de linhas de crédito especiais, aliado a uma política de preços favoráveis, impulsionaram o desenvolvimento do setor alcooleiro nas décadas de 70 e 80. Os esforços empreendidos pelos empresários e pesquisadores deste setor foram direcionados principalmente para *i)* Otimizar o processo fermentativo em regime de batelada alimentada, responsável por cerca de 70% da produção nacional até meados da década de 1990; e *ii)* Desenvolver e implementar o processo de fermentação contínua em escala industrial (ANDRIETTA e STUPIELLO, 1990a; ANDRIETTA e STUPIELLO, 1990b; WHE-

6.1 Introdução

ALS *et al.*, 1999).

Embora a produtividade das plantas contínuas seja comprovadamente maior do que a de suas correspondentes descontínuas e semi-contínuas, houve uma resistência inicial relativamente grande à implantação do processo contínuo nas destilarias brasileiras. Segundo ANDRIETTA (1994), os principais motivos para essa aparente incoerência foram a resistência natural a inovações tecnológicas e o relato de alguns problemas técnicos na implantação das primeiras plantas contínuas (causados pela adaptação inadequada de plantas semi-contínuas já existentes). Somente mais tarde, depois do esgotamento da capacidade produtiva das plantas semi-contínuas é que houve uma retomada de esforços no sentido de um novo salto tecnológico. Incentivados pelo aumento de eficiência proporcionado pela implantação de processos contínuos de fermentação alcoólica, vários pesquisadores intensificaram os estudos visando o desenvolvimento destes processos, que são caracterizados pela alta produtividade (RODRIGUES *et al.*, 1992; ANDRIETTA *et al.*, 1994; ANDRIETTA e MAUGERI FILHO, 1994).

Apesar do significativo aumento de produtividade proporcionado pelo aprimoramento do processo contínuo de fermentação alcoólica, um estudo apresentado por WHEALS *et al.* (1999) sobre a utilização do etanol como combustível dá conta que até 1999, o Brasil e os Estados Unidos eram os únicos países que produziam grandes quantidades de etanol com esta finalidade.

No Brasil, o etanol chegou a ser pensado como uma fonte alternativa de combustível para a frota auto-motiva nacional e a solução para a dependência da importação de petróleo da indústria brasileira. A análise das características físico-químicas do etanol o classificaram como um excelente combustível auto-motivo pois apresenta as seguintes vantagens: índice de octanagem superior e pressão de vapor inferior aos da gasolina, resultando em menores emissões evaporativas; sua combustão no ar é inferior à da gasolina, o que reduz o número e a severidade de fogo nos veículos; o etanol anidro tem poder calorífico na faixa de 21,2 e 23,4 MJ/l (megajoule por litro), contra 30,1 e 34,9 MJ/l da gasolina. Em relação aos custos, a produção do etanol atingiu valores de cerca de 100 dólares o barril no estágio inicial do Programa ProÁlcool, em 1980, tendo caído rapidamente para metade deste valor em 1990, seguido de um pequeno declínio nos últimos anos. Esta queda no preço do barril de etanol foi devido às economias propiciadas pelo aumento de escala na produção e ao avanço tecnológico do setor

(MOREIRA e GOLDEMBERG, 2002).

As propriedades do etanol como combustível levaram ao desenvolvimento de motores a álcool e motores para a mistura álcool-gasolina no Brasil. Até 1988, os desenvolvimentos foram realizados pela indústria automobilística (GM, Ford, Volkswagen, Fiat) para sistemas carburados; atualmente, encontram-se disponíveis para todos os motores e sistemas com injeção eletrônica. No Brasil o etanol combustível é utilizado de duas maneiras: *i*) misturado à gasolina em proporções que já variaram de 20 a 25% de etanol anidro (a 99,6 °GL¹ e 0,4% de água), formando uma mistura chamada de “gasohol”, cujo principal objetivo é aumentar a octanagem da gasolina; *ii*) Como etanol puro, na forma de etanol hidratado, a 95,5 °GL (GOLDEMBERG e MACEDO, 1994).

Posteriormente, houve no Brasil uma mudança de prioridades e a produção da frota auto-motiva movida a álcool passou de cerca de 95% na década de 1980 (auge do programa ProÁlcool) para menos de 5% atualmente. Hoje a maior motivação para para a utilização do etanol está relacionada a questões ambientais devido à redução das emissões de CO₂ quando comparado aos combustíveis fósseis.

A introdução da mistura gasolina/álcool teve um impacto imediato na qualidade do ar das grandes cidades, em particular na cidade de São Paulo. Inicialmente, aditivos como chumbo tiveram seu uso reduzido na medida em que a quantidade de álcool na gasolina aumentava, até serem totalmente eliminados em 1991. Os hidrocarbonetos aromáticos tais como o benzeno, que são particularmente tóxicos e comprovadamente carcinogênicos, foram eliminados e o conteúdo de enxofre da gasolina reduzido. Nos automóveis movidos a etanol puro a emissão de enxofre foi eliminada trazendo um benefício duplo pois, devido ao alto teor de enxofre da gasolina brasileira, os carros equipados com catalisadores teriam seus equipamentos contaminados rapidamente. Desse modo, o uso de etanol viabilizou de uma forma indireta a introdução dos catalisadores no Brasil. Além disso, as emissões de monóxido de carbono (CO) foram drasticamente reduzidas: antes de 1980, quando a gasolina era o único combustível em uso, as emissões de CO eram superiores a 50 g/km; em 1995 estas emissões foram reduzidas para menos de 5,8 g/km (MOREIRA e GOLDEMBERG, 2002). Este aspecto é bastante enfatizado quando se procura uma solução plausível para a substituição dos combustíveis fósseis nos países industrializados e para o cumprimento dos compromissos ambientais com

¹Graus Gay-Lussac.

6.1 Introdução

a *United Nations Framework Climate Change Convention* e mais recentemente com o *Protocolo de Kioto*.

As questões econômicas sobre o uso do etanol como combustível voltaram à tona novamente. Considerando a grande economia de divisas com a diminuição da importação de petróleo, fruto da substituição significativa da gasolina pelo etanol, é possível demonstrar que o Programa ProÁlcool foi um meio eficiente de trocar débitos externos por subsídios nacionais. Mesmo com estes ganhos para a sociedade, a continuidade do Programa não está assegurada. Duas possíveis soluções para este problema têm sido discutidas: a expansão interna do uso do etanol e a sua exportação para países industrializados, onde o etanol possa ser utilizado como aditivo para o aumento da octanagem da gasolina e na diminuição de problemas ambientais (MOREIRA e GOLDEMBERG, 2002).

6.2 Descrição do Processo

6.2.1 Discussões Preliminares

O caso de estudo apresentado a seguir deriva da evolução dos processos fermentativos no Brasil. A planta em questão é fruto do trabalho de um grupo de pesquisadores (ANDRIETTA e STUPIELLO, 1990a; ANDRIETTA e STUPIELLO, 1990b; RODRIGUES *et al.*, 1992; ANDRIETTA *et al.*, 1994; ANDRIETTA e MAUGERI FILHO, 1994) que culminou com o trabalho de tese de ANDRIETTA (1994) cujo objetivo foi projetar a unidade de fermentação alcoólica contínua da Usina Guarani (Severínia - SP). Neste momento, onde a produção de etanol já não é tão atrativa quanto no auge da crise do petróleo, a otimização do processo associada a estratégias de controle robustas e eficientes, capazes de aumentar a produtividade sem comprometer o rendimento do processo utilizando o mínimo de insumos, tornou-se uma necessidade premente podendo, inclusive, determinar a viabilidade econômica do processo fermentativo.

Existem várias dificuldades no desenvolvimento de pesquisas quando se trabalha diretamente com uma planta de fermentação durante sua operação industrial. Os maiores inconvenientes se devem aos altos custos envolvidos nas interrupções necessárias na etapa de testes e ajustes, uma vez que o processo fermentativo, apesar de seu grande porte, é apenas uma das partes de uma usina de açúcar e álcool (ver figura 6.1). Sendo assim, optou-se por trabalhar com um simulador do processo cujos parâmetros cinéticos foram determinados e validados experimentalmente por ANDRIETTA (1994) e ANDRIETTA e MAUGERI FILHO (1994). Além disso, a modelagem matemática também utiliza os parâmetros de processo que levam a operação da planta a uma região ótima, previamente determinada e implementada na Usina Guarani.

Considerando as dificuldades envolvidas com a realização de testes diretamente na planta industrial e, ressaltada a importância econômica deste processo para setor produtivo de açúcar e álcool, ficam evidentes a necessidade e a relevância do desenvolvimento de modelos e controladores que possam melhorar o desempenho deste tipo de processo. Os modelos identificados para este processo foram baseados nas abordagens linear, neural e nebulosa e desenvolvidos dando ênfase na sua posterior aplicação em estratégias avançadas de controle baseado em modelo. Os controladores foram pro-

6.2 Descrição do Processo

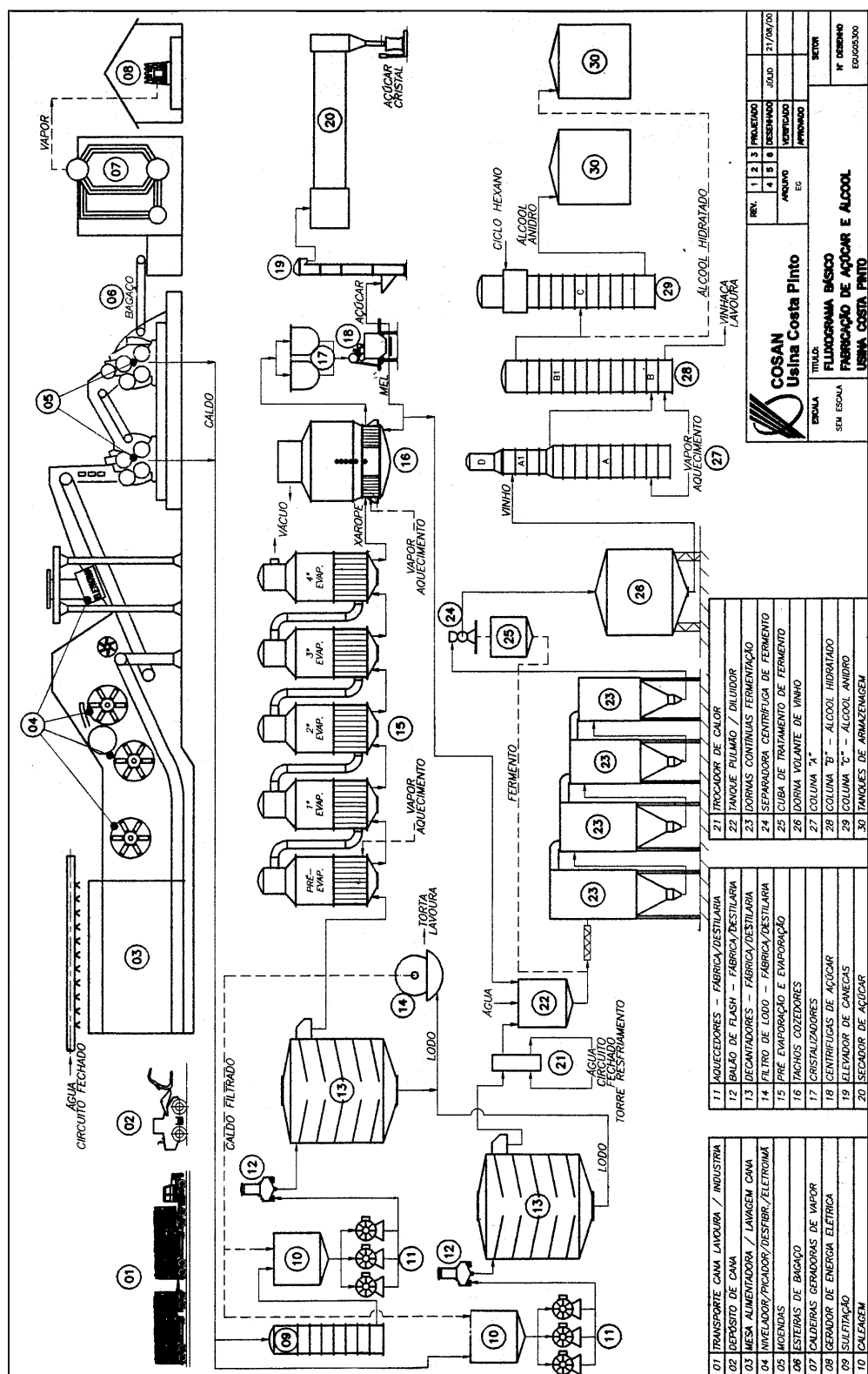


Figura 6.1: Fluxograma básico do processo de produção de açúcar e álcool.

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

jetados para atuar dentro desta região de operação pré-otimizada para atender aos seguintes objetivos: *i)* manter constante a concentração de açúcares redutores totais (ART) na saída da quarta dorna (problema regulador) ou, *ii)* fazer com que o valor desta variável siga uma determinada trajetória estabelecida (problema servo). Deve-se ressaltar que esta variável tem um forte impacto sobre a viabilidade econômica do processo (ANDRIETTA, 1994; DECHECHI, 1998).

O sistema a ser estudado é um típico processo industrial de grande porte (figura 6.2), composto por quatro fermentadores de mistura perfeita (CSTR) ligados em série e operando com reciclo de células. Cada reator possui um sistema externo de troca térmica cujo objetivo é manter a temperatura do meio reacional constante. O processo é alimentado com uma mistura composta de açúcares redutores e fontes de nitrogênio e sais minerais conhecido como mosto que, após a fermentação, é convertido em álcool etílico como produto principal. Após o tratamento ácido efetuado no tanque de diluição, as células de levedura (fermento tratado) são misturadas com o meio de alimentação (mosto) e enviadas novamente aos fermentadores. A fração entre a vazão de fermento tratado e a vazão total de alimentação dos fermentadores é chamada de taxa de reciclo. Após a fermentação ter ocorrido, o vinho fermentado contendo células de levedura (vinho bruto) é enviado para a unidade de separação. Esta unidade é formada por centrífugas de pratos que recebem o vinho bruto, contendo de 30 a 45 kg/m^3 de células, e o separa em dois produtos com densidades distintas. A fase pesada, chamada de leite ou creme de leveduras contendo de 160 a 200 kg/m^3 de células, é enviada para a unidade de tratamento ácido. A fase leve, com concentração máxima permitida de 3 kg/m^3 de células, é chamada de vinho delevedurado e é enviada para o tanque “pulmão” (ou dorna volante) da unidade de destilação. A unidade de tratamento ácido é formada por 2-4 tanques equipados com sistemas de agitação, chamados de cubas. O tratamento é feito de forma contínua e consiste em receber o creme de levedura na primeira cuba, diluí-lo com água na proporção de 1:1 e, em seguida, ajustar o pH do creme diluído para a faixa de 2,0-2,5 com ácido sulfúrico concentrado. Este processo pode ser dividido em três grandes partes:

1. Fermentadores
2. Unidade de separação de células de levedura

6.2 Descrição do Processo

3. Unidade de tratamento ácido

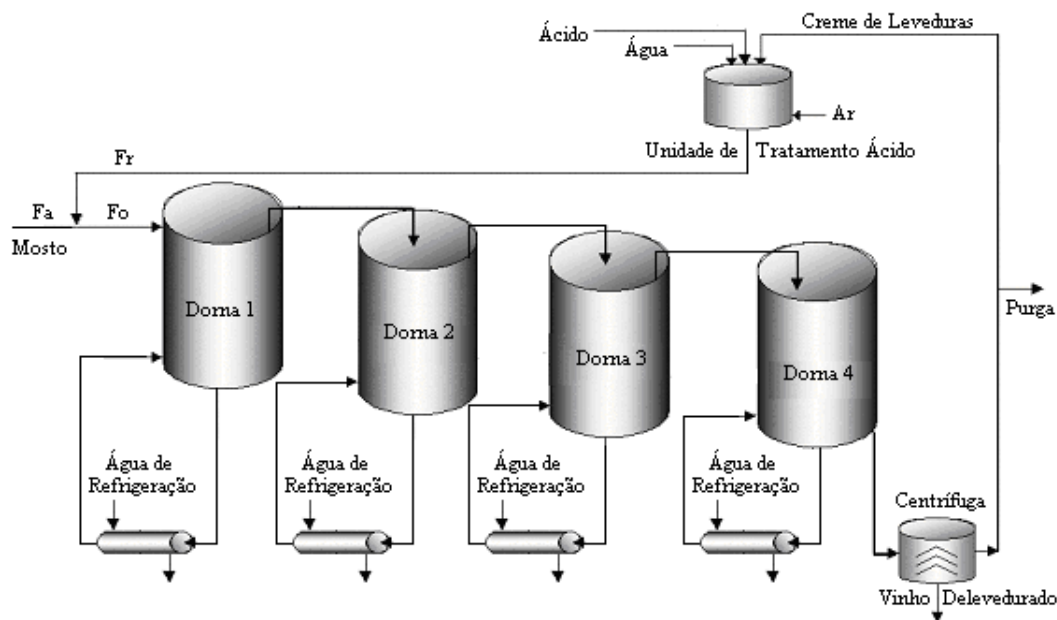


Figura 6.2: Representação esquemática da planta de fermentação alcoólica contínua.

As variáveis de maior importância associadas a este sistema são descritas a seguir:

- Concentração de açúcares redutores totais (ART) no meio de alimentação (mosto): A concentração de ART no meio de alimentação deve ser tal que a concentração de etanol obtida esteja dentro de uma faixa que não prejudique a fermentação nem a destilação. Em geral, recomenda-se que a concentração de etanol esteja entre $65\text{-}70\text{ kg/m}^3$ para que se consiga uma boa produtividade fermentativa. Para que esta faixa de concentração seja atingida, a concentração de ART no meio de alimentação deve estar em torno de 180 kg/m^3 .
- Taxa de Reciclo: Esta variável é definida como a razão entre a vazão de fermento tratado e a vazão total de alimentação do fermentador. O valor utilizado com maior frequência nas unidades industriais (e também adotado para as simulações) é de 30%. Valores muito acima desse limite perturbam demais a dinâmica do sistema além de sobrecarregar as centrífugas que, junto com os trocadores de calor,

são os equipamentos que consomem mais energia (ANDRIETTA e STUPIELLO, 1990b; RODRIGUES *et al.*, 1992; ANDRIETTA, 1994).

- Concentração de Células no Reciclo: É fixada em 90 kg/m^3 na maioria das unidades industriais e está diretamente ligada ao desempenho das unidades separadoras e à quantidade de água adicionada na cuba de tratamento.
- Concentração de Células no Creme de Levedura: Para que se possa manter a concentração de células no reciclo em 90 kg/m^3 e o volume de água recomendado durante o tratamento ácido, a concentração de células no creme de levedura deve ser de 180 kg/m^3 .
- Concentração de Células no Vinho Delevedurado: Devido às condições operacionais das separadoras centrífugas, uma perda de até 3 kg/m^3 de células é considerado normal.
- Temperatura de Operação: Normalmente a temperatura é estabelecida de modo que o volume do reator necessário para se obter a conversão desejada e a área de troca térmica requerida pelo sistema de resfriamento, combinados, forneçam um custo mínimo. Para este caso de estudo, o valor estabelecido foi de $33,5^\circ\text{C}$.
- Temperatura da Água de Resfriamento: Pelo fato desta variável depender de vários fatores, adotou-se a temperatura de 28°C como padrão. Este valor corresponde à temperatura máxima da água captada diretamente de rios e lagoas da região, que é a prática geralmente adotada pela indústria alcooleira.

O estudo do comportamento dinâmico deste processo quando sujeito a testes degrau nas principais variáveis manipuláveis e perturbações foi realizado em trabalhos anteriores (ANDRIETTA, 1994; ASSIS, 1996; DECHECHI, 1998), evidenciando um comportamento fortemente não linear.

6.2.2 Modelagem Matemática

As equações do modelo matemático que descrevem o comportamento dinâmico deste sistema foram desenvolvidas por ANDRIETTA (1994) e são originadas dos balanços

6.2 Descrição do Processo

de massa e energia para os componentes da mistura reacional e do balanço de energia do sistema de troca térmica. Considerou-se que os reatores são de mistura perfeita, o que para este processo é uma boa aproximação. No balanço de massa, desprezou-se a retenção gasosa (*hold-up*), pois na prática este valor é bastante baixo. Desprezou-se também o volume das células (*Saccharomyces cerevisiae*), já que existem estudos na literatura que afirmam que este volume é desprezível para reatores operando com volume de células em torno de 10%, o que corresponde a 30 kg/m³ de massa seca.

6.2.2.1 Balanço de Massa para cada Reator

- Balanço de Massa Global

Assumindo a densidade do líquido reagente constante para cada reator:

$$\frac{dV_i}{dt} = \frac{F_{i-1}\rho_{i-1}}{\rho} - F_i \quad (6.1)$$

onde V é o volume do reator, F é a vazão de alimentação e ρ é a densidade da fase líquida (reagentes e produtos).

- Balanço de Massa para o Substrato

Para o sistema fermentativo no regime transiente, obtém-se:

$$\frac{d(V_i S_i)}{dt} = F_{i-1} S_{i-1} - F_i S_i - \left(\frac{V_i X_i}{Y_{X/S}} \right) \mu_i \quad (6.2)$$

onde S é a concentração de substrato (ART) e X é a concentração de microorganismos (células). $Y_{X/S}$ é o fator de rendimento teórico em biomassa que relaciona a variação da concentração de células com as de substrato (equação 6.3). O parâmetro μ_i (equação 6.4) corresponde ao modelo cinético adotado que é derivado do modelo de MONOD (BAILEY e OLLIS, 1986), onde se assume crescimento balanceado:

$$Y_{X/S} = \frac{dX/dt}{dS/dt} \quad (6.3)$$

$$\mu_i = \mu_{max_i} \left(\frac{S_i}{S_i + K_S} \right) \left(1 - \frac{P_i}{P_{max}} \right)^n \left(1 - \frac{X_i}{X_{max}} \right)^m \quad (6.4)$$

onde μ_{max} é a velocidade máxima de crescimento da levedura, K_S é a constante de afinidade ao substrato, P é a concentração de etanol, P_{max} é a tolerância da levedura ao etanol e X_{max} é a concentração de microorganismos a partir da qual o crescimento celular cessa.

Este modelo é válido para processos contínuos de fermentação alcoólica no estado estacionário ou para processos com dinâmica lenta, como é o caso do sistema estudado aqui. A velocidade máxima de crescimento da levedura tem sua dependência com a temperatura descrita pela equação de ARRHENIUS:

$$\mu_{max} = A \exp \left(\frac{-E}{RT} \right) \quad (6.5)$$

A tolerância da levedura ao etanol, P_{max} , depende da temperatura somente se esta for maior do que $32^\circ C$, a temperatura crítica superior. Nestas situações, a relação é dada por:

$$\begin{aligned} P_{max} &= 103 \quad \text{se } T \leq 32^\circ C \\ P_{max} &= k_0 \exp(aT) \quad \text{se } T > 32^\circ C \end{aligned} \quad (6.6)$$

- Balanço de Massa para o Etanol

$$\frac{d(V_i P_i)}{dt} = F_{i-1} P_{i-1} - F_i P_i + V_i X_i \left(\frac{Y_{P/S}}{Y_{X/S}} \right) \mu_i \quad (6.7)$$

onde $Y_{P/S}$ é o rendimento teórico em etanol, definido como a relação entre as variações de produto e substrato:

$$Y_{P/S} = \frac{dP/dt}{dS/dt} \quad (6.8)$$

6.2 Descrição do Processo

- Balanço de Massa para o Microorganismo

$$\frac{d(V_i X_i)}{dt} = F_{i-1} X_{i-1} - F_i X_i + V_i X_i \mu_i \quad (6.9)$$

A tabela 6.1 mostra os valores dos parâmetros cinéticos utilizados no modelo matemático do processo:

Tabela 6.1: Parâmetros cinéticos do modelo (para *Saccharomyces cerevisiae*).

Parâmetros	Valor
A	$4,50 \cdot 10^{10}$
R	$1,987 \text{ cal/Kmol}$
n	3
m	0,9
E	$1,54 \cdot 10^4 \text{ cal/Kmol}$
k_0	895,6
X_{max}	100 g/l
a	$-0,0676 \text{ l/}^\circ\text{C}$
K_S	1,6 g/l
$Y_{X/S}$	0,033
$Y_{P/S}$	0,445

6.2.2.2 Balanços de Energia

- Balanço de Energia do Fluido Reagente nos Reatores:

$$\frac{dV_i T_i}{dt} = F_{i-1} T_{i-1} - F_i T_i + F c_i (T c_i - T_i) - \left(\frac{V_i \Delta H X_i}{\rho C_p Y_{X/S}} \right) \mu_i \quad (6.10)$$

onde $F c$ é a vazão de fluido reagente nos trocadores de calor, $T c$ é a temperatura do fluido reagente na saída dos trocadores de calor, ΔH é o calor de reação e C_p é a capacidade calorífica do fluido reagente.

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

- Balanço de Energia do Fluido Reagente nos Trocadores de Calor:

Considerando que o volume de fluido reagente, Vc , contido no trocador de calor é constante, tem-se:

$$\frac{dTc_i}{dt} = \frac{Fc_i}{Vc_i}(T_i - Tc_i) - \left(\frac{UA_i}{\rho C_p Vc_i} \right) LMDT_i \quad (6.11)$$

onde U é o coeficiente global de troca térmica, A é a área total de troca térmica e $LMDT$ é a média logarítmica das temperaturas, que é dada por:

$$LMDT_i = \frac{(T_i - Tj_i) - (Tc_i - Tj_e)}{\ln \left(\frac{T_i - Tj_i}{Tc_i - Tj_e} \right)} \quad (6.12)$$

onde Tj é a temperatura do fluido refrigerante na saída do trocador de calor e Tj_e é a temperatura do fluido refrigerante na entrada do trocador de calor.

- Balanço de Energia do Fluido Refrigerante nos Trocadores de Calor:

Considerando que o volume de fluido refrigerante, Vj , é constante, tem-se:

$$\frac{dTj_i}{dt} = \frac{Fj_i}{Vj_i}(Tj_e - Tj_i) - \left(\frac{UA_i}{\rho_j C_{p_j} Vj_i} \right) LMDT_i \quad (6.13)$$

onde Fj é a vazão de fluido refrigerante no trocador de calor, Vj é o volume de fluido refrigerante no trocador de calor, ρ_j e C_{p_j} são a densidade e a capacidade calorífica do fluido refrigerante, respectivamente.

O modelo matemático que descreve o processo é formado por um sistema de equações diferenciais ordinárias de primeira ordem. Para obter a solução numérica deste sistema, é necessário que se conheça uma condição inicial para cada uma das EDOs. As condições iniciais utilizadas foram o estado estacionário de referência (calculado para uma vazão de alimentação de mosto $Fa = 100 \text{ kg/m}^3$), cujos valores encontram-se

6.2 Descrição do Processo

descritos na tabela 6.2. As vazões de fluido reagente e refrigerante dentro dos trocadores de calor foram calculadas para cada reator de modo a maximizar a troca térmica e minimizar o custo operacional. Os valores destas variáveis para cada reator, assim como os parâmetros de projeto utilizados na simulação do modelo dinâmico foram obtidos diretamente da unidade industrial e são apresentados na tabela 6.3. Os demais parâmetros necessários para a integração das equações são apresentados na tabela 6.4.

Tabela 6.2: Estado estacionário de referência.

Reator	$S(kg/m^3)$	$P(kg/m^3)$	$X(kg/m^3)$	$T(^{\circ}C)$	$Tc(^{\circ}C)$	$Tj(^{\circ}C)$
1	54,237	41,829	29,373	33,5	31,2	30,2
2	21,443	56,423	30,455	33,5	31,3	30,2
3	5,045	63,719	30,996	33,5	31,3	30,1
4	0,883	65,572	31,133	33,5	31,9	29,6

Tabela 6.3: Parâmetros de projeto.

Reator	$A(m^2)$	$V(m^3)$	$Fc(m^3/h)$	$Fj(m^3/h)$	$Vc(m^3)$	$Vj(m^3)$
1	76,361	210,374	400	400	20	20
2	63,242	268,037	350	350	20	20
3	31,061	316,663	180	180	20	20
4	6,809	208,208	60	60	20	20

Tabela 6.4: Parâmetros do modelo fenomenológico.

Parâmetros	Valores
ΔH	-157,57 kcal/h
ρ	950 kg/m ³
Cp	1 kcal/kg ^o C
U	3500 kcal/h ^o Cm ²
ρ_j	1000 kg/m ³
Cp_j	1 kcal/kg ^o C

6.2.2.3 Equações para o Reciclo

Conforme mencionado anteriormente, as correntes de alimentação e de reciclo se juntam para formar a corrente de alimentação do processo. O equacionamento desta etapa do processo pode ser descrito por balanços estacionários uma vez que as dinâmicas das operações envolvidas no reciclo (centrifugação, purga, diluição e mistura) são significativamente mais rápidas do que a dinâmica do processo fermentativo. Desse modo, os comportamentos dinâmicos destes equipamentos podem ser desprezados, assumindo-se a condição de estado pseudo-estacionário. Além disso, considera-se que, ao passar pela centrífuga, as concentrações de substrato e de etanol permanecem constantes e iguais aos seus respectivos valores na saída do quarto reator (ANDRIETTA, 1994).

Desse modo, dadas a vazão de alimentação de mosto, Fa , e a razão de reciclo, R , e desprezando as variações de densidade ao longo do processo, pode-se determinar as vazões de alimentação, F_0 , e de reciclo, F_R , da seguinte maneira:

$$F_0 = \frac{Fa}{(1 - R)} \quad (6.14)$$

$$F_R = F_0 - Fa \quad (6.15)$$

Considerando que a concentração de células de levedura, X_R , é mantida constante no reciclo através da manipulação da vazão de água, F_{H_2O} , de acordo com a concentração de células no creme de levedura, X_L , o balanço de massa em torno do tanque de diluição é escrito da seguinte forma (FONSECA, 1998):

$$Fl_1 = \frac{F_R X_R}{X_L} \quad (6.16)$$

$$F_{H_2O} = F_R - Fl_1 \quad (6.17)$$

6.2 Descrição do Processo

$$S_R = \frac{Fl_1 S_4}{F_R} \quad (6.18)$$

$$P_R = \frac{Fl_1 P_4}{F_R} \quad (6.19)$$

onde Fl_1 é a vazão de creme de levedura após a purga. As vazões do vinho delevedurado, F_V , e do creme de levedura (após a centrifuga e antes da purga), F_L , são obtidas através do balanço de massa em torno da centrifuga:

$$F_V = F_0 \frac{X_L - X_4}{X_L - X_V} \quad (6.20)$$

$$F_L = F_0 - F_V \quad (6.21)$$

onde X_4 e X_V são as concentrações de levedura na saída do quarto fermentador e no vinho delevedurado, respectivamente. O balanço material em torno do ponto de purga é dado por:

$$F_P = F_L - Fl_1 \quad (6.22)$$

As concentrações da corrente de alimentação do primeiro fermentador são obtidas através do balanço de massa em torno do ponto de mistura das vazões de mosto e de reciclo:

$$S_E = \frac{F_R S_R + F_a S_0}{F_0} \quad (6.23)$$

$$P_E = \frac{F_R P_R}{F_0} \quad (6.24)$$

$$X_E = \frac{F_R X_R}{F_0} \quad (6.25)$$

onde S_R e S_0 são as concentrações de substrato na corrente de reciclo e no mosto, respectivamente, e P_R é a concentração de etanol na corrente de reciclo.

Segundo DECHECHI (1998), a análise do comportamento dinâmico do sistema em malha aberta demonstrou que este processo é altamente complexo pois apresenta dinâmicas não lineares e resposta inversa quando submetido a determinadas perturbações. Estes resultados, segundo este autor, justificam a análise da viabilidade da implementação de técnicas de controle avançado.

Através da análise do comportamento dinâmico do processo, verificou-se que duas variáveis - a vazão de alimentação do mosto Fa e razão de refluxo R - são candidatas à manipulação devido às suas influências nas variáveis de saída. Contudo, dados os inconvenientes associados à manipulação da razão de refluxo (discutidos anteriormente), a vazão de alimentação do mosto foi escolhida como a variável manipulada do processo.

6.3 Estratégias de Identificação para o Processo de Fermentação Contínua

Conforme destacado nos capítulos 3 e 4, a identificação de processos a partir de dados amostrados vem despontando cada vez mais como uma alternativa aos modelos puramente fenomenológicos. Dentre as possíveis razões para a popularização da identificação de sistemas via modelos caixa preta, destacam-se a complexidade dos sistemas de interesse e a inerente dificuldade da modelagem fenomenológica aliados ao barateamento e ao bom desempenho dos computadores pessoais, que possibilitou a aquisição e o processamento de dados coletados diretamente das plantas (AGUIRRE, 2000).

As técnicas de identificação de sistemas (particularmente as não lineares) têm sido cada vez mais empregadas em abordagens relacionadas com o projeto de controladores. Esta abordagem tem especial importância nas estratégias avançadas de controladores preditivo e adaptativo, onde os modelos identificados são empregados na predição do comportamento do processo ou mesmo como parte integrante da estrutura do próprio controlador.

Nesta seção serão apresentadas e discutidas as etapas necessárias para identificar o processo contínuo de fermentação alcoólica. O objetivo principal é identificar modelos deste processo, empregando diferentes técnicas, que sejam capazes de fornecer predições acuradas do comportamento dinâmico do sistema em horizontes de longo alcance. Estes modelos serão posteriormente utilizados em estratégias de controle avançado (seção 6.4). A abordagem empregada é baseada na teoria apresentada no Capítulo 4, seguindo o procedimento geral de identificação de sistemas não lineares utilizando redes neurais.

Conforme discutido anteriormente neste capítulo, a viabilidade econômica deste processo depende diretamente do controle adequado da concentração de substrato (S_4) na saída da quarta dorna. O controle desta variável, realizado através da manipulação da vazão de mosto que alimenta o primeiro reator, tem como objetivo aumentar a produtividade do processo sem comprometer seu rendimento, através da minimização dos gastos com insumos. Desse modo, o desenvolvimento de um modelo que seja capaz de descrever o comportamento dinâmico desta variável é de fundamental impor-

tância para o projeto de controladores avançados. Contudo, outras duas variáveis - as concentrações de etanol (P_4) e de microorganismos (X_4) na saída da quarta dorna - estão fortemente relacionadas com a variável controlada através das cinéticas de crescimento do microorganismo e de produção de etanol. Sendo assim, o desenvolvimento de modelos dinâmicos confiáveis para estas variáveis pode ser bastante útil tanto para simulações da planta em diferentes pontos de operação, quanto para o desenvolvimento de estratégias de controle multi-variável.

Ao longo desta seção serão apresentados os procedimentos de identificação deste processo empregando três modelos MISO independentes (um para cada variável de interesse) e também modelos MIMO (considerando todas as variáveis mensuráveis do processo). A identificação e a posterior avaliação do desempenho dos modelos identificados envolverão modelos lineares, neurais e neuro-nebulosos, visto que um dos objetivos deste trabalho é explorar o potencial de diferentes modelos para fins de identificação e controle de processos.

6.3.1 Determinação das Variáveis de Entrada, Saída e Perturbação

Dadas as condições pré-otimizadas de operação da planta (seção 6.2), consideraram-se as seguintes variáveis de entrada do processo:

- Vazão de Alimentação do Mosto (Fa [m^3/h]): Esta é a principal variável manipulada da planta. A faixa de variação desta variável é representado pelo intervalo $[50, 150]$, que foi definido em termos da viabilidade econômica e operacional do processo. O intervalo representa os limites inferior e superior da vazão de alimentação de substrato e está relacionado com as limitações da operação da válvula e das dornas de fermentação.
- Razão de Reciclo (R [*adimensional*]): Esta variável pode ser considerada como uma perturbação medida ou como uma variável manipulada, dependendo do modo como ela é operada. Em condições normais de operação seu valor é de 0,3 porém, este valor pode ser alterado para corrigir problemas geralmente relacionados com os microorganismos. Contudo, mesmo nesses casos, a faixa de variação

6.3 Estratégias de Identificação para o Processo de Fermentação Contínua

não costuma exceder os 10%, resultando no intervalo $[0.27, 0.33]$.

- Concentração de ART no Mosto (S_0 $[kg/m^3]$): O valor nominal desta variável nas condições normais de operação é de $180\text{ }kg/m^3$. Contudo, uma vez que ela depende da cana-de-açúcar utilizada e da capacidade de homogeneização do tanque de alimentação, podem ocorrer pequenas variações de pelo menos $\pm 5\%$ em torno deste valor. Neste caso, esta variável passa a ser uma perturbação mensurável variando dentro do intervalo $[170, 190]$.

As principais variáveis de saída consideradas nesta planta são:

- Concentração de ART na saída da quarta dorna (S_4 $[kg/m^3]$)
- Concentração de etanol na saída da quarta dorna (P_4 $[kg/m^3]$)
- Concentração de microorganismos na saída da quarta dorna (X_4 $[kg/m^3]$)

6.3.2 Geração dos Dados para Identificação do Processo

Quando é possível trabalhar com um simulador validado do processo, tem-se a vantagem de poder gerar arquivos de dados para identificação tão grandes quanto se queira. Assim, foi gerado um conjunto de dados contendo os sinais de entrada e saída referentes a 5000 horas de operação deste processo. As variáveis de entrada do processo foram perturbadas de acordo com o procedimento descrito no item I (página 83) da seção 4.3.4, gerando sinais semelhantes àqueles ilustrados na figura 4.2 (página 84).

Nestes dados, a variável manipulada, Fa , foi variada como uma seqüência de degraus com período de 10 horas (longo o suficiente para que o processo pudesse atingir o estado estacionário) e amplitude uniformemente distribuída dentro do intervalo operacional $[50, 150]$.

As entradas R e S_0 (perturbações) também foram variadas com degraus com período constante (25h e 50h, respectivamente), porém suas amplitudes tiveram seus valores variados - dentro das respectivas faixas de variação - com uma distribuição Gaussiana. Isto foi feito com o objetivo de expressar as características estatísticas inerentes às perturbações medidas, ou seja, a maior probabilidade de ocorrência dos

valores nominais destas variáveis, ou valores próximos a eles, durante a maior parte do tempo de operação. Para tanto, as funções de distribuição de probabilidade Gaussiana foram centralizadas nos valores nominais de R e S_0 e os desvios padrão foram ajustados para $1/6$ dos respectivos intervalos operacionais ($S_0 = [170, 190]$ e $R = [0.27, 0.33]$), de tal modo que as amplitudes dos degraus gerados aleatoriamente pertençam a estes intervalos com uma probabilidade de aproximadamente 99%.

6.3.3 Determinação do Período de Amostragem

O conjunto de dados gerado com as características descritas na seção 6.3.2 foi amostrado de acordo com o procedimento tradicional de amostragem de sistemas dinâmicos (descrito por ÅSTRÖM e WITTENMARK (1997)), no qual o período de amostragem, T , é dado por:

$$N_T = \frac{T_S}{T} \approx 4 \text{ a } 10 \quad (6.26)$$

onde T_S é o tempo de subida do processo e N_T é o número de dados amostrados durante este tempo.

Contudo, no caso de processos multi-variáveis não lineares, o tempo de subida depende dos valores das entradas e pode ser diferente para cada saída. Em casos como este, deve-se escolher a dinâmica mais rápida. O tempo de subida do processo de fermentação contínua foi estimado aproximadamente entre 2 e 3 horas através de vários experimentos de simulação. Assim, da equação (6.26) obtém-se que $T \approx [12 \text{ min}, 45 \text{ min}]$. Além disso, o período de amostragem deve ser múltiplo de 15 min, que é o tempo médio de análise do cromatógrafo utilizado para medir a concentração de ART durante a fermentação. Com base nestas considerações, selecionou-se $T = 30 \text{ min}$. Este valor é grande o suficiente para garantir que a análise da concentração de açúcares seja realizada assim como para evitar problemas numéricos na etapa de estimação dos parâmetros do modelo. O procedimento de amostragem das 5000 horas de simulação gerou um conjunto de 10000 dados discretos, parte dos quais foi utilizado para a estimação dos modelos e a parte restante para a sua validação.

6.3 Estratégias de Identificação para o Processo de Fermentação Contínua

6.3.4 Determinação da Estrutura dos Modelos

Os estudos preliminares voltados para a identificação deste processo foram realizados tendo como base os modelos com estrutura hierárquica neuro-nebulosa propostos por CAMPELLO (2002) e CAMPELLO e AMARAL (1999b; 2000b) (também discutida na seção 3.4.5). Embora os modelos hierárquicos neuro-nebulosos (HNN) sejam um tipo muito particular de modelo não linear, o procedimento para a determinação dos regressores é o mesmo para qualquer tipo de modelo desta classe. Sendo assim, o vetor de regressão determinado para este modelo também foi utilizado para os demais modelos não lineares empregados para identificar o processo fermentativo apresentado neste capítulo.

Na seqüência, será discutida a determinação da estrutura dos modelos HNN, cujo procedimento envolve três conceitos distintos que estão associados à estrutura destes modelos: 1) Os regressores (também chamados de vetor de regressão) das variáveis de entrada e saída do modelo dinâmico; 2) A ordem hierárquica das variáveis; e 3) A característica estrutural interna do modelo HNN.

Os regressores do modelo foram selecionados através de simulações em um processo de tentativa e erro, de acordo com os critérios descritos a seguir. Como o objetivo desta etapa é determinar apenas o vetor de regressão, optou-se por fazer um estudo de vários modelos SISO, considerando todas as 9 combinações possíveis de variáveis de entrada e saída (3×3). Neste procedimento foram utilizados conjuntos contendo 1000 pares de dados entrada-saída gerados especificamente com esta finalidade. Em cada conjunto de dados, apenas a variável de entrada de interesse foi variada² enquanto as demais variáveis de entrada foram mantidas em seus valores de referência. Ou seja, considerando que o modelo SISO a ser identificado envolva as variáveis ($Fa \rightarrow S_4$), a variável Fa foi variada aleatoriamente dentro do intervalo $[50, 150] \text{ kg/m}^3$ enquanto as variáveis S_0 e R foram mantidas constantes nos valores de 180 kg/m^3 e $0,3$, respectivamente.

Considerando a simplicidade da representação SISO e também o número de

²Com período constante e igual a $10h$, degraus com amplitude aleatória gerados de acordo com o procedimento descrito no item I (página 83) da seção 4.3.4 e variando dentro do universo de discurso apresentado na seção 6.3.1.

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

simulações necessárias para avaliar diferentes regressores para cada uma das nove diferentes combinações de pares entrada-saída, optou-se por trabalhar com um modelo mais simples nesta fase da identificação do processo (determinação do vetor de regressão). O modelo utilizado foi uma rede neural RBF convencional com 5 neurônios treinada com o estimador dos mínimos quadrados recursivos.

Os regressores foram selecionados de acordo com o desempenho dos modelos SISO em previsões um passo adiante e em simulação recursiva. Para cada par de variáveis entrada-saída, selecionaram-se os regressores que proporcionavam o melhor ajuste com a menor ordem possível (número de termos passados). Foi dada atenção especial na determinação da ordem das variáveis de saída, uma vez que estas variáveis estão sujeitas a erros de predição no caso de simulações em horizontes maiores do que um. Os vetores de regressão para cada variável selecionados com esta metodologia foram os seguintes:

$$[Fa(k-1) Fa(k-2)], [S_0(k-1)], [R(k-1)], [S_4(k-1)], [P_4(k-1)] \text{ e } [X_4(k-1)].$$

Os resultados apresentados pelos modelos SISO, que consideravam apenas uma saída e desprezavam o efeito das demais, mostraram que seria possível construir um modelo MIMO a partir de três modelos MISO com estrutura hierárquica completamente desacoplados. Esta propriedade é bastante desejável, uma vez que a independência dos modelos implica na independência dos seus erros de predição que, por sua vez, pode resultar em um melhor desempenho global de modelagem em horizontes de longo alcance. A estrutura interna destes três modelos MISO hierárquicos foi definida na seção 3.4.5 (e discutida com mais detalhes em CAMPELLO (2002) e MELEIRO *et al.* (2001)) com $c = 5$ cinco conjuntos nebulosos que, segundo PEDRYCZ (1995), é um valor típico no contexto da lógica nebulosa.

A ordem hierárquica das variáveis foi selecionada utilizando o conhecimento *a priori* sobre o comportamento dinâmico do processo. A variável Fa foi posicionada no primeiro nível hierárquico do modelo pelo fato de ser a variável manipulada do processo e a principal variável de entrada, além de possuir o maior regressor. Deve-se destacar que o primeiro bloco de processamento é o que possui o mapeamento entrada-saída mais complexo. A realimentação da variável de saída de cada um dos modelos foi posicionada no nível hierárquico mais baixo, tomando como base as informações obtidas sobre suas

6.3 Estratégias de Identificação para o Processo de Fermentação Contínua

relevâncias sobre o desempenho dos modelos durante a etapa de seleção dos regressores. Além disso, posicionando estas variáveis no nível hierárquico mais baixo, minimiza-se o impacto causado pela realimentação de erros de predição no caso de simulações recursivas. Finalmente, para selecionar a ordem hierárquica das variáveis perturbação, considerou-se um conjunto de simulações realizadas por DECHECHI (1998) na mesma planta de produção de etanol. Os resultados obtidos por aquele autor indicaram que as não linearidades associadas às perturbações em S_0 são mais fortes do que aquelas associadas às perturbações em R . Por este motivo, a ordem hierárquica da primeira variável foi feita maior do que a da segunda.

A figura 6.3 ilustra a estrutura do modelo hierárquico identificado para a predição da variável S_4 . A disposição das variáveis de entrada dos modelos utilizados para representar o comportamento dinâmico de P_4 e X_4 obedecem à mesma ordem hierárquica, *mutatis mutandis*.

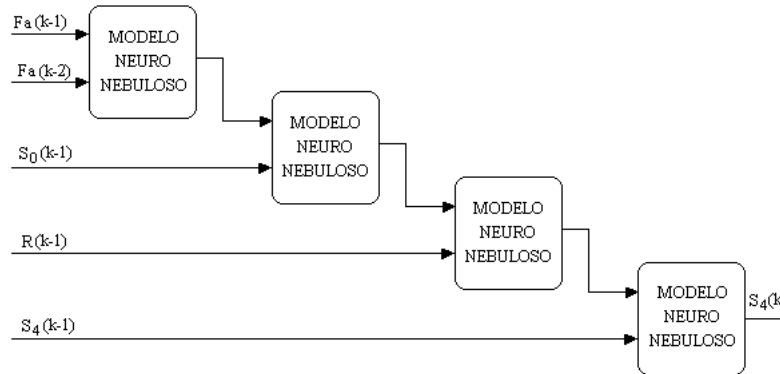


Figura 6.3: Modelo hierárquico neuro-nebuloso para predição de S_4 .

Os demais tipos de modelos identificados utilizaram os mesmos vetores de regressão selecionados para o modelo HNN, enquanto que a ordem hierárquica das variáveis³ somente diz respeito a estes modelos. A estrutura selecionada para o modelo linear foi do tipo ARX (seção 3.3.3). Para os modelos neurais MLP e Construtivo, empregou-se a estrutura não linear correspondente ao modelo linear ARX, ou seja, do tipo NNARX (seção 4.4.2). Contudo, tanto o modelo linear quanto os modelos neurais apresentaram melhor desempenho quando a representação MIMO foi empregada, sobretudo em simulações recursivas. A estrutura geral dos modelos neurais é ilustrada

³Item 2 da página 173.

na figura 6.4.

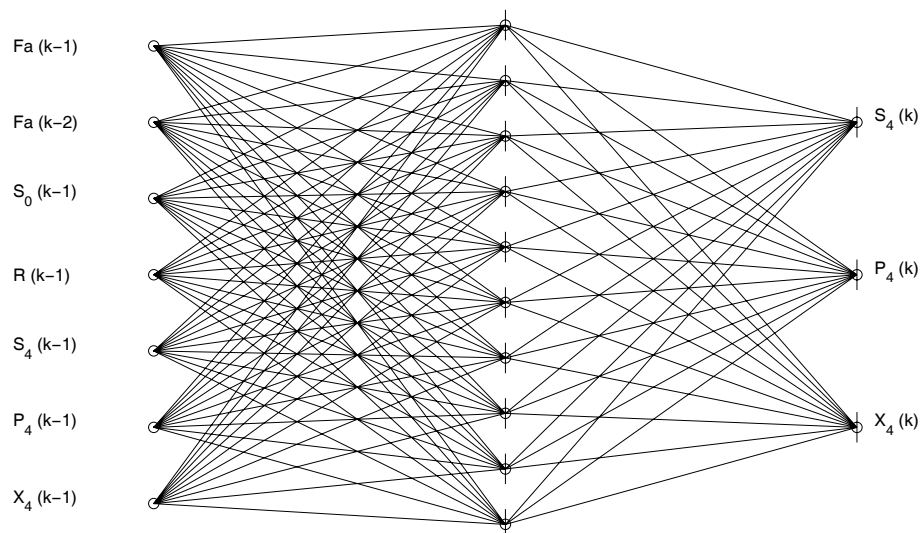


Figura 6.4: Estrutura dos modelos NNARX.

6.3.5 Estimação e Validação dos Modelos

Utilizando o arquivo de dados entrada-saída representativo do comportamento dinâmico do processo correspondente a 5000 horas de operação (seção 6.3.2), separaram-se as primeiras 4500 horas para a estimação dos modelos e as 500 horas restantes foram utilizadas na validação dos mesmos. Todos os modelos propostos (HNN, neurais e linear) foram estimados e validados com os mesmos conjuntos de dados.

Os modelos foram estimados até o número de iterações/épocas onde seus algoritmos de ajuste de parâmetros não mais pudessem melhorar o desempenho do modelo frente aos dados de validação. O critério para evitar o sobre-ajuste ou *overfitting* foi interromper o treinamento quando o erro quadrático médio referente ao conjunto de dados de validação permanecesse estacionado ou voltasse a crescer sistematicamente.

6.3 Estratégias de Identificação para o Processo de Fermentação Contínua

6.3.6 Identificação Utilizando Modelos Hierárquicos HNN

Uma vez que cada modelo hierárquico apresenta $n = 5$ entradas e $c = 5$ conjuntos *fuzzy* para cada entrada e considerando ainda que os centros e as aberturas das regras *fuzzy* (ver seção 3.4.5) também foram otimizados, o número total de parâmetros para cada um dos modelos hierárquicos é igual a $p = (n - 1) c^2 + 4(n - 1) c = 180$.

A tabela 6.5 apresenta os erros de modelagem (EQM) apresentados pelos modelos HNN na predição do comportamento dinâmico das variáveis S_4 , P_4 e X_4 .

Tabela 6.5: EQM de modelagem para os dados de validação - modelos HNN.

Variável de Saída	EQM Prev. 1 Passo	EQM Sim. Recursiva
S_4	0,016914	0,227619
P_4	0,012912	0,355494
X_4	0,000920	0,049269

Contudo, estes erros não são comparáveis entre si uma vez que os universos de discurso das variáveis de saída são bastante diferentes. Para permitir uma comparação quantitativa entre os desempenhos dos modelos de cada uma das saídas, utilizou-se o erro quadrático médio normalizado (EQMN) definido assim (MELEIRO *et al.*, 2001):

$$\text{EQMN} \left(\{y(k), \hat{y}(k)\}_{k=1}^N \right) = \frac{1}{N} \sum_{k=1}^N \left(\frac{y(k) - \hat{y}(k)}{\bar{y}(k)} \right)^2 \quad (6.27)$$

onde $y(k)$ representa uma das saídas do processo e $\hat{y}(k)$ a respectiva predição do modelo. A variável $\bar{y}(k)$ é o termo de normalização dado por:

$$\bar{y}(k) = y(k) - \frac{1}{N} \sum_{k=1}^N y(k) \quad (6.28)$$

A tabela 6.6 mostra os erros de modelagem, segundo o critério EQMN, nas simulações um passo adiante e recursiva quando os dados de validação são utilizados.

O desempenho dos modelos HNN resultantes quando se utilizou os dados de validação são apresentados a seguir. As figuras 6.5, 6.6 e 6.7 apresentam o resultado

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

Tabela 6.6: EQMN de modelagem para os dados de validação - modelos HNN.

Variável de Saída	EQMN Prev. 1 Passo	EQMN Sim. Recursiva
S_4	1,5888	12,1473
P_4	21,8146	262,1233
X_4	3,0208	145,9554

das simulações um passo adiante (acima) e recursiva (abaixo) para as três variáveis de saída do processo.

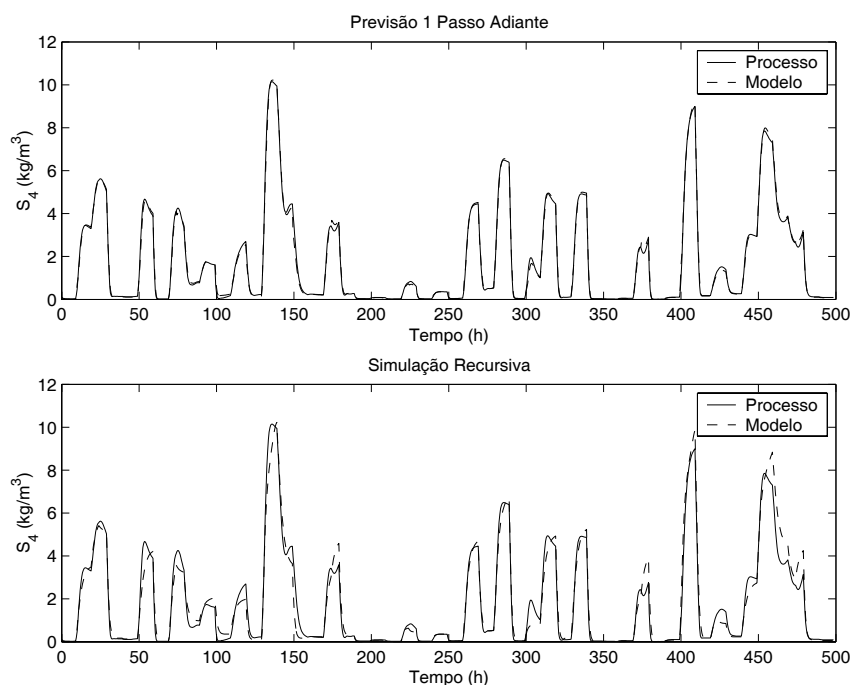


Figura 6.5: Desempenho do modelo HNN para a variável S_4 .

Estas figuras ilustram o bom desempenho dos modelos para as simulações um passo adiante e recursiva, especialmente considerando os longos horizontes de predição envolvidos nas simulações. Os maiores erros de predição estão associados à saída P_4 , especialmente na simulação recursiva, o que reflete a maior sensibilidade deste modelo a realimentação de erros de predição. Contudo, no contexto de aplicações de controle baseado em modelos - tais como os controladores preditivo - os modelos HNN identificados podem ser aplicados adequadamente, visto que tais controladores podem ser sintonizados para usar horizontes de predição relativamente pequenos.

6.3 Estratégias de Identificação para o Processo de Fermentação Contínua

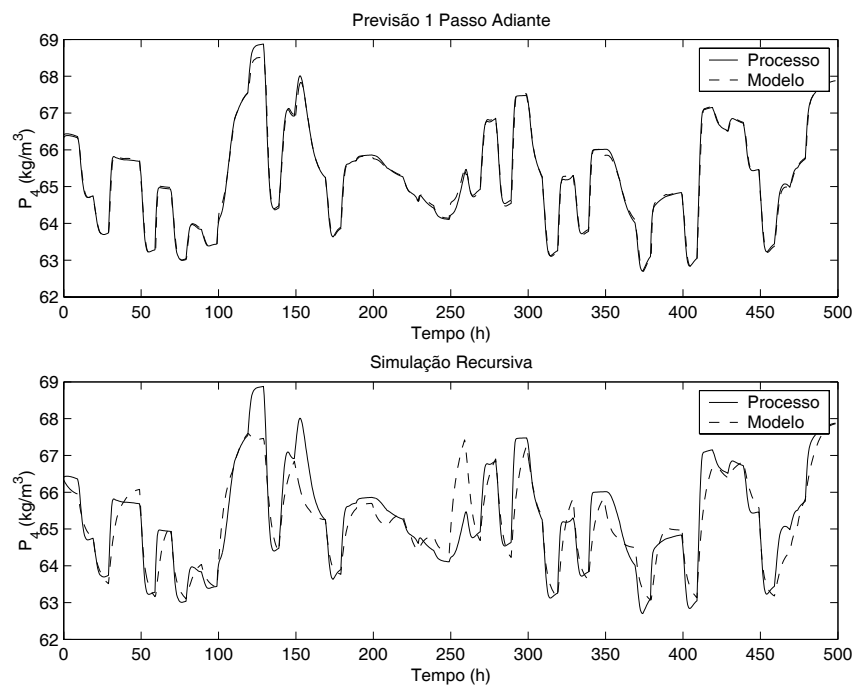


Figura 6.6: Desempenho do modelo HNN para a variável P_4 .

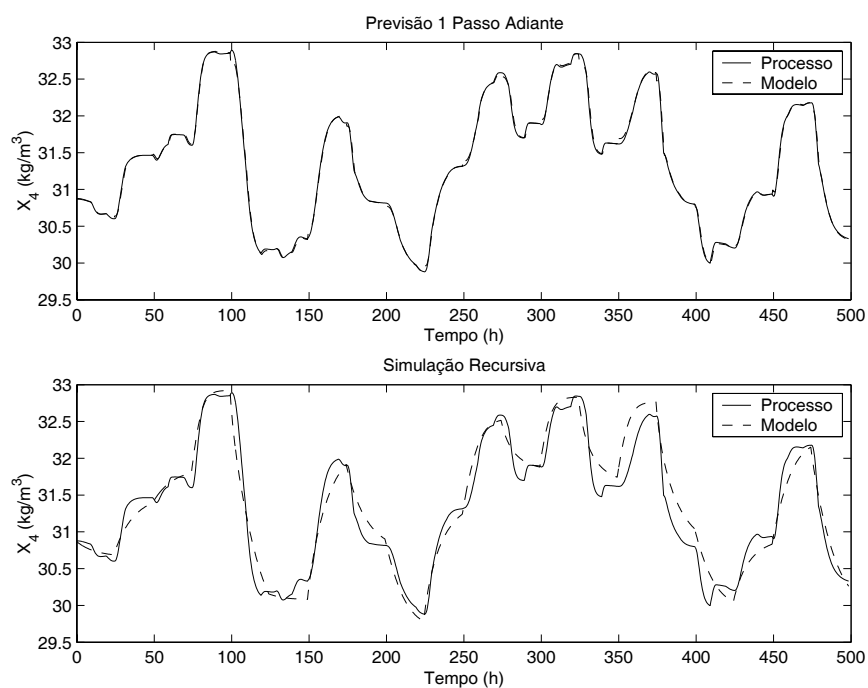


Figura 6.7: Desempenho do modelo HNN para a variável X_4 .

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

Outro ponto importante que deve ser ressaltado é que a maioria das aplicações de controle desenvolvidas para este processo, inclusive a proposta nesta tese, são voltadas para a variável S_4 , cujo EQMN referente à simulação recursiva é o menor das três variáveis.

6.3 Estratégias de Identificação para o Processo de Fermentação Contínua

6.3.7 Identificação Utilizando Modelos Neurais MLP

Nesta seção apresenta-se o resultado da modelagem do processo contínuo de fermentação alcoólica fornecido pelo modelo não linear representado por uma rede neural *feedforward* do tipo Perceptron Multicamadas (MLP).

Conforme mencionado anteriormente, os modelos neurais apresentaram melhor desempenho quando a representação MIMO foi utilizada. Assim, as redes neurais selecionadas apresentam a estrutura geral mostrada na figura 6.4, com $n_e = 7$ entradas e $n_s = 3$ saídas. As redes testadas utilizam funções de ativação do tipo tangente hiperbólica na camada escondida, l_e , e linear na camada de saída, l_s . O modelo neural MLP utilizado nesta seção foi treinado com um algoritmo de otimização que utiliza o método do *Gradiente Conjugado Escalonado Modificado* - GCEM (DE CASTRO e VON ZUBEN, 1998a), que permite o cálculo exato da informação de 2ª ordem com custo computacional substancialmente reduzido. O número de neurônios da camada escondida, n_{l_e} , foi determinado através da comparação dos resultados obtidos para o conjunto de dados de validação quando diferentes valores de n_{l_e} foram testados. A avaliação do desempenho dos modelos foi feita considerando tanto os erros finais de aproximação (EQM e EQMN), como também através da análise visual de simulações para previsão um passo adiante e recursiva. As tabelas 6.7 e 6.8 mostram os EQM e EQMN para vários valores de n_{l_e} .

Tabela 6.7: EQM de modelagem para os dados de validação - modelos MLP.

	Previsão 1 Passo				Simulação Recursiva		
n_{l_e}	S_4	P_4	X_4		S_4	P_4	X_4
3	0.0310	0.0213	0.0158		0.1016	0.1298	0.1251
5	0.0199	0.0152	0.0140		0.0649	0.0817	0.1068
10	0.0157	0.0115	0.0131		0.0587	0.0694	0.1006
15	0.0181	0.0138	0.0131		0.0585	0.0781	0.1044
20	0.0156	0.0114	0.0131		0.0522	0.0665	0.1002
25	0.0145	0.0109	0.0130		0.0489	0.0679	0.1019
50	0.0201	0.0149	0.0131		0.0616	0.0726	0.0926

Uma vez que os modelos identificados serão utilizados no projeto do controlador

**ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO
ALCOÓLICA CONTÍNUA**

Tabela 6.8: EQMN de modelagem para os dados de validação - modelos MLP.

	Previsão 1 Passo				Simulação Recursiva		
n_{l_e}	S_4	P_4	X_4		S_4	P_4	X_4
3	0.5187	2.7223	1.9451		2.6729	44.4250	28.4125
5	0.5089	0.7780	1.2546		3.5455	83.5960	7.8831
10	0.2824	0.0643	1.0046		1.3830	49.8079	9.8839
15	0.6105	0.1538	1.4277		3.9642	24.3317	8.1228
20	0.4076	0.2207	1.1240		3.4918	13.1790	5.8431
25	0.1744	0.2460	1.1125		2.0769	23.2255	10.7421
50	0.6205	0.2805	1.0119		3.3304	27.7519	6.8559

deste processo, os resultados de maior interesse são os desempenhos dos modelos em simulações recursivas pois a estratégia de controle preditivo utiliza previsões do modelo em horizontes geralmente maiores do que um para o cálculo das ações de controle. Analisando os erros de predição referentes à variável controlada (S_4), percebe-se que a partir de $n_{l_e} = 10$, a diminuição do EQM (tabela 6.7) não justifica o aumento de neurônios na camada escondida (os resultados referentes ao EQMN (tabela 6.8) também são coerentes com esta avaliação). As figuras 6.8, 6.9 e 6.10 apresentam o desempenho do modelo neural para $n_{l_e} = 10$.

Embora o EQM apresentado pelos modelos neurais com $n_{l_e} = 3$ e 5 não sejam tão ruins quando comparados com o EQM para o modelo com $n_{l_e} = 10$, a avaliação do desempenho do comportamento dinâmico (simulação recursiva) através da análise visual daqueles modelos (figuras 6.11 e 6.12) foi determinante para verificar que, apesar de valores de erro razoavelmente baixos, ambos os modelos fornecem valores negativos (e conseqüentemente sem sentido físico) para a concentração de S_4 . Por outro lado, mesmo o modelo neural com $n_{l_e} = 3$, fornece resultados dinâmicos melhores do que o apresentado pelo modelo linear ARX (figura 6.20).

Assim, a rede neural escolhida foi a que utilizou $n_{l_e} = 10$ neurônios na camada escondida. O número total de parâmetros a serem estimados para este modelo neural é dado por $p = (n_e + 1) n_{l_e} + (n_{l_e} + 1) n_{l_s} = 113$, onde n_{l_s} representa o número de neurônios da camada de saída.

6.3 Estratégias de Identificação para o Processo de Fermentação Contínua

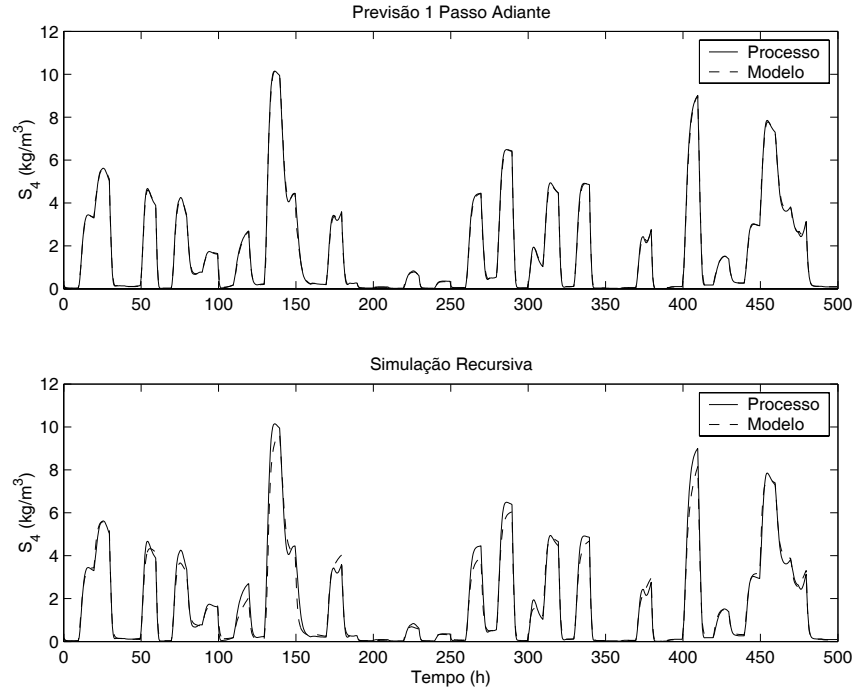


Figura 6.8: Desempenho do modelo MLP com $n_{l_e} = 10$ para a variável S_4 .

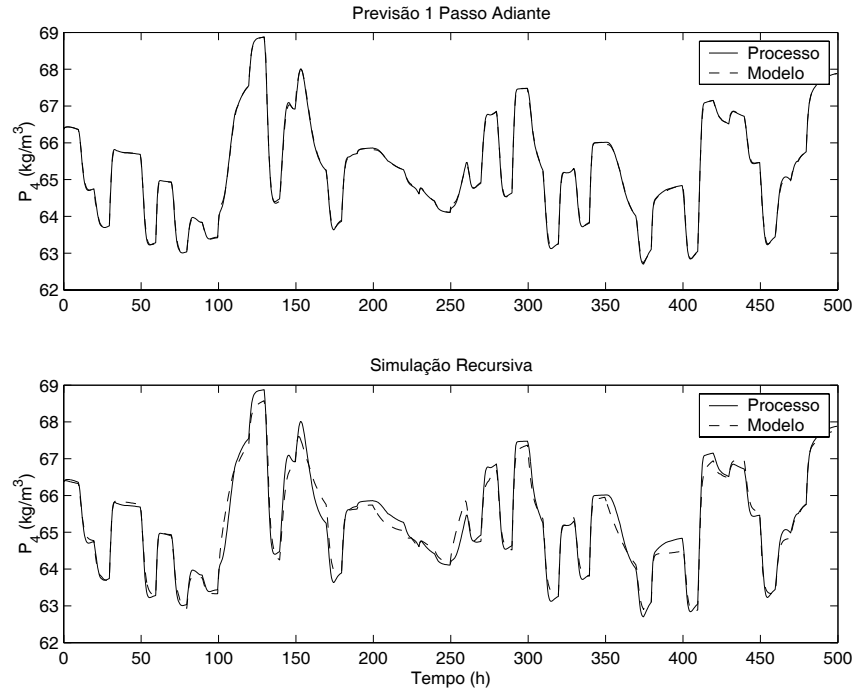


Figura 6.9: Desempenho do modelo MLP com $n_{l_e} = 10$ para a variável P_4 .

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

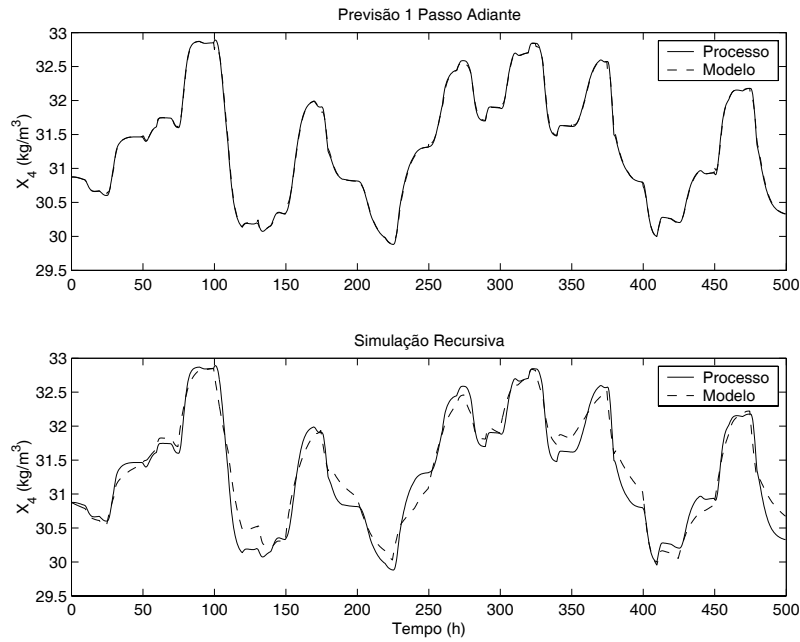


Figura 6.10: Desempenho do modelo MLP com $n_{l_e} = 10$ para a variável X_4 .

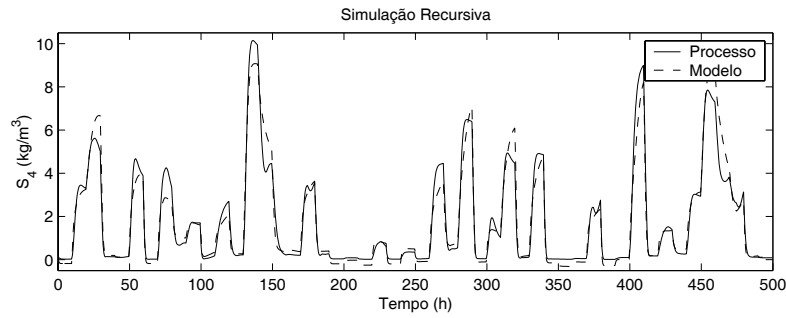


Figura 6.11: Simulação recursiva do modelo MLP com $n_{l_e} = 3$.

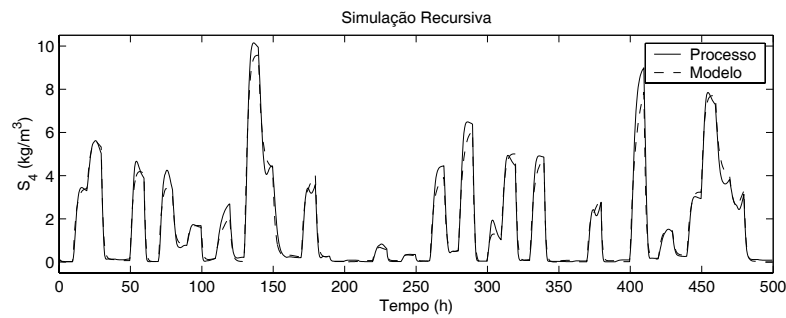


Figura 6.12: Simulação recursiva do modelo MLP com $n_{l_e} = 5$.

6.3.8 Identificação Utilizando o Modelo Neural Construtivo UGL

O principal inconveniente da aplicação da abordagem neural clássica (e da maioria dos modelos caixa-preta lineares e não lineares) na modelagem de sistemas dinâmicos é a dificuldade para determinar a estrutura interna ótima do modelo, ou seja, o número de neurônios da camada interna ou escondida, n_{le} . Além do tempo gasto com a determinação do número de neurônios escondidos, que a rigor é um método de tentativa e erro e por isso depende da experiência do projetista e de seu conhecimento sobre o processo, o procedimento para a determinação do número adequado de neurônios escondidos enfrenta ainda uma dificuldade extra, qual seja o fato de que a maioria dos algoritmos de treinamento serem sensíveis aos parâmetros de inicialização. Assim, uma rede com a mesma estrutura interna pode apresentar resultados finais diferentes dependendo dos pesos iniciais (ponto de partida na superfície de erro), que por sua vez são determinados aleatoriamente.

Alguns trabalhos vêm sendo desenvolvidos no sentido de superar um dos principais obstáculos para a maior utilização da abordagem neural nas tarefas de identificação e controle de sistemas não lineares, ou seja, a elaboração e o aperfeiçoamento de algoritmos capazes de determinar automaticamente a dimensão da rede neural (número de neurônios da camada escondida). Dentre os métodos disponíveis com esta finalidade, destacam-se os métodos de poda e os métodos construtivos, este último utilizado neste trabalho como ferramenta para identificar um modelo neural do processo fermentativo para posterior uso no projeto de um controlador preditivo não linear eficiente. Os pormenores acerca da estimação do modelo neural identificado via método construtivo são apresentados a seguir.

A exemplo do modelo neural MLP (e também para facilitar a comparação dos modelos), o modelo neural construtivo UGL utilizado para identificar este processo apresenta a estrutura MIMO representada na figura 6.4 ($n_e = 7$ entradas e $n_s = 3$ saídas). O modelo neural UGL, que pode ser entendido como uma rede neural não paramétrica, foi treinado com o algoritmo de aprendizado construtivo com funções de ativação paramétricas na forma de expansão truncada, utilizando funções de Hermite como funções básicas (VON ZUBEN, 1996). A ordem máxima adotada para as funções de Hermite foi $P = 6$.

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

O algoritmo de aprendizado construtivo UGL foi utilizado por MELEIRO *et al.* (2002b) para projetar uma rede neural de uma camada com estrutura ótima para obter a melhor aproximação do mapeamento gerado pelos dados de entrada-saída do processo fermentativo. A seguir são apresentados os resultados da identificação do processo fornecidos pelo modelo neural UGL.

As tabelas 6.9 e 6.10 apresentam os erros de modelagem, segundo os critérios EQM e EQMN, apresentados pelo modelo neural UGL na predição do comportamento dinâmico das variáveis S_4 , P_4 e X_4 em simulações um passo adiante e recursiva. A análise dos resultados apresentados pelos modelos neurais MLP e UGL, na forma dos erros EQM e EQMN (tabelas 6.7 e 6.8 para $n_{l_e} = 10$, versus tabelas 6.9 e 6.10), revela que o modelo neural UGL apresentou, de modo geral, resultados superiores em quase todas as situações.

Tabela 6.9: EQM de modelagem para os dados de validação - modelo UGL.

Variável de Saída	EQM Prev. 1 Passo	EQM Sim. Recursiva
S_4	0.0113	0.0393
P_4	0.0081	0.0522
X_4	0.0111	0.1314

Tabela 6.10: EQMN de modelagem para os dados de validação - modelo UGL.

Variável de Saída	EQMN Prev. 1 Passo	EQMN Sim. Recursiva
S_4	1.0530	0.9711
P_4	0.0457	20.3150
X_4	0.1348	21.5099

A avaliação do comportamento dinâmico do modelo neural UGL feita através da análise visual, tanto em simulações um passo adiante quanto recursiva (figuras 6.13, 6.14 e 6.15), corrobora o seu grande potencial em tarefas de identificação de processos não lineares multi-variáveis.

Outra vantagem prática fornecida por uma rede neural com arquitetura definida pelo algoritmo de aprendizado por busca de projeção é a interpretabilidade direta do papel individual de cada neurônio da camada escondida, uma vez que a informação

6.3 Estratégias de Identificação para o Processo de Fermentação Contínua

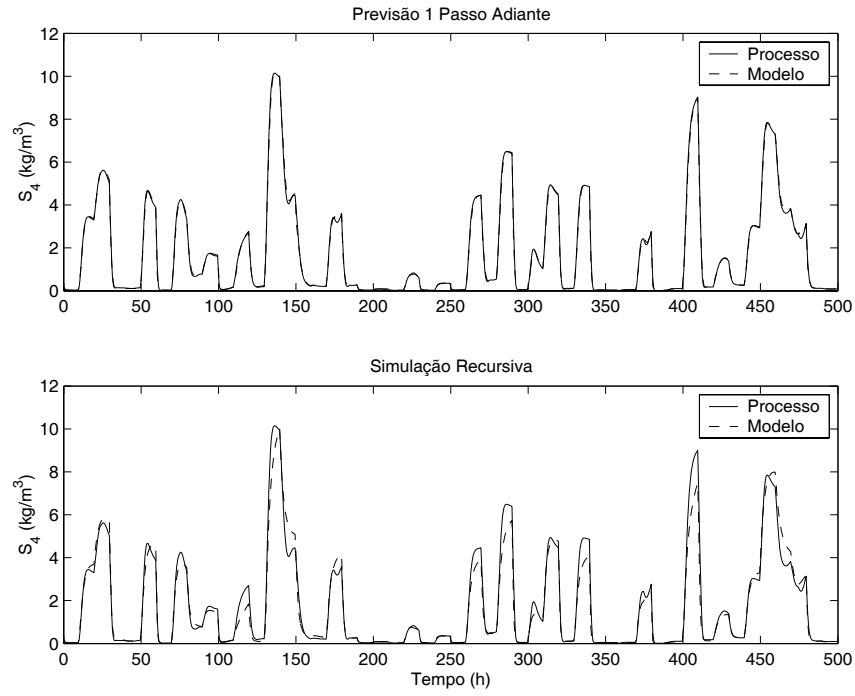


Figura 6.13: Desempenho do modelo UGL com $n_{l_e} = 4$ para a variável S_4 .

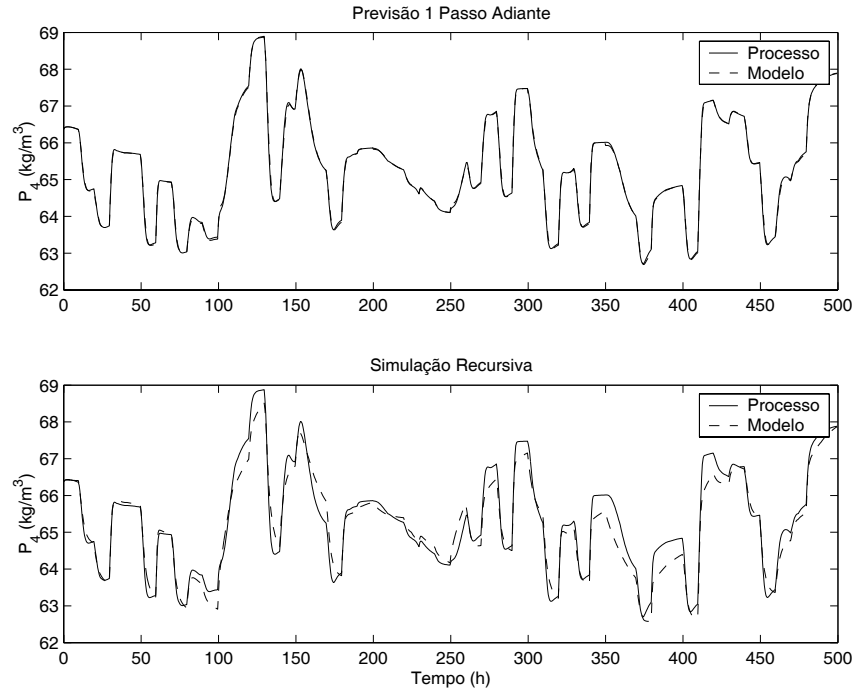


Figura 6.14: Desempenho do modelo UGL com $n_{l_e} = 4$ para a variável P_4 .

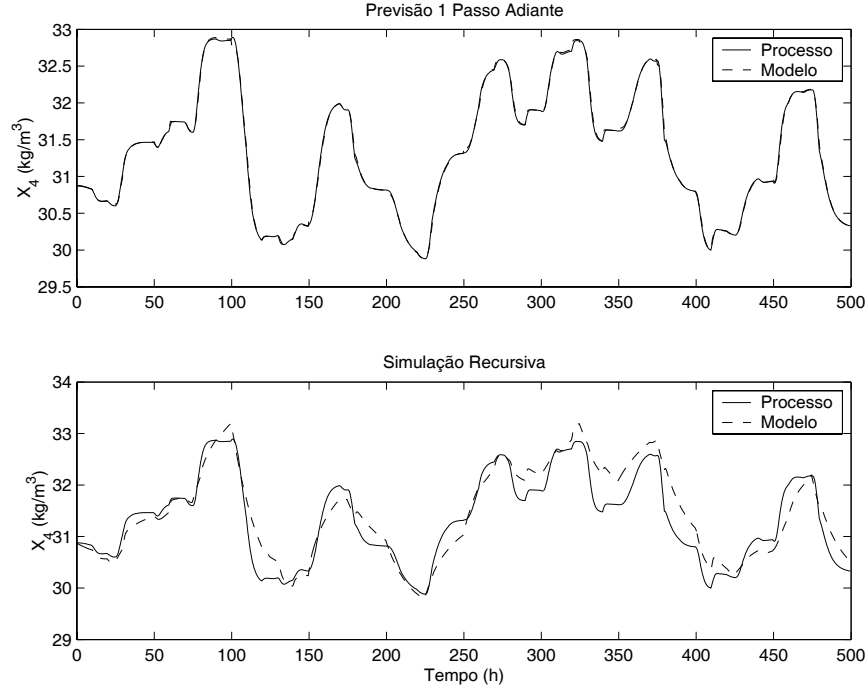


Figura 6.15: Desempenho do modelo UGL com $n_{l_e} = 4$ para a variável X_4 .

fornecida na entrada de cada neurônio é dependente da direção de projeção (MELEIRO *et al.*, 2002b). As figuras 6.16 a 6.19 exibem as funções de ativação de cada neurônio para as três saídas do processo (S_4 , P_4 e X_4 , respectivamente). Estas figuras ajudam a ilustrar a habilidade do algoritmo UGL para selecionar diferentes funções de ativação para cada neurônio de modo a realizar a melhor aproximação possível para um dado mapeamento entrada-saída.

O algoritmo de aprendizado construtivo foi capaz de dimensionar automaticamente uma rede neural com a mesma capacidade de aproximação da melhor rede neural MLP ($n_{l_e} = 10$) utilizando apenas $n_{l_e} = 4$ neurônios escondidos. A dimensão reduzida do modelo neural UGL (número de parâmetros: $p = (n_e + 1) n_{l_e} + (n_{l_e} + 1) n_{l_s} = 47$) apresenta a vantagem adicional de gerar um modelo neural mais parcimonioso e com capacidade de generalização potencialmente maior.

6.3 Estratégias de Identificação para o Processo de Fermentação Contínua

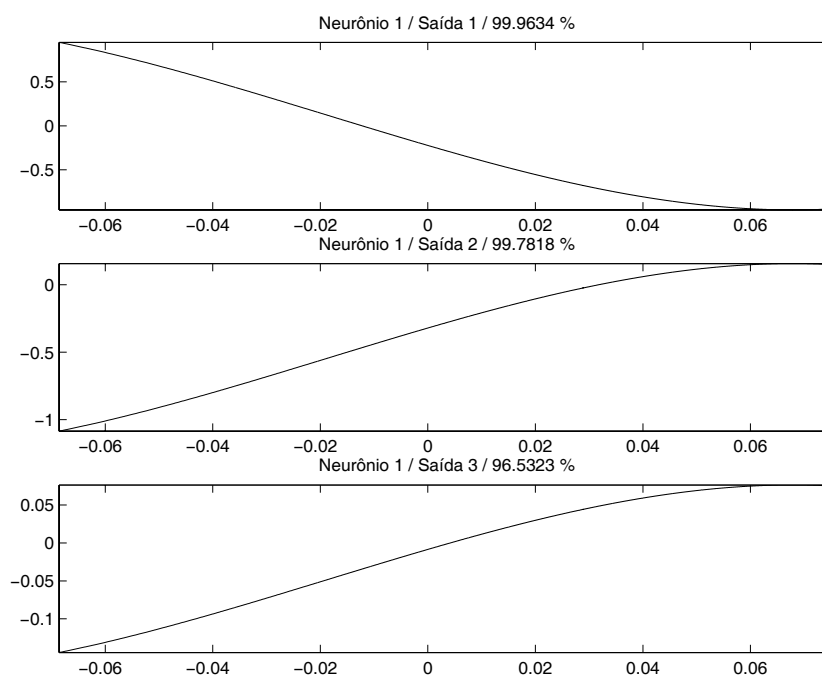


Figura 6.16: Funções de ativação do neurônio 1.

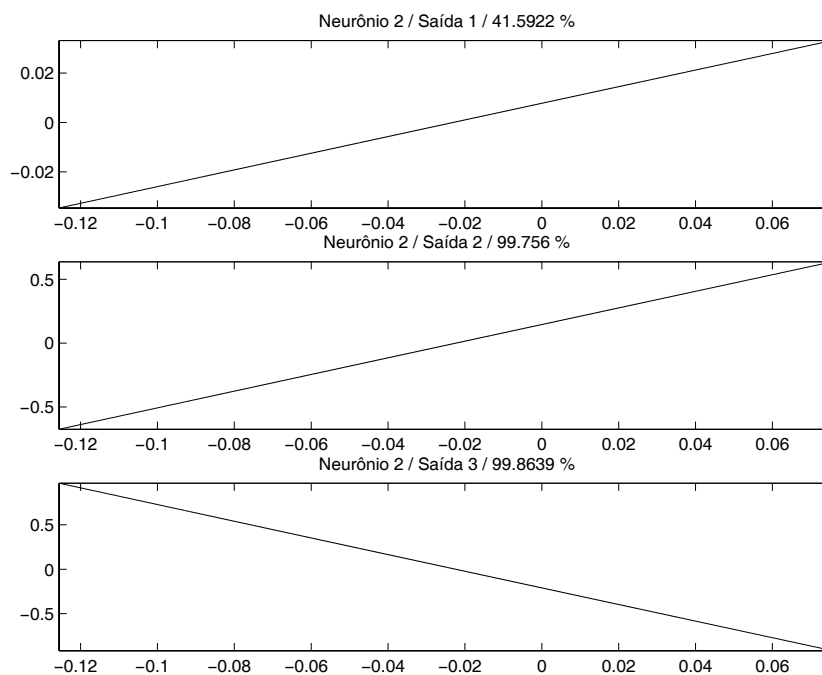


Figura 6.17: Funções de ativação do neurônio 2.

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

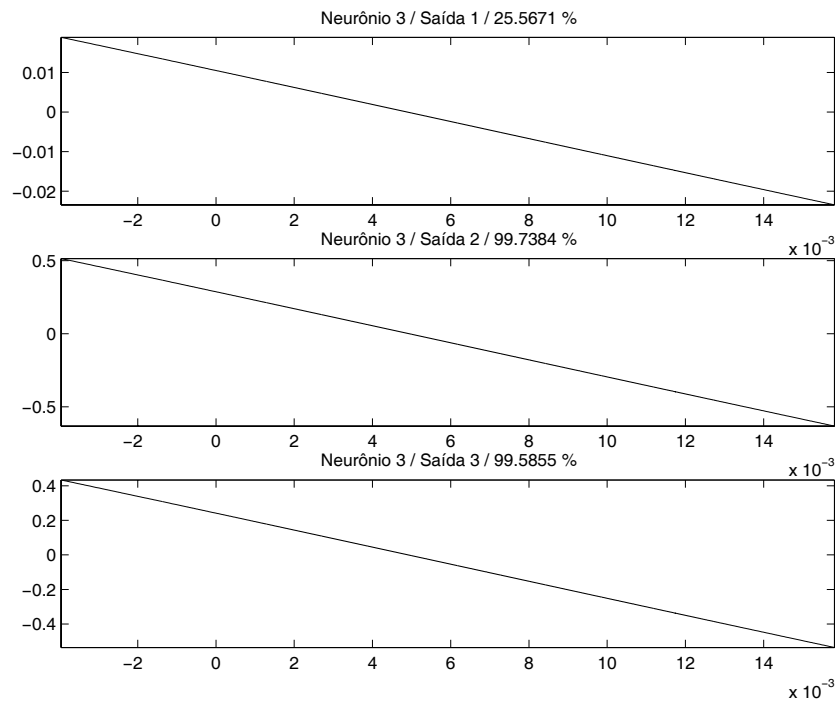


Figura 6.18: Funções de ativação do neurônio 3.

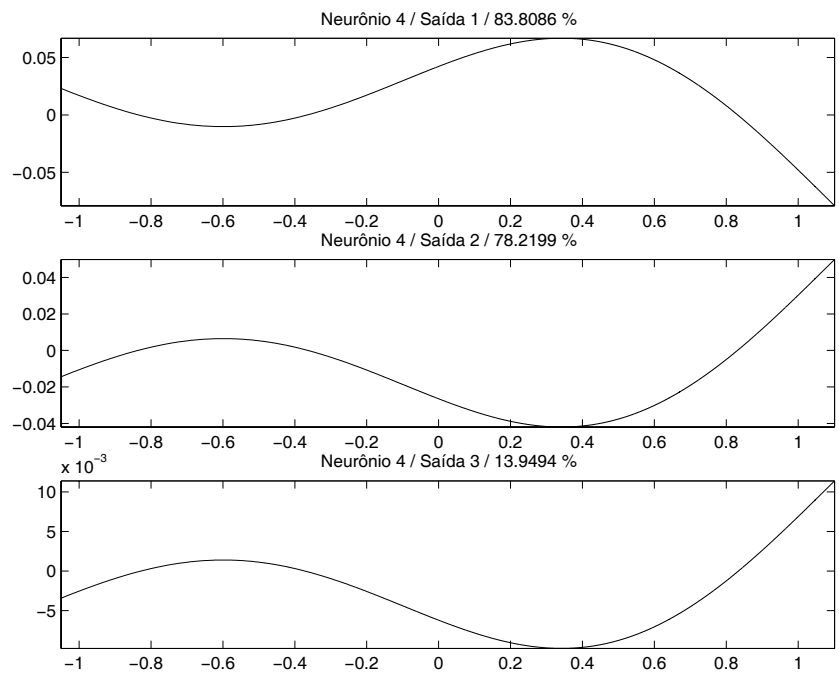


Figura 6.19: Funções de ativação do neurônio 4.

6.3.9 Identificação Utilizando o Modelo Linear ARX

Uma prática prudente em identificação de sistemas é tentar, em uma primeira etapa, identificar o processo de interesse com modelos lineares pois, além de se tratar de modelos mais simples, o resultado da identificação pode ajudar na avaliação do quão não linear é o processo. O desempenho do modelo resultante também pode ser útil como uma referência para analisar o desempenho de modelos não lineares mais elaborados. Ao comparar o desempenho dos modelos linear e não linear, pode-se chegar à conclusão de que ambos representam a dinâmica do sistema igualmente bem e, nesse caso, o modelo linear deve ser o escolhido por conta de sua simplicidade. Outro aspecto que também deve ser considerado é que tanto a seleção do método de estimação de parâmetros do modelo quanto o projeto de sistemas de controle baseado em modelos lineares são tarefas consideravelmente mais fáceis do que as suas correspondentes não lineares. Na seqüência, serão apresentados os resultados da modelagem do processo fermentativo contínuo fornecidos por um modelo linear multi-variável com estrutura ARX.

Com o objetivo de representar o comportamento dinâmico do processo, empregou-se um modelo linear ARX MIMO, representado pela seguinte expressão (LJUNG, 1999; AGUIRRE, 2000):

$$\begin{aligned} \mathbf{y}(k) = & -A_1\mathbf{y}(k-1) - \dots - A_{n_y}\mathbf{y}(k-n_y) \\ & + B_1\mathbf{u}(k-1) + \dots + B_{n_u}\mathbf{u}(k-n_u) + \boldsymbol{\nu}(k) \end{aligned} \quad (6.29)$$

onde $A_i \in \mathbb{R}^{n_s \times n_s}$, $B_i \in \mathbb{R}^{n_s \times n_e}$, n_u é o máximo atraso entre os regressores de entrada e n_y é o máximo atraso entre os regressores de saída. Os vetores de saída, entrada e ruído branco, \mathbf{y} , \mathbf{u} e $\boldsymbol{\nu}$, respectivamente, são definidos da seguinte forma:

$$\begin{aligned} \mathbf{y}(k) &= [y_1(k) \ y_2(k) \ \dots \ y_{n_s}(k)]^T \\ \mathbf{u}(k) &= [u_1(k) \ u_2(k) \ \dots \ u_{n_e}(k)]^T \\ \boldsymbol{\nu}(k) &= [\nu_1(k) \ \nu_2(k) \ \dots \ \nu_{n_s}(k)]^T \end{aligned} \quad (6.30)$$

De modo análogo aos polinômios $A(q^{-1})$ e $B(q^{-1})$ definidos na seção 3.3.3, pode-se definir as seguintes matrizes de polinômios:

$$\mathbf{A}(q^{-1}) = I + A_1 q^{-1} + \dots + A_{n_y} q^{-n_y};$$

$$\mathbf{B}(q^{-1}) = B_1 q^{-1} + \dots + B_{n_u} q^{-n_u};$$

resultando na seguinte expressão:

$$\mathbf{A}(q^{-1})\mathbf{y}(k) = \mathbf{B}(q^{-1})\mathbf{u}(k) + \boldsymbol{\nu}(k) \quad (6.31)$$

ou alternativamente

$$\mathbf{y}(k) = \boldsymbol{\Theta}^T \boldsymbol{\psi}(k-1) + \boldsymbol{\nu}(k) \quad (6.32)$$

onde

$$\begin{aligned} \boldsymbol{\Theta} &= [-A_1 \quad -A_2 \quad \dots \quad -A_{n_y} \quad B_1 \quad B_2 \quad \dots \quad B_{n_u}]^T \\ \boldsymbol{\psi}(k-1) &= [\mathbf{y}^T(k-1) \quad \dots \quad \mathbf{y}^T(k-n_y) \quad \mathbf{u}^T(k-1) \quad \dots \quad \mathbf{u}^T(k-n_u)]^T \end{aligned}$$

O modelo ARX MIMO identificado para este processo pode ser escrito da seguinte forma:

$$A_0 \mathbf{y}(k) = -A_1 \mathbf{y}(k-1) + B_1 \mathbf{u}(k-1) + B_2 \mathbf{u}(k-2) \quad (6.33)$$

onde

$$\begin{aligned} A_0 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{y}(k) = \begin{bmatrix} S_4(k) \\ P_4(k) \\ X_4(k) \end{bmatrix} \\ A_1 &= \begin{bmatrix} -0.9891 & -0.0594 & -0.0228 \\ -0.0084 & -0.9460 & 0.0046 \\ 0.0013 & 0.0069 & -0.9548 \end{bmatrix} \quad \mathbf{y}(k-1) = \begin{bmatrix} S_4(k-1) \\ P_4(k-1) \\ X_4(k-1) \end{bmatrix} \end{aligned}$$

6.3 Estratégias de Identificação para o Processo de Fermentação Contínua

$$B_1 = \begin{bmatrix} 0.1144 & -0.0190 & -0.0155 \\ -0.0812 & 0.0254 & -0.0061 \\ -0.0096 & 0.0061 & 0.0493 \end{bmatrix} \quad \mathbf{u}(k-1) = \begin{bmatrix} Fa(k-1) \\ S_0(k-1) \\ R(k-1) \end{bmatrix}$$

$$B_2 = \begin{bmatrix} -0.0655 & 0 & 0 \\ 0.0412 & 0 & 0 \\ 0.0045 & 0 & 0 \end{bmatrix} \quad \mathbf{u}(k-2) = \begin{bmatrix} Fa(k-2) \\ S_0(k-2) \\ R(k-2) \end{bmatrix}$$

Apresenta-se a seguir o desempenho do modelo ARX MIMO frente aos dados de validação. Pode-se verificar através das figuras 6.20, 6.21 e 6.22, que o modelo linear forneceu um desempenho sofrível para as simulações recursivas, embora os resultados para as previsões um passo adiante tenham sido bons.

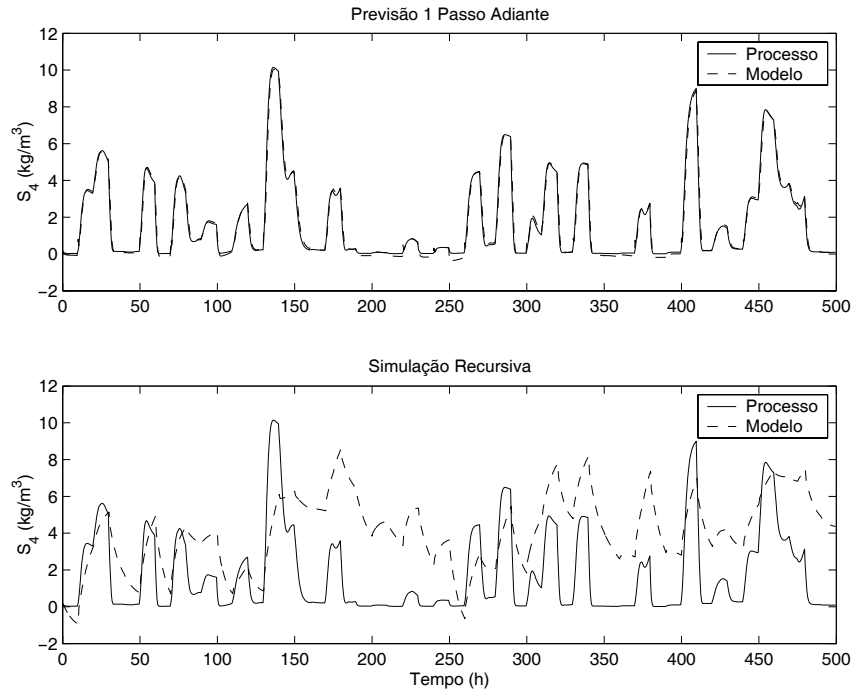


Figura 6.20: Desempenho do modelo ARX para a variável S_4 .

A razão para este comportamento está relacionada com a teoria de identificação de sistemas dinâmicos, onde a validação de um modelo não é tão trivial quanto para sistemas estáticos (cujo objetivo é, de fato, minimizar o erro quadrático médio da

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

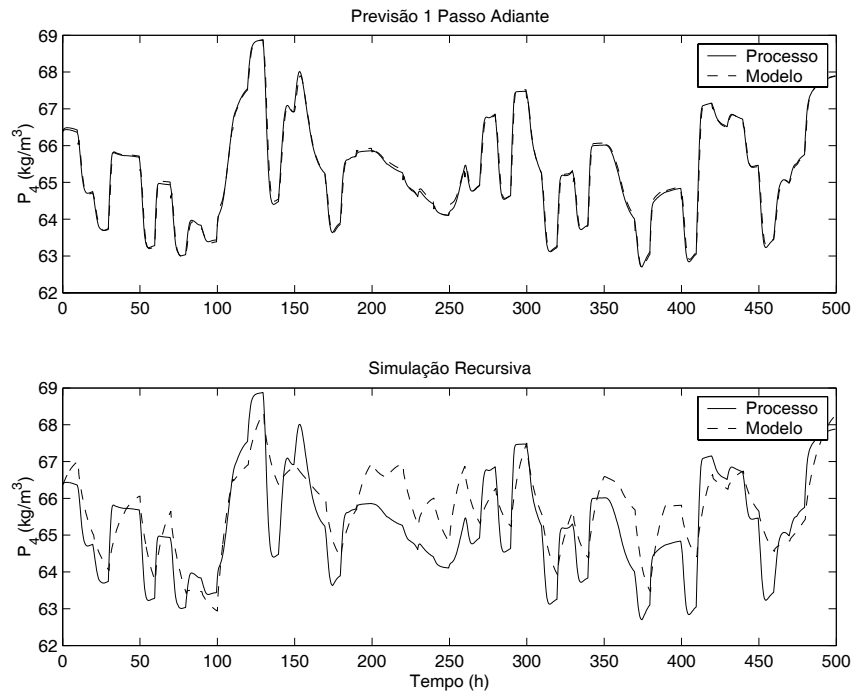


Figura 6.21: Desempenho do modelo ARX para a variável P_4 .

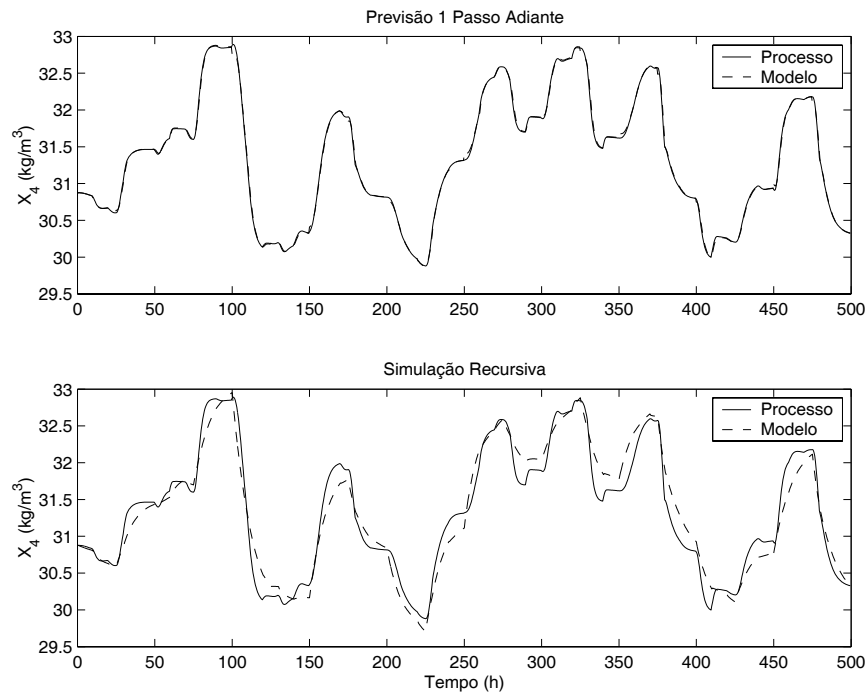


Figura 6.22: Desempenho do modelo ARX para a variável X_4 .

6.3 Estratégias de Identificação para o Processo de Fermentação Contínua

aproximação de uma dada superfície). Os bons resultados apresentados pelo modelo ARX para as previsões um passo adiante ocorrem porque o critério de validação utilizado é o mesmo adotado para a estimação do modelo, ou seja, a minimização do erro associado à previsão das saídas um passo adiante. Cumpre destacar a importância da verificação do comportamento do modelo em simulações recursivas, principalmente nos casos onde o modelo é utilizado em estratégias de controle que necessitam da previsão da saída do processo em horizontes relativamente longos.

Outro ponto que merece destaque é que o algoritmo dos mínimos quadrados levou a obtenção de pólos associados às saídas atrasadas com valores muito elevados (elementos $a_{11} = 0.9891$, $a_{22} = 0.9460$ e $a_{33} = 0.9548$ da matriz A_1). Estes pólos indicam que toda a confiança do modelo está depositada no valor anterior da saída, $\mathbf{y}(k-1)$. Este valor está disponível na previsão um passo adiante mas não previsão recursiva. Isto significa dizer que a qualidade de previsão do modelo é quase que totalmente dependente da informação dos valores reais da saída no instante anterior, fato que explica o desempenho inadequado do modelo em previsões de longo alcance. Além disso, pólos com valores tão elevados representam uma dinâmica muito lenta e, neste caso especificamente, representam uma dinâmica muito mais lenta do que a do sistema real (conforme comprovado nos experimentos descritos nas sub-seções anteriores). O fato desses pólos não representarem sequer a dinâmica dominante do sistema é a principal causa do modelo não apresentar um desempenho aceitável nas simulações recursivas.

6.4 Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável

Apesar das várias vantagens apresentadas pelo etanol como combustível alternativo, ou mesmo como substituto para a gasolina e o óleo diesel (seções 6.1 e 6.2.1), ainda há um grande interesse na otimização de todas as etapas de seu processo produtivo. Existem basicamente duas abordagens relatadas na literatura para resolver este problema. Uma das opções está relacionada com modificações no processo produtivo, destacando-se as estratégias de reatores com células imobilizadas, como por exemplo o trabalho de VIEGAS *et al.* (2002), entre outros, e os processos contínuos com retirada contínua de etanol do meio reacional durante o processo produtivo (Capítulo 7). A outra opção envolve propostas de estratégias de controle mais robustas e eficientes, capazes de aumentar a produtividade sem comprometer o rendimento do processo, utilizando o mínimo de insumos e, de preferência, aproveitando o *layout* e a capacidade instalada de uma planta já existente. No entanto, os processos biotecnológicos caracterizam-se por seu comportamento dinâmico complexo, exibindo fortes não linearidades, resposta inversa e tempo morto. Este comportamento é devido, principalmente, da natureza do metabolismo microbiano, que apresenta taxas cinéticas altamente sensíveis às variações das condições operacionais. Desse modo, o controle de processos biotecnológicos não é uma tarefa trivial, não havendo ainda uma metodologia totalmente estabelecida.

Os algoritmos de controle preditivo baseados em modelo, MPC⁴ (CLARKE *et al.*, 1987a; CLARKE *et al.*, 1987b; CLARKE, 1994), por sua vez, têm apresentado bons desempenhos em aplicações industriais e por este motivo vêm sendo cada vez mais utilizados nos últimos anos (SOETERBOEK, 1992; CAMACHO e BORDONS, 1999). Estes algoritmos são bastante apropriados para o desenvolvimento de estratégias de controle de alto desempenho, pois são adequados para a aplicação direta em processos multi-variáveis, não necessitam do acoplamento explícito entre as variáveis manipuladas e controladas, além de oferecerem a possibilidade de incorporar restrições nas variáveis de entrada e saída do processo diretamente no projeto do controlador (HENSON, 1998; MORARI *et al.*, 1994; MORARI e LEE, 1999). Por estes motivos, este tipo de controlador desponta como uma solução promissora para os problemas apresentados pelos processos

⁴Model Predictive Control.

6.4 Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável

biotecnológicos.

Não obstante, na maioria das aplicações industriais dos MPCs, o controlador ainda é implementado utilizando modelos lineares do processo. Contudo, os processos da indústria química são predominantemente não lineares e, nestes casos, o uso de MPCs baseados em modelos lineares pode levar a desempenhos sofríveis em malha fechada. De acordo com PICHÉ *et al.* (2000), o uso de MPCs baseados em modelos lineares somente é aceitável quando o processo opera em torno de um único *set-point* e o controlador é utilizado principalmente para rejeitar perturbações. Como a maioria dos processos não operam em torno de um único estado estacionário, o desenvolvimento e o uso de algoritmos de controle preditivo não lineares é cada vez mais recomendável.

Recentemente, tem havido um grande interesse na aplicação de redes neurais artificiais (RNA) na modelagem e controle de processos químicos. Isto é fruto principalmente da conhecida capacidade das RNA de aproximar mapeamentos não lineares com precisão arbitrária. Diferentes estruturas de redes neurais têm sido utilizadas como modelos não lineares dos processos em estratégias de controle avançado. Algumas destas propostas podem ser encontradas em NARENDRA e PARTHASARATHY (1990), BAUGHMAN e LIU (1995), MILLS *et al.* (1996), ŻBIKOWSKI e HUNT (1996), NG (1997), HAGAN e DEMUTH (1999), NØRGAARD *et al.* (2000), MUJTABA e HUSSAIN (2001), sem mencionar a miríade de teses e artigos relacionados com o tema que são submetidos anualmente em congressos e periódicos. HUSSAIN (1999), por exemplo, em sua revisão sobre as aplicações de redes neurais artificiais em controle de processos, concluiu que as RNA são capazes de descrever vários processos químicos operando em um grande número de situações, além de comprovar a sua versatilidade para ser incorporada em diferentes estratégias de controle não linear.

Embora o uso de modelos não lineares possa melhorar o desempenho do controlador, a identificação de tais modelos e o projeto deste tipo de controlador nem sempre são tarefas simples. Faz-se necessário sistematizar este procedimento de tal forma que o processo de identificação de modelos não lineares e o subsequente projeto do controlador não linear possam ser realizados da maneira mais eficiente possível.

Conforme mencionado na seção 6.2.1, a concentração de açúcares redutores totais (ART) na saída da quarta dorna tem um forte impacto sobre a viabilidade econômi-

ca do processo (ANDRIETTA, 1994; DECHECHI, 1998). Por este motivo, os controladores desenvolvidos foram projetados para: 1) manter constante a concentração de ART na saída da quarta dorna (problema regulador) ou, 2) fazer com que o valor desta variável siga uma determinada trajetória estabelecida (problema servo).

A seguir, serão apresentadas a formulação geral do controlador preditivo não linear utilizado neste capítulo e as etapas descrevendo o projeto e a aplicação deste tipo de controlador utilizando os modelos não lineares HNN, MLP e UGL. O desempenho dos MPCs não lineares serão comparados entre si e com o controlador DMC⁵, versão linear clássica dos controladores preditivos baseados em modelo. Para facilitar a comparação, todos os controladores serão submetidos às mesmas seqüências de *set-points* e perturbações.

6.4.1 Formulação Geral do Controlador Preditivo Não Linear Mono-Variável

O objetivo de um controlador preditivo não linear do tipo MPC, é calcular um conjunto de ações de controle futuras de forma a minimizar a seguinte função objetivo:

$$J = \sum_{i=N_1}^{N_y} (\hat{y}_{k+i}^c - w_{k+i})^2 + \lambda \sum_{i=1}^{N_u} (\Delta u_{k+i-1})^2 \quad (6.34)$$

onde λ é o fator de ponderação (parâmetro de projeto que penaliza as ações de controle), Δu são os incrementos na variável manipulada, N_1 é o horizonte inicial, N_y é o horizonte de predição, N_u é o horizonte de controle, w é a trajetória de referência (*set-points*) e \hat{y}^c é a previsão do modelo (HNN, MLP ou UGL) corrigida. Esta predição é obtida recursivamente até um total de N_y predições futuras, que são corrigidas da seguinte forma:

$$\hat{y}_{k+i}^c = \hat{y}_{k+i} + d_k, \quad (i = 1, \dots, N_y) \quad (6.35)$$

⁵*Dynamic Matrix Control.*

6.4 Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável

Na equação (6.35), \hat{y}_{k+i} é a predição feita pelo modelo para a saída controlada do processo no instante de amostragem futuro, $k+i$, e d_k é o termo de correção dado por:

$$d_k = y_k - \hat{y}_k \quad (6.36)$$

onde y_k é a saída medida do processo no instante de amostragem presente e \hat{y}_k é a respectiva predição do modelo (calculada no instante de amostragem anterior).

O algoritmo de otimização calcula os incrementos, Δu , para a variável manipulada de modo a minimizar a função objetivo (6.34). As ações de controle futuras são então derivadas a partir dos incrementos ótimos de controle:

$$u_{k+i} = u_{k+i-1} + \Delta u_{k+i}, \quad (i = 0, \dots, N_u - 1) \quad (6.37)$$

É importante observar que somente as primeiras N_u ações de controle são otimizadas, enquanto que as demais são mantidas constantes, ou seja:

$$u_{k+i} = u_{k+N_u-1}, \quad (i = N_u, \dots, N_y - 1) \quad (6.38)$$

O problema de otimização também está sujeito às seguintes restrições:

$$y_{min} \leq \hat{y}_{k+i} \leq y_{max}, \quad (i = 1, \dots, N_y) \quad (6.39)$$

$$u_{min} \leq u_{k+i-1} \leq u_{max}, \quad (i = 1, \dots, N_u) \quad (6.40)$$

$$|u_{k+i-1} - u_{k+i-2}| \leq \Delta u_{max}, \quad (i = 1, \dots, N_u) \quad (6.41)$$

onde y_{min} e y_{max} , u_{min} e u_{max} e Δu_{min} e Δu_{max} são os limites superiores e inferiores para y , u e Δu , respectivamente.

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

Finalmente, a estratégia do “Horizonte Móvel”⁶ é adotada (GARCIA *et al.*, 1989; SOETERBOEK, 1992; CAMACHO e BORDONS, 1999), onde somente a primeira ação de controle u_k , é implementada e o problema de otimização é resolvido novamente em cada instante de amostragem. Devido às características não lineares das equações que governam as predições dos modelos, o problema de otimização (com restrições) é não convexo e, desta forma, uma metodologia apropriada deve ser empregada para resolvê-lo. A técnica utilizada nesta tese para resolver este tipo de problema de otimização foi a *Programação Quadrática Sucessiva* - SQP (EDGAR e HIMMELBLAU, 1988).

Considerando o processo contínuo de fermentação alcoólica, as restrições operacionais impostas às variáveis manipulada, $u = Fa$, e controlada, $y = S_4$, do processo são dadas por:

- Restrições na Variável Controlada:

$$S_4(k+i) \geq 0, \quad (i = 1, \dots, N_y) \quad (6.42)$$

- Restrições na Variável Manipulada:

$$50 \leq Fa(k+i) \leq 150, \quad (i = 0, \dots, N_u - 1) \quad (6.43)$$

- Restrições nas Ações de Controle:

$$| Fa(k+i) - Fa(k+i-1) | \leq 50, \quad (i = 0, \dots, N_u - 1) \quad (6.44)$$

⁶*Receding Horizon.*

6.4 Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável

6.4.2 Controle Preditivo Não Linear Baseado no Modelo HNN: MPC-HNN

O modelo HNN (para a variável S_4) desenvolvido na seção 6.3.6 foi utilizado como modelo não linear para o controlador preditivo descrito na seção 6.4.1. A seguir, será apresentado o desempenho do controlador MPC-HNN para os problemas servo e regulador. Conforme foi discutido na seção 6.3.1, as perturbações que agem sobre o processo são a concentração de ART no mosto, S_0 , e a taxa de reciclo, R . Segundo ANDRIETTA (1994) e DECHECHI (1998), estas perturbações influenciam fortemente a dinâmica do processo.

Inicialmente será considerado que não há perturbações agindo sobre o processo, ou seja, S_0 e R são mantidos em seus valores estacionários de referência ($S_0 = 180 \text{ kg/m}^3$ e $R = 0.3$). Os parâmetros do controlador foram sintonizados da seguinte forma: $N_1 = 1$, $N_y = 10$, $N_u = 2$ e $\lambda = 0,001$.

A figura 6.23 apresenta o bom desempenho do controlador (e as respectivas ações de controle) frente a várias mudanças de *set-point* quando o processo era operado nas condições descritas acima.

Na sequência, apresentam-se os resultados considerando a presença de perturbações agindo sobre o processo. No primeiro experimento, a concentração de ART no mosto, S_0 , ainda é considerada constante enquanto a taxa de reciclo, R , é variada (figura 6.24). O resultado da simulação é apresentado na figura 6.25, que mostra que o controlador foi capaz de rejeitar as perturbações de forma bastante eficiente.

Nos experimentos seguintes, as simulações anteriores são repetidas considerando que S_0 é uma perturbação estocástica variando em torno de três níveis (estados estacionários) diferentes (figura 6.26). Os resultados correspondentes são apresentados nas figuras 6.27, 6.28, 6.29 (problema servo). A figura 6.30 ilustra a capacidade do controlador rejeitar as perturbações (problema regulador) somente para valores de S_0 variando estocasticamente em torno do estado estacionário de referência ($S_0 = 180 \text{ kg/m}^3$), já que o seu desempenho, assim como nos problemas servo, também foi satisfatório para os outros níveis de S_0 .

Cumprе destacar que, neste estudo de caso, os controladores têm ação *feed-*

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

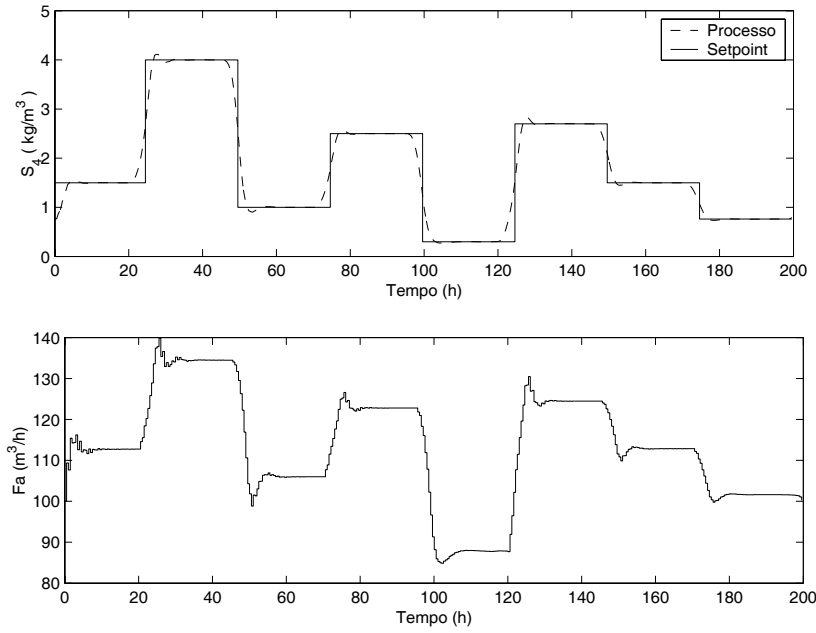


Figura 6.23: Controlador MPC-HNN - Problema Servo: Referência e saída do processo (acima); variável manipulada (abaixo).

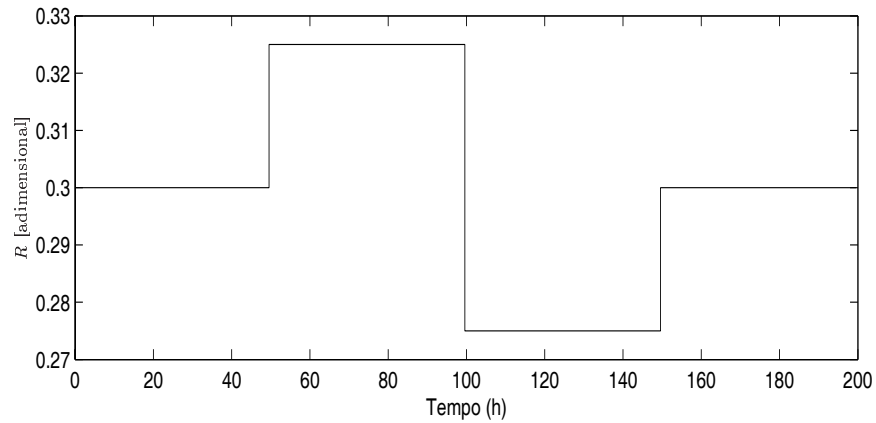


Figura 6.24: Perturbações na taxa de reciclo.

forward pois as perturbações (S_0 e R) são medidas.

Conforme era esperado, a presença de perturbações degradou o desempenho do controlador. Não obstante, o esquema de controle ainda foi capaz de controlar o processo adequadamente.

6.4 Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável

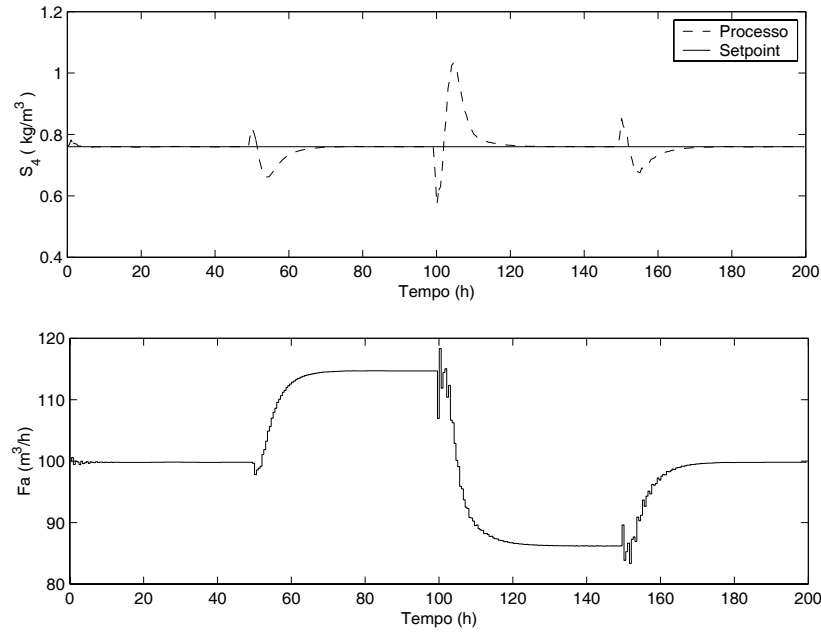


Figura 6.25: Controlador MPC-HNN - Problema Regulador: Referência e saída do processo (acima); variável manipulada (abaixo).

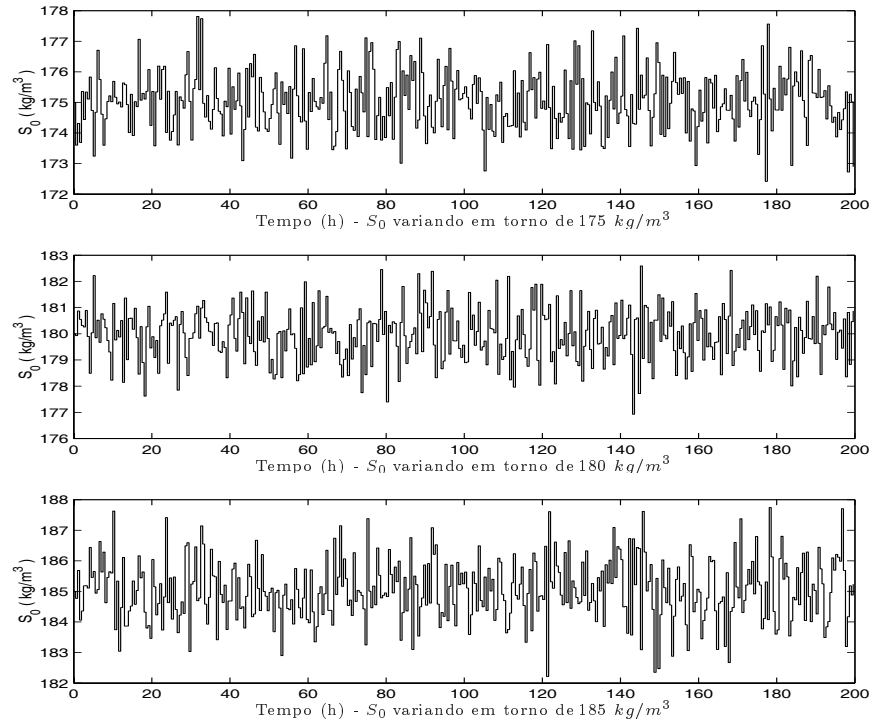


Figura 6.26: Perturbações na concentração de S_0 .

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

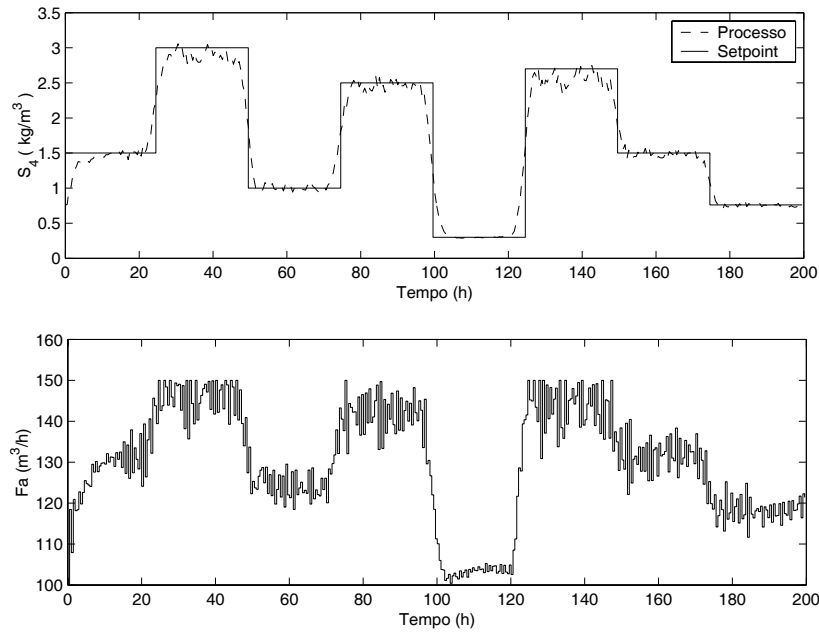


Figura 6.27: Controlador MPC-HNN - Problema Servo (S_0 variando em torno de 175 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).

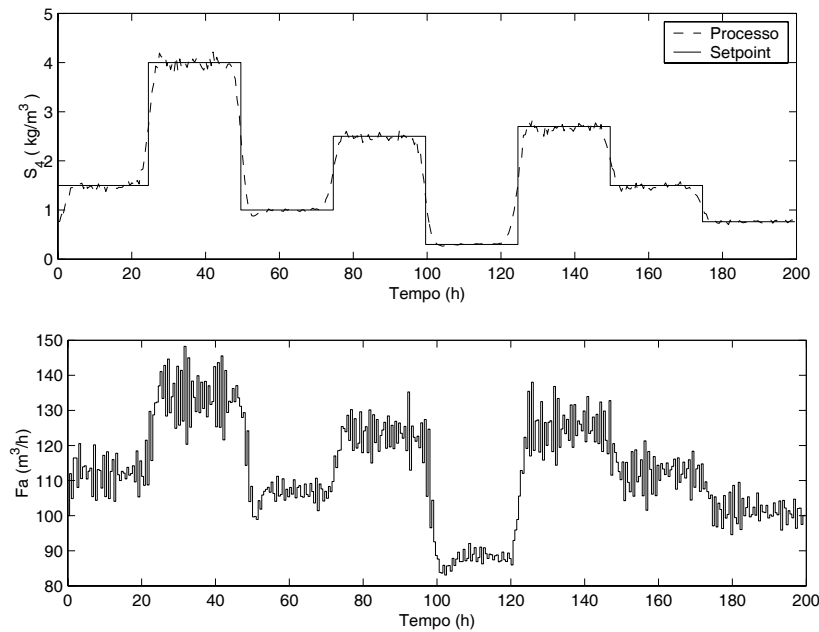


Figura 6.28: Controlador MPC-HNN - Problema Servo (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).

6.4 Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável

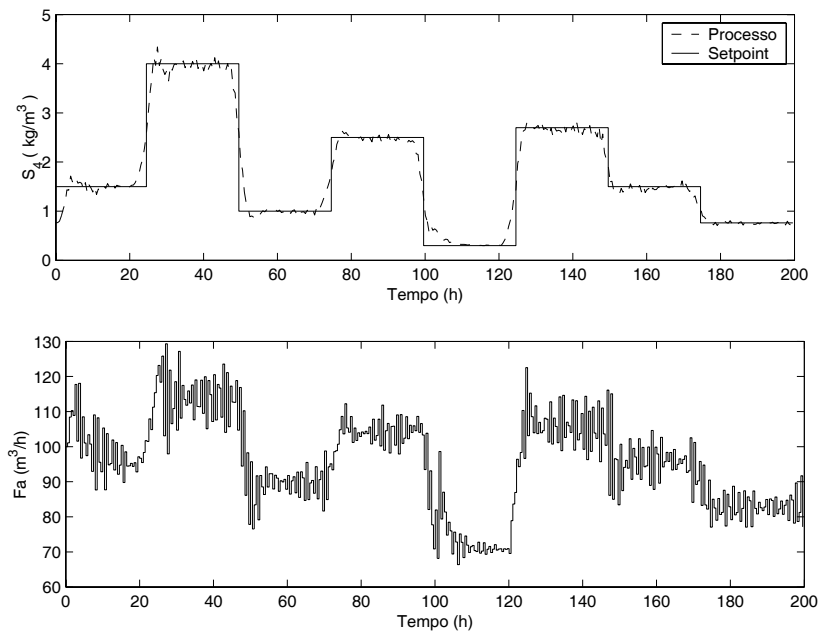


Figura 6.29: Controlador MPC-HNN - Problema Servo (S_0 variando em torno de 185 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).

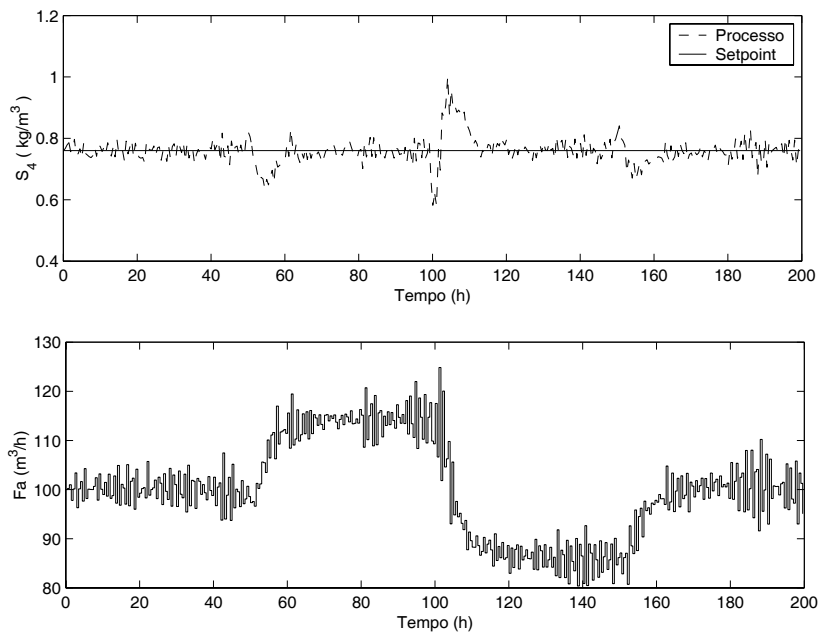


Figura 6.30: Controlador MPC-HNN - Problema Regulador (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).

6.4.3 Controle Preditivo Não Linear Baseado no Modelo MLP: MPC-MLP

Nesta seção são apresentados os resultados obtidos quando o modelo MLP (seção 6.3.7) é utilizado no algoritmo do controlador preditivo não linear. Conforme destacado na introdução da seção 6.4, todos os controladores foram submetidos às mesmas condições operacionais (*set-points* e perturbações) e às mesmas restrições (equações 6.42 a 6.44). Os parâmetros do controlador MPC-MLP foram sintonizados em $N_1 = 1$, $N_y = 5$, $N_u = 2$ e $\lambda = 0,005$.

A figura 6.31 ilustra o bom desempenho do controlador MPC-MLP na tarefa de fazer com que a variável controlada (S_4) siga uma seqüência pré-estabelecida de valores de referência (problema servo).

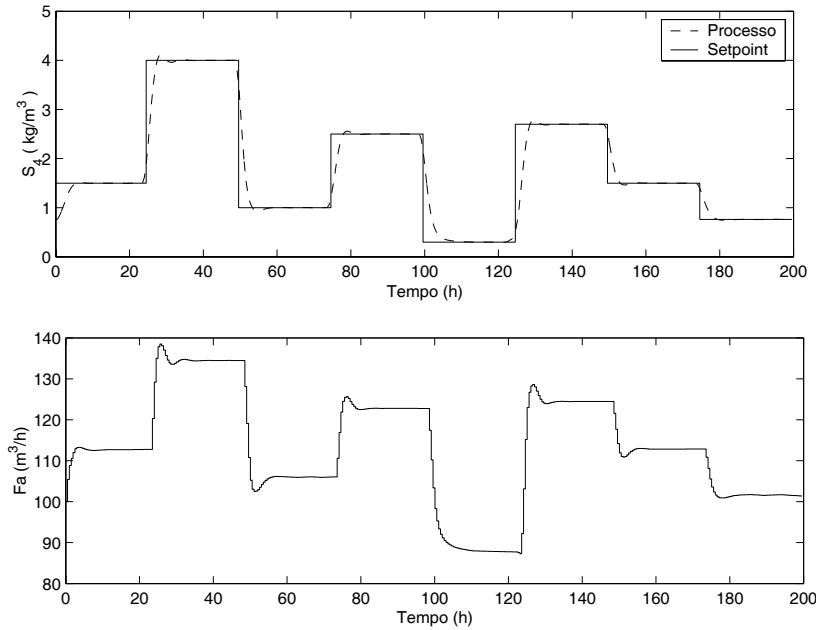


Figura 6.31: Controlador MPC-MLP - Problema Servo: Referência e saída do processo (acima); variável manipulada (abaixo).

A capacidade do controlador MPC-MLP rejeitar perturbações impostas na taxa de reciclo de microorganismos, R , foi testada empregando-se os mesmos valores apresentados na figura 6.24. O resultado do problema regulador é mostrado na figura 6.32.

6.4 Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável

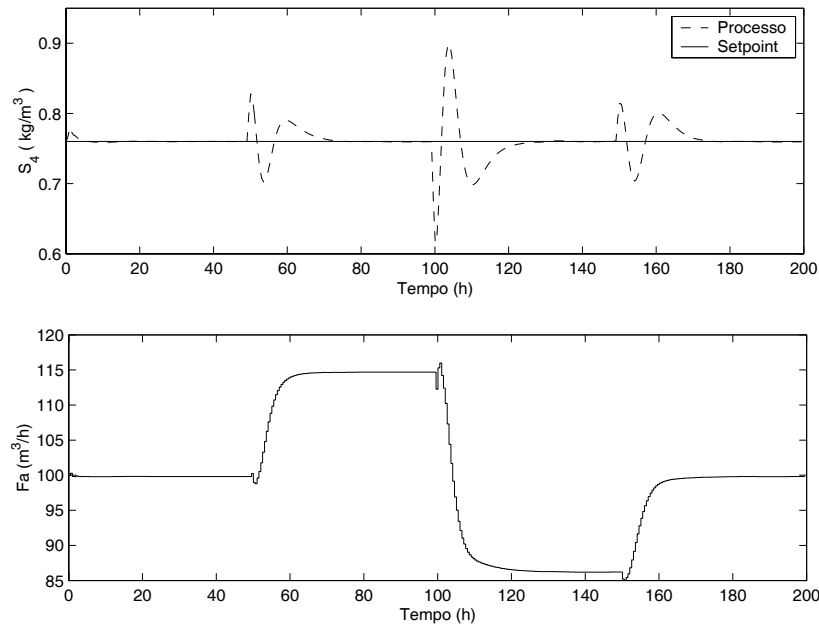


Figura 6.32: Controlador MPC-MLP - Problema Regulador: Referência e saída do processo (acima); variável manipulada (abaixo).

Comparando estes resultados com aqueles apresentados pelo controlador MPC-HNN (figuras 6.23 e 6.25), verifica-se que o desempenho apresentado por ambos é semelhante, embora o controlador MPC-HNN apresente sobre-elevações (“overshoot”) levemente superiores para o caso servo, no entanto, nada que possa comprometer o seu desempenho. Da mesma forma, as ações de controle impostas por ambos os controladores no cenário supervisorório foram bastante similares, não apresentando variações bruscas entre dois instantes de amostragem consecutivos. Resultados similares também foram obtidos em relação ao desempenho exibido pelos dois controladores no problema regulatório.

As figuras 6.33 e 6.34 apresentam os resultados de simulações da planta nas mesmas condições anteriores (problemas servo e regulador), com a diferença de que agora a variável S_0 é uma perturbação estocástica, tendo seus valores variando aleatoriamente em torno do estado estacionário de referência ($S_0 = 180 \text{ kg/m}^3$). O controlador MPC-MLP mostrou-se capaz de rejeitar adequadamente este tipo de perturbação em ambas as situações.

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

Comparando estes resultados com aqueles apresentados pelo controlador MPC-HNN (figuras 6.28 e 6.30), percebe-se que o controlador MPC-MLP apresentou um desempenho superior, sobretudo no problema servo, pois além da saída do processo (S_4) oscilar menos, as ações de controle impostas pelo algoritmo foram mais suaves, o que sempre é desejável em ambientes industriais. Resultados semelhantes, embora menos acentuados, também foram verificados no problema regulador (figuras 6.30 e 6.34).

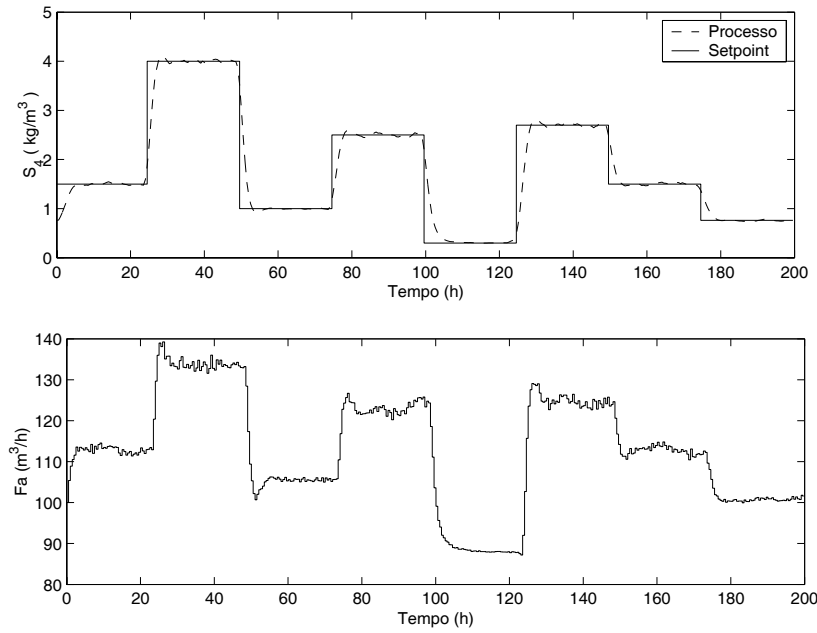


Figura 6.33: Controlador MPC-MLP - Problema Servo (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).

6.4 Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável

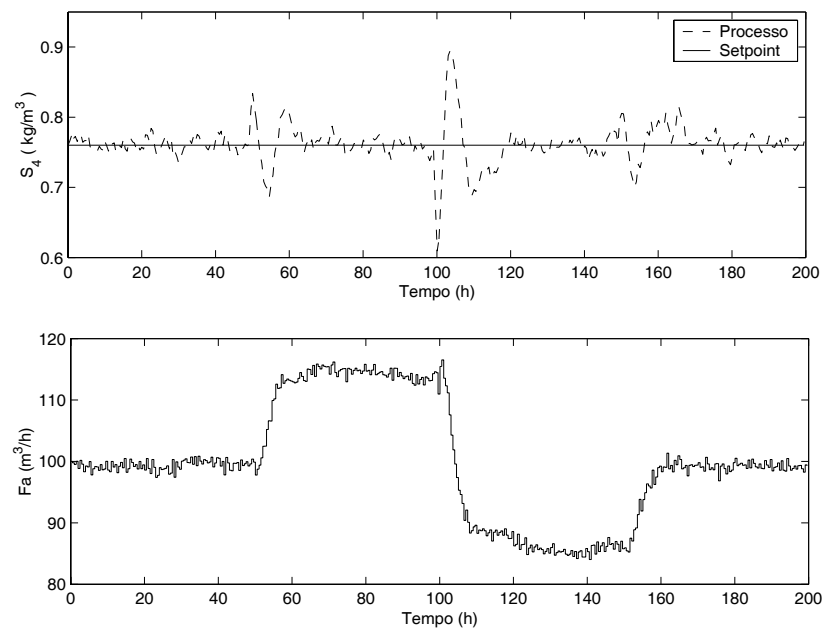


Figura 6.34: Controlador MPC-MLP - Problema Regulador (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).

6.4.4 Controle Preditivo Não Linear Baseado no Modelo UGL: MPC-UGL

A seguir são apresentados os resultados obtidos quando o modelo UGL (seção 6.3.8) foi utilizado no algoritmo do controlador preditivo não linear (seção 6.4.1). O controlador MPC-UGL foi submetido aos mesmos testes descritos nas seções anteriores (6.4.2 e 6.4.3) e, de uma maneira geral, o seu desempenho foi muito semelhante ao desempenho do controlador MPC-MLP e, conseqüentemente, ligeiramente superior ao desempenho exibido pelo controlador MPC-HNN. A sintonia adequada para o MPC-UGL foi: $N_1 = 1$, $N_y = 6$, $N_u = 2$ e $\lambda = 0,002$.

As figuras 6.35 a 6.38 ilustram o desempenho do controlador MPC-UGL projetado para controlar a variável S_4 em cenários supervisórios e regulatórios, com e sem a presença de perturbações. A análise destas figuras permite concluir que este controlador mostrou-se adequado para lidar com todas as situações de operação do processo propostas.

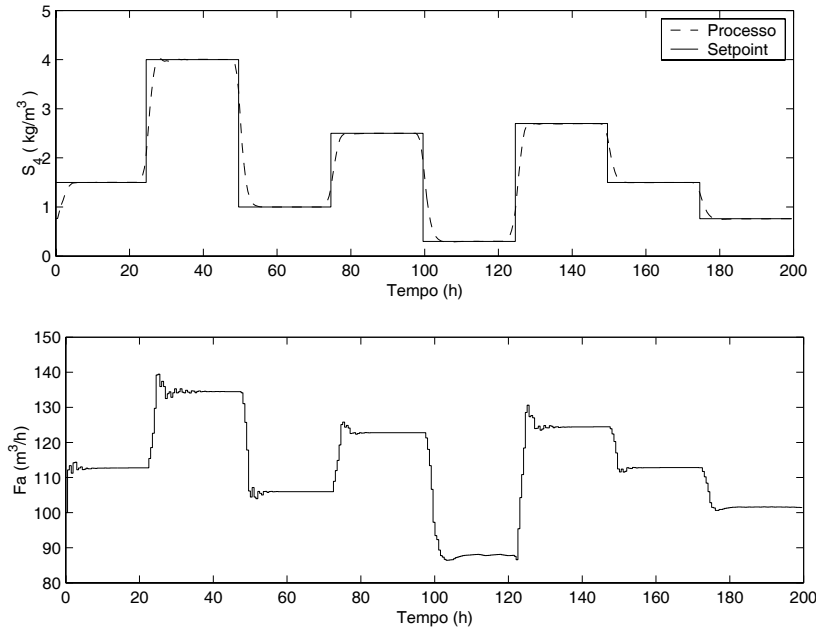


Figura 6.35: Controlador MPC-UGL - Problema Servo: Referência e saída do processo (acima); variável manipulada (abaixo).

O objetivo principal de controle proposto neste capítulo foi controlar a concen-

6.4 Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável

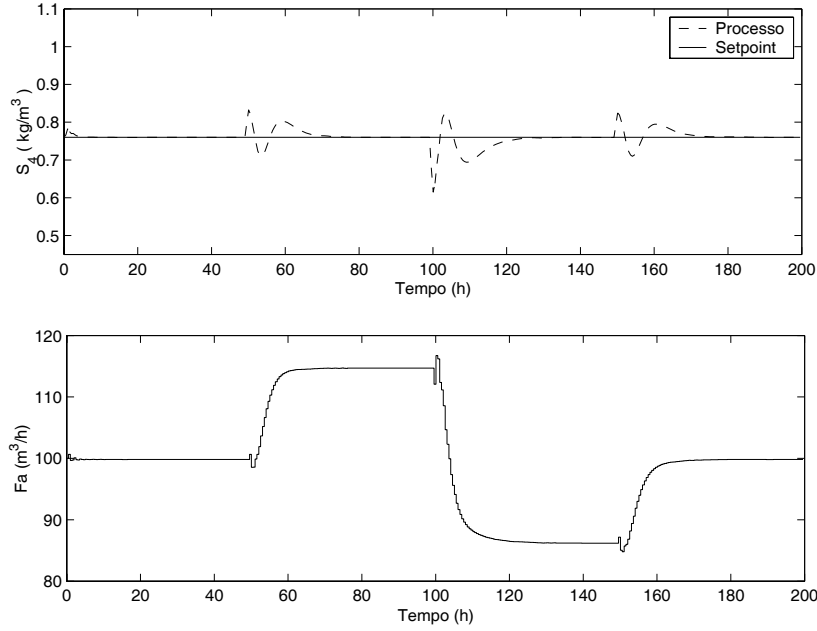


Figura 6.36: Controlador MPC-UGL - Problema Regulador: Referência e saída do processo (acima); variável manipulada (abaixo).

tração de açúcares redutores totais na saída da quarta dorna de fermentação, S_4 . No entanto, em algumas situações, pode ser necessário realizar o controle diretamente na principal variável de processo, a concentração de etanol na saída do processo, P_4 . Embora as condições pré-otimizadas de operação do processo (fornecidas por ANDRIETTA (1994)) limitem a faixa de variação desta variável, decidiu-se ilustrar este problema de controle utilizando o controlador MPC-UGL, que apresentou o melhor desempenho para esta variável. A sintonia adequada para o MPC-UGL neste caso foi: $N_1 = 1$, $N_y = 10$, $N_u = 1$ e $\lambda = 0,0005$.

As figuras 6.39 a 6.42 ilustram o bom desempenho exibido por este controlador, inclusive na presença de ruídos aleatórios na concentração de S_0 . Deve-se destacar, no entanto, que a taxa de reciclo de microorganismos, R , exerce um forte impacto sobre a dinâmica do processo, dificultando o controle regulatório desta variável quando esta variável se encontra em níveis muito acima de seu valor de referência, $R = 0.3$ (DECHECHI, 1998).

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

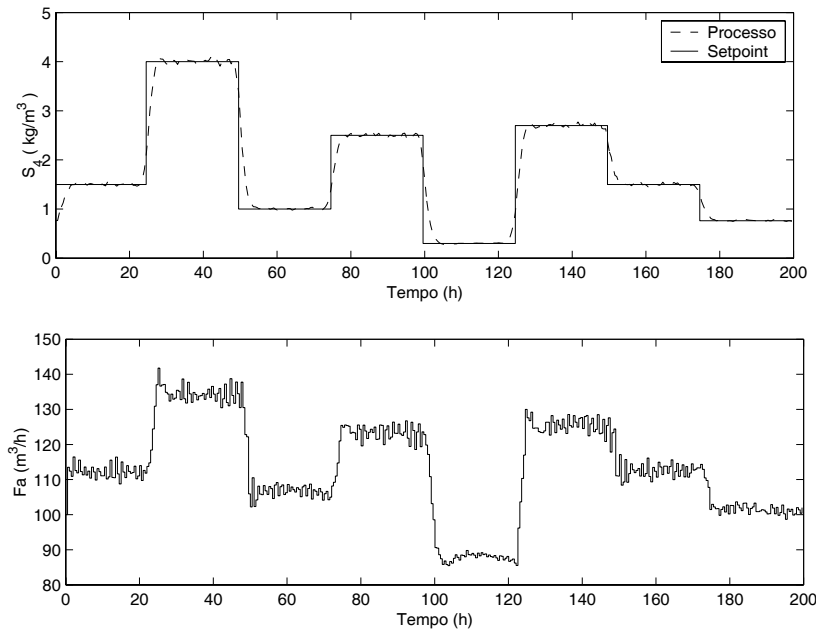


Figura 6.37: Controlador MPC-UGL - Problema Servo (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).

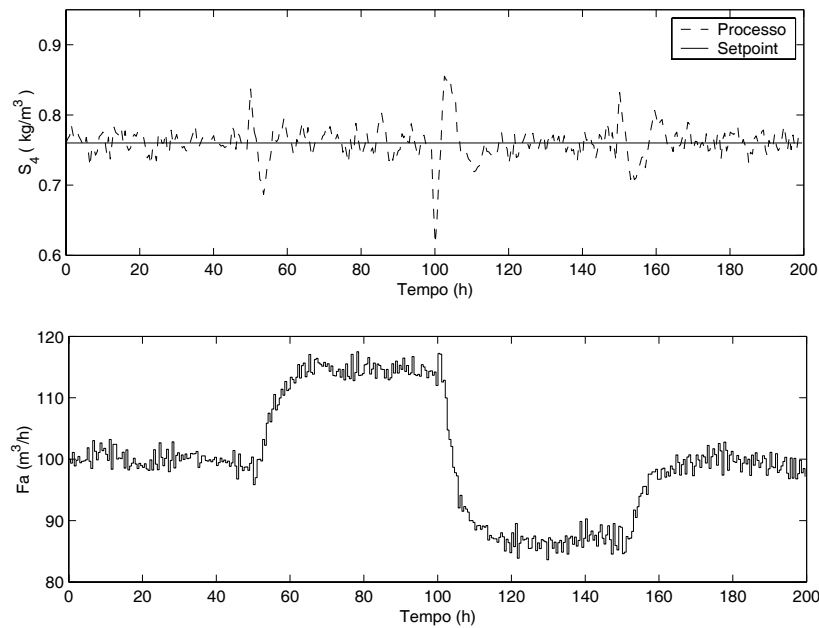


Figura 6.38: Controlador MPC-UGL - Problema Regulador (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).

6.4 Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável

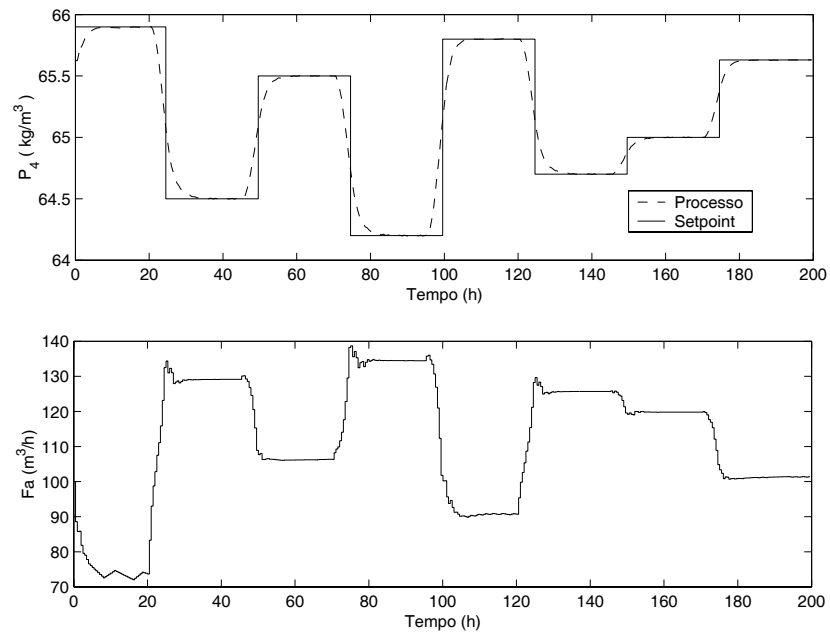


Figura 6.39: Controlador MPC-UGL - Problema Servo para a Concentração de Etanol: Referência e saída do processo (acima); variável manipulada (abaixo).

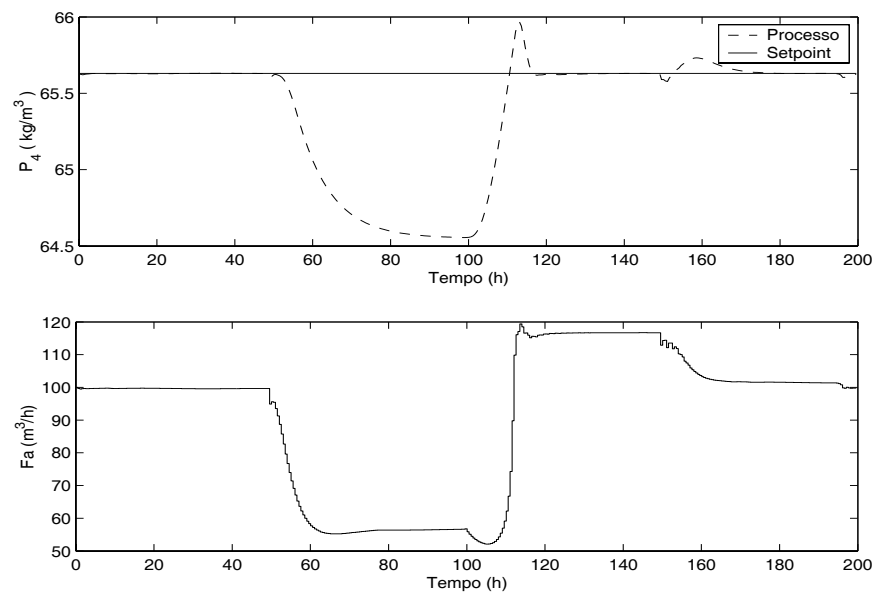


Figura 6.40: Controlador MPC-UGL - Problema Regulador para a Concentração de Etanol: Referência e saída do processo (acima); variável manipulada (abaixo).

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

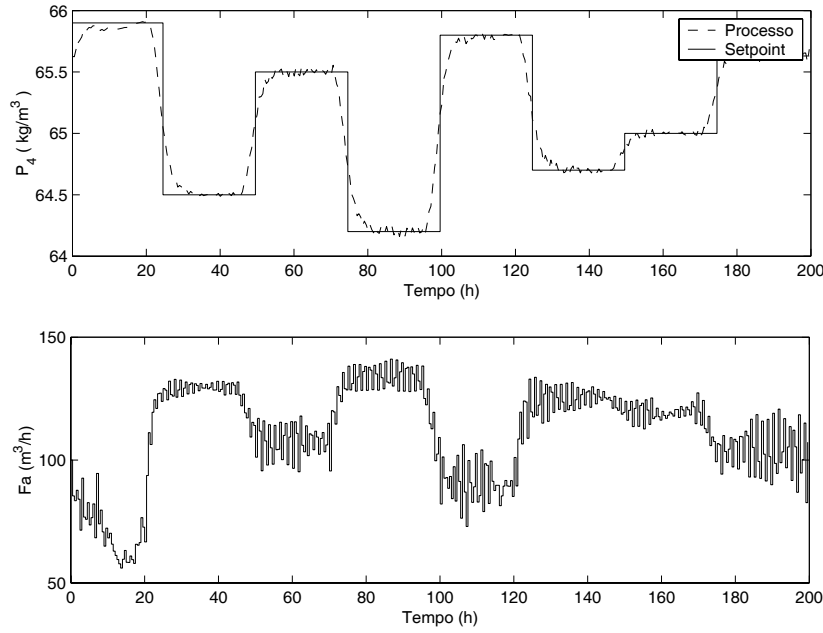


Figura 6.41: Controlador MPC-UGL - Problema Servo para P_4 (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).

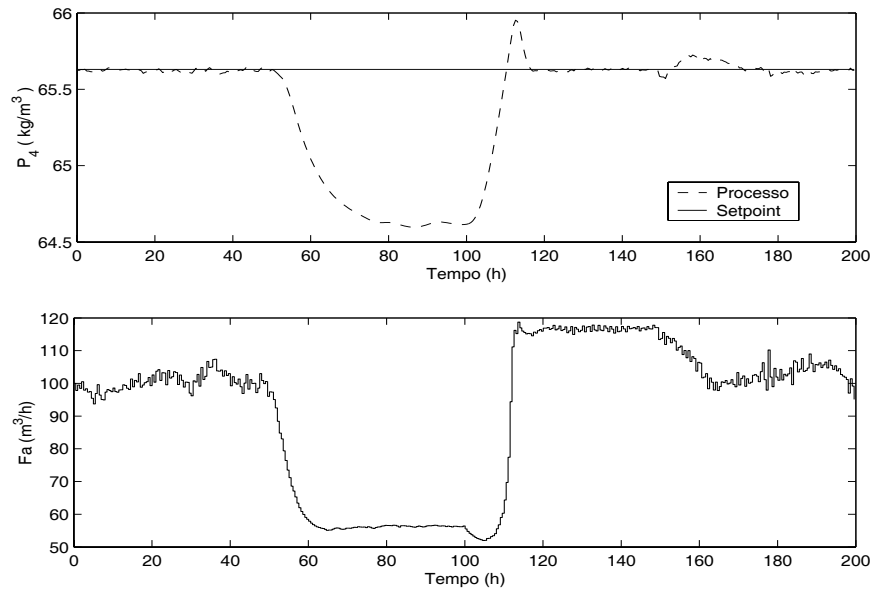


Figura 6.42: Controlador MPC-UGL - Problema Regulador para P_4 (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); var. manipulada (abaixo).

6.4 Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável

6.4.5 Controle Preditivo Linear: O Controlador por Matriz Dinâmica - DMC

Considerando que o controlador DMC (CUTLER e RAMAKER, 1979; CUTLER e RAMAKER, 1980) foi um dos primeiros algoritmos de controle preditivo empregados com sucesso na indústria química e, ainda hoje, é um dos controladores mais utilizados em aplicações industriais (QIN e BADGWELL, 1997; MORARI e LEE, 1999), decidiu-se aplicar este controlador no processo fermentativo para servir como referência e auxiliar na avaliação do desempenho dos controladores MPC não lineares propostos nesta tese (seções 6.4.2, 6.4.3 e 6.4.4).

Apresentam-se a seguir os resultados de controle da planta fornecidos pelo controlador preditivo DMC. A planta foi simulada nas mesmas condições operacionais e submetida às mesmas perturbações descritas nas seções anteriores. As figuras 6.43 a 6.46 apresentam o desempenho do controlador nos cenários servo e regulador, com e sem perturbações estocásticas na concentração de ART no mosto, S_0 .

Os parâmetros do controlador DMC foram sintonizados com os seguintes valores: $N_m = 20$ (horizonte de modelo), $N_u = 1$ (horizonte inicial), $N_y = 5$ (horizonte de predição), $N_c = 1$ (horizonte de controle) e $\lambda = 0,01$ (fator de ponderação).

A análise dos resultados fornecidos pelo controlador DMC permite concluir que este algoritmo também foi capaz de controlar o processo de forma adequada. Contudo, comparando com os resultados apresentados pelos controladores não lineares (MPC-HNN, MPC-MLP e MPC-UGL), o processo respondeu de forma mais lenta quando controlado pelo DMC, principalmente em situações onde havia perturbações na variável S_0 . Cabe ressaltar ainda que o DMC não foi capaz de controlar a planta adequadamente para alguns valores de referência quando S_0 variava em torno de 175 kg/m^3 . Além disso, as ações de controle impostas por este controlador foram mais severas, sobretudo nos problemas servo, o que deve ser evitado sempre que possível em plantas industriais.

ESTUDO DE CASO 1: PROCESSO TRADICIONAL DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

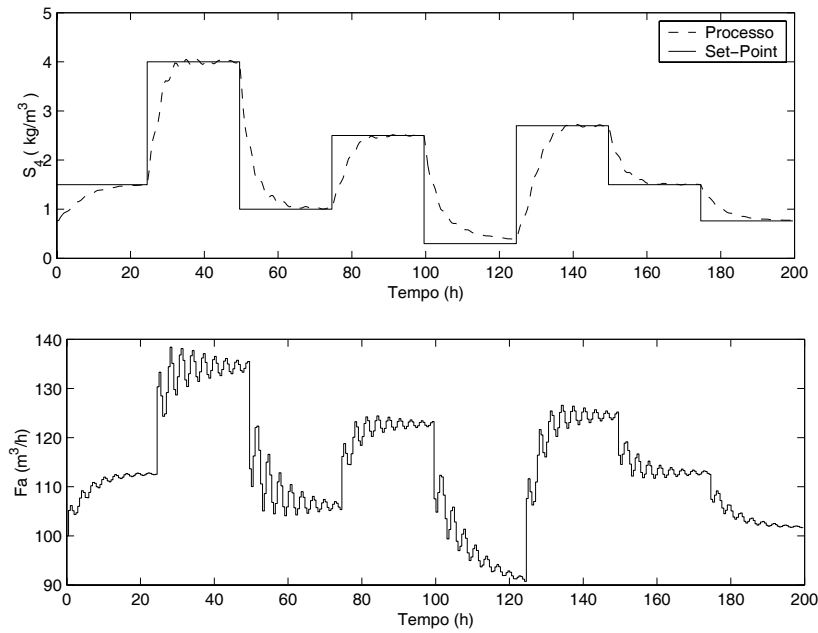


Figura 6.43: Controlador DMC - Problema Servo: Referência e saída do processo (acima); variável manipulada (abaixo).

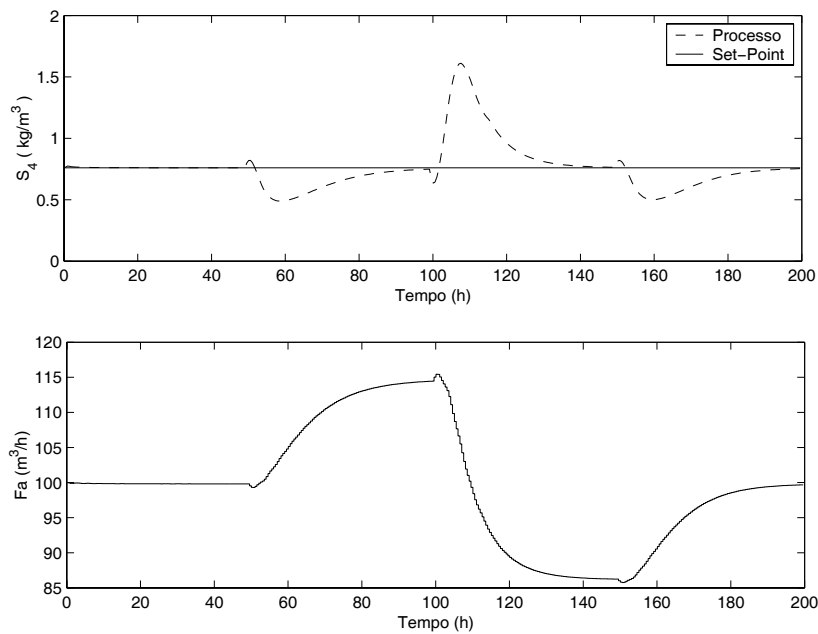


Figura 6.44: Controlador DMC - Problema Regulador: Referência e saída do processo (acima); variável manipulada (abaixo).

6.4 Estratégias de Controle para o Processo de Fermentação Contínua: Caso Mono-Variável

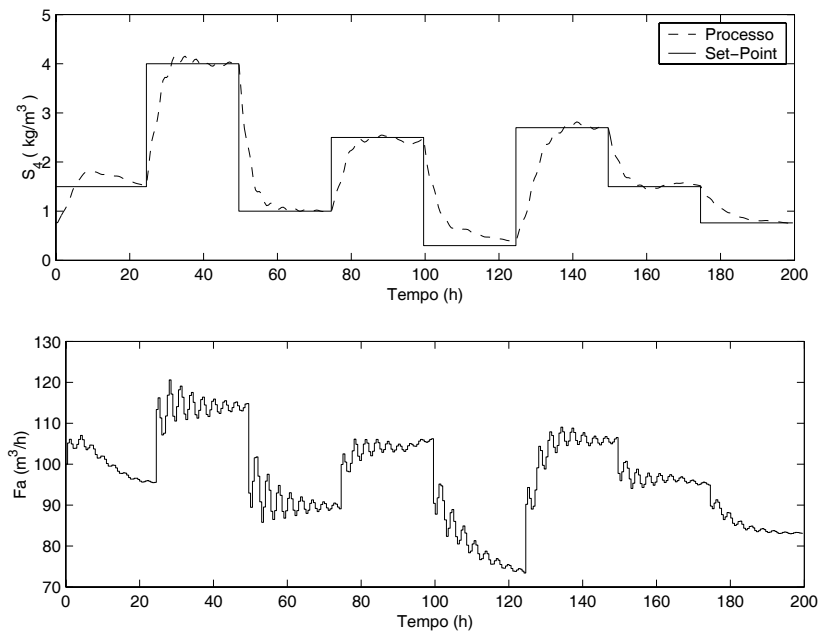


Figura 6.45: Controlador DMC - Problema Servo (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).

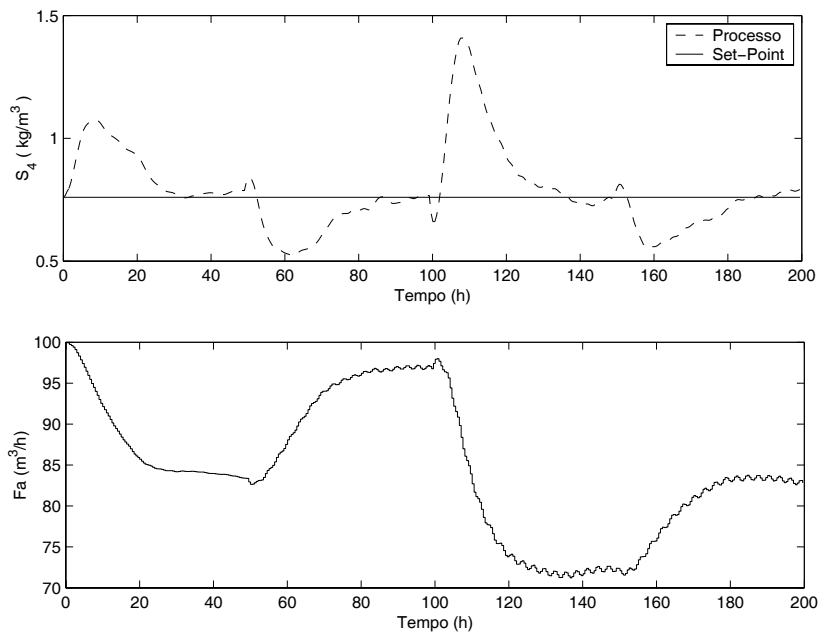


Figura 6.46: Controlador DMC - Problema Regulador (S_0 variando em torno de 180 kg/m^3): Referência e saída do processo (acima); variável manipulada (abaixo).

6.5 Conclusões

Apresentaram-se neste capítulo os principais resultados obtidos da aplicação das técnicas não lineares de identificação e controle em um processo de fermentação alcoólica contínuo. As técnicas de identificação utilizando modelos não lineares foram formalizadas no capítulo 4 e o projeto de controladores preditivos baseados naqueles modelos foi formalizado no capítulo 5.

Na seção 6.3, além da descrição do processo, foram discutidas todas as etapas do procedimento de identificação utilizando-se três modelos não lineares (HNN, MLP e UGL) e um modelo linear (ARX). A análise dos resultados deixou claro a importância da verificação do desempenho dos modelos identificados através de outras técnicas além das tradicionais, tais como a análise do erro quadrático médio e previsão um passo adiante. A inspeção visual das previsões dos modelos em simulação recursiva foi decisiva para determinar a superioridade dos modelos não lineares e a inépcia do modelo linear na tarefa de identificação da dinâmica do processo. Os três modelos não lineares foram capazes de representar o comportamento dinâmico do sistema de forma adequada, mostrando ser potencialmente úteis para a utilização em estratégias de controle avançado. Merece destaque nesta seção a formalização do procedimento de identificação de sistemas de grande porte utilizando os novos paradigmas não lineares: As abordagens hierárquica neuro-nebulosa (HNN) e neural com aprendizado construtivo (UGL). A aplicação destes modelos - que são mais parcimoniosos e apresentam capacidade de generalização potencialmente maior do que os modelos não lineares usuais (incluindo as redes neurais e os sistemas *fuzzy* tradicionais) - no projeto de controladores preditivos não lineares é a principal contribuição desta tese.

O projeto de controladores preditivos baseado nos modelos HNN, MLP e UGL foi discutido na seção 6.4. O desempenho destes controladores em malha fechada foi comparado com o desempenho de um controlador linear do tipo DMC. Embora pareça razoável pensar que a relação custo-benefício (computacional e/ou de desenvolvimento-fornecido pelo controlador) seja desfavorável para os controladores não lineares, deve-se considerar que a planta em questão é operada em torno de condições de processo pré-otimizadas (concentração média de S_0 em torno de 180 kg/m^3 , temperaturas das dornas

6.5 Conclusões

de fermentação constantes⁷, etc.). Sob estas condições, os controladores lineares podem ser capazes de controlar a planta de forma aceitável. Em condições operacionais menos adequadas, como por exemplo o caso descrito na seção 6.4.5 onde o valor médio de S_0 caiu para 175 kg/m^3 , a diferença de desempenho apresentada por controladores preditivos baseados em modelos lineares e não lineares fica mais evidente e a complexidade inerente ao projeto e desenvolvimento dos MPCs não lineares passa a ser ainda mais justificada.

As metodologias de identificação de processos e projeto de controladores não lineares propostas nesta tese provaram ser bastante eficientes, pois forneceram bons resultados em todas as situações a que foram submetidas. A mesma estratégia de identificação e controle será testada em outro processo fermentativo (descrito no capítulo 7), nesse caso porém, o problema de controle é multi-variável.

⁷cujo controle é realizado por uma malha a parte.

Capítulo 7

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

Preâmbulo

O estudo de caso apresentado neste capítulo também trata da produção de etanol combustível por via fermentativa. A principal diferença em relação ao problema estudado no capítulo 6, é que neste caso um tanque *flash* é utilizado para extrair continuamente parte do etanol formado e, além disso, somente um fermentador é utilizado. SILVA *et al.* (1999) constataram que o processo extrativo de fermentação alcoólica contínua, entre outras vantagens, apresenta taxas de produtividade até três vezes maior do que o processo fermentativo convencional.

Neste capítulo, serão apresentados os resultados da aplicação de técnicas não lineares de identificação e controle em um processo fermentativo multi-variável. Na seção 7.1, destaca-se a importância do desenvolvimento de novas técnicas que contribuam para o aumento da eficiência do processo de produção do etanol, para torná-lo economicamente mais competitivo em relação aos combustíveis tradicionais. Ainda nesta seção, é feita uma breve revisão das principais propostas envolvendo mudanças na configuração dos equipamentos que possibilite a remoção contínua do etanol, durante a fermentação, à medida em que ele é formado. Esta medida evita o fenômeno conhecido como *inibição*

por produto e, conseqüentemente, aumenta a produtividade do processo fermentativo. Outra medida discutida na seção 7.1 visando o aumento de produtividade, está relacionada com o desenvolvimento de estratégias de controle eficientes, capazes de manter o processo operando em regiões onde os custos de produção sejam minimizados.

O processo fermentativo estudado neste capítulo é descrito na seção 7.2, onde o esquema geral da planta e todas as variáveis de processo são detalhadas. A modelagem matemática (fenomenológica) do processo também é apresentada nesta seção, assim como as constantes físicas, parâmetros cinéticos e parâmetros de processo.

As estratégias de identificação propostas para modelar esta planta, além de todas as etapas envolvidas neste processo, são discutidas na seção 7.3. O processo fermentativo é então identificado utilizando-se dois modelos não lineares (Modelo Neural Convencional - MLP e Modelo Neural Construtivo - UGL) e um modelo linear do tipo ARX, todos utilizando a representação MIMO.

Finalmente, apresentam-se, na seção 7.4, as estratégias de controle preditivo baseadas nos modelos identificados do processo. Os controladores preditivos resultantes (MPC-MLP, MPC-UGL e DMC) são testados em situações de controle multi-variável considerando, simultaneamente, os problemas servo e regulador para as variáveis controladas do processo. Todas as simulações em malha fechada foram feitas na presença de perturbações não modeladas, ou seja, os seus efeitos sobre o comportamento dinâmico do processo não foram considerados em nenhum dos modelos identificados.

7.1 Introdução

Apesar das várias vantagens que o etanol apresenta como combustível (conforme descrito na seção 6.1), o seu uso como substituto viável para a gasolina somente será possível se a sua produção for economicamente competitiva. Desse modo, há um grande interesse na otimização de todas as etapas do processo de produção do etanol. Uma das opções para melhorar a produtividade deste processo é através da remoção contínua de parte do etanol assim que ele é formado (SILVA, 1998).

Como o processo fermentativo é inibido pelo etanol, a extração seletiva deste produto durante a fermentação é altamente desejável pois evita o fenômeno da inibição por produto (frequente nos processos convencionais) e, conseqüentemente, promove o aumento de produtividade. Vários esquemas combinando a fermentação com um processo de separação têm sido desenvolvidos e, dentre as alternativas propostas para atingir este objetivo, destacam-se a fermentação sob vácuo (RAMALINGHAN e FINN, 1977; CYSEWSKI e WILKE, 1977); extração por solvente (MINIER e GOMA, 1982); fermentação combinada com tanque *flash* operando sob vácuo (MAIORELLA *et al.*, 1984); ultra-filtração (SOURIRAJAN, 1987); pervaporação (CHRISTEN *et al.*, 1990); fermentação combinada com tanque *flash* operando sob pressão atmosférica (ISHIDA e SHIMIZU, 1996) e *stripping* com gás CO_2 (TAYLOR *et al.*, 1998). SILVA *et al.* (1999) demonstraram que um esquema que combina um fermentador com um tanque *flash* operando sob vácuo apresenta várias vantagens e melhor desempenho do que os processos fermentativos convencionais, além de se ajustar bem às plantas industriais existentes no Brasil. Segundo COSTA *et al.* (2001), o processo extrativo de fermentação alcoólica proposto por SILVA *et al.* (1999) apresenta alta produtividade ($21 - 22 \text{ kg}/(m^3/h)$) quando comparado aos processos contínuos convencionais ($12 \text{ kg}/(m^3/h)$).

A operação do processo fermentativo em modo contínuo é bastante interessante do ponto de vista econômico, pois propicia o aumento de produtividade e melhora o seu rendimento. Contudo, a implementação de processos contínuos requer um estudo prévio do comportamento do processo. A influência da temperatura sobre os parâmetros cinéticos, por exemplo, deve ser levada em conta pois existe uma certa dificuldade para manter a temperatura de fermentação constante durante os processos industriais de grande porte (COSTA *et al.*, 2001). ATALA *et al.* (2001) relatam que pequenos desvios na

temperatura da dorna de fermentação ($2-4^{\circ}\text{C}$) podem deslocar o processo das condições ótimas de operação. Outro aspecto importante a ser considerado na otimização dos processos fermentativos é o desenvolvimento de uma estratégia de controle eficiente pois, através de sua implementação, pode-se manter o processo nas condições ótimas de operação e com isso minimizar os custos de produção (COSTA *et al.*, 2002a; COSTA *et al.*, 2002b).

O caso de estudo apresentado nesta seção é um processo extrativo de fermentação alcoólica contínua baseado no esquema proposto por SILVA *et al.* (1999). COSTA *et al.* (2001) estudaram o projeto, a otimização e a determinação de uma estrutura de controle (DMC-SISO) adequada para este processo através de simulações. Contudo, os parâmetros cinéticos do modelo matemático foram determinados experimentalmente e consideraram explicitamente a sua dependência com a temperatura (ATALA *et al.*, 2001). Posteriormente, COSTA *et al.* (2002b) e MELEIRO *et al.* (2002c) implementaram um controlador preditivo não linear (SISO e MIMO, respectivamente) no mesmo processo.

O objetivo deste capítulo é aplicar as metodologias de identificação e controle descritas nos capítulos 4 e 5 no processo de fermentação extrativa. Assim como no capítulo 6, o processo estudado neste capítulo é uma planta contínua para produção de etanol, contudo, além das diferenças de projeto e de equipamentos, este caso de estudo é tipicamente multi-variável (apresentando três variáveis manipuladas e três variáveis controladas), constituindo um desafio ideal para testar a versatilidade e potencialidade das estratégias de identificação e controle não lineares propostas nesta tese.

7.2 Descrição do Processo

O esquema geral do processo de fermentação alcoólica extrativa (proposto por SILVA *et al.* (1999)) está representado na figura 7.1. A planta é formada por quatro unidades interligadas: *i)* O Fermentador, que é a unidade onde o etanol é produzido; *ii)* A Centrífuga, unidade responsável pela separação das fases sólida (microorganismos) e líquida (mosto e etanol); *iii)* A Cuba, unidade de tratamento dos microorganismos, e *iv)* O Tanque *Flash*, unidade de separação da mistura etanol-mosto (fase aquosa). Esta planta foi construída com o objetivo de emular as condições industriais convencionais descritas por ANDRIETTA e MAUGERI FILHO (1994) (e utilizado como caso de estudo no capítulo 6) . A diferença é a presença de um tanque *flash* para extrair continuamente parte do etanol formado e que somente um fermentador é utilizado, ao invés de um sistema de fermentadores em cascata.

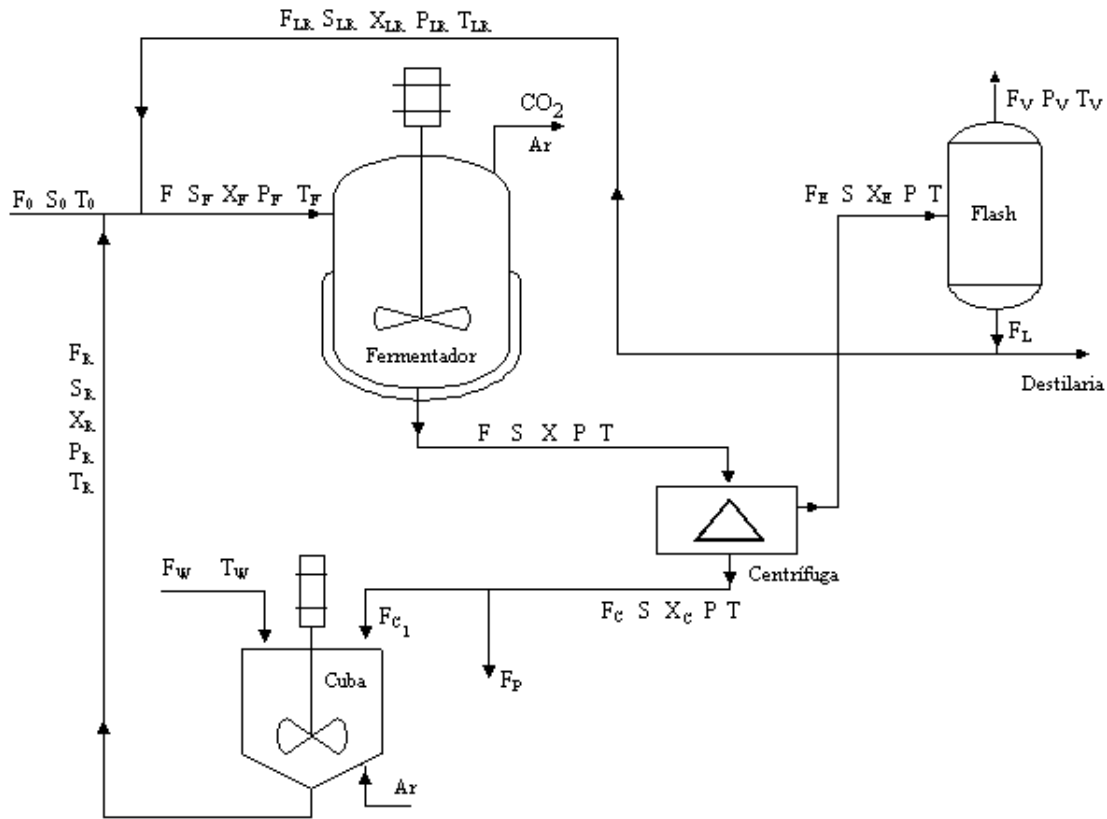


Figura 7.1: Representação esquemática da planta de fermentação alcoólica extrativa.

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

O processo inicia-se com a alimentação de substrato no fermentador, que contém a concentração de microorganismos adequada para dar início ao processo fermentativo. A corrente de saída do fermentador, contendo substrato, células e etanol, é enviada para a centrífuga e separada em duas fases: a *fase pesada*, contendo a maior parte das células (microorganismos); e a *fase leve*, praticamente isenta de células. A fase pesada, após sofrer uma purga para eliminar parte das células, é enviada para a unidade de tratamento de microorganismos (cuba). Na cuba, o creme de leveduras é diluído com água e acidulado com ácido sulfúrico, sob forte agitação, até que o pH esteja em torno de 2,0 – 2,5. Este tratamento ácido tem como objetivo diminuir a contaminação do processo por bactérias (geralmente sensíveis a valores baixos de pH). A suspensão de células tratadas é enviada para o fermentador para um novo ciclo do processo.

Enquanto o estado estacionário não é atingido, a operação do processo extrativo é similar ao processo contínuo convencional: a fase leve que deixa a centrífuga é enviada para a unidade de destilação (que não está representada na figura 7.1) para a separação da mistura etanol-água. Quando o estado estacionário é atingido, o sistema de separação à 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. O *flash* é operado em uma faixa de temperatura entre 28 – 30°C, escolhida com o objetivo de eliminar a necessidade de instalação de um trocador de calor, o que reduz drasticamente os custos¹ fixos e de manutenção do processo fermentativo. A faixa de pressão associada é de 30 – 40 $mmHg$. O esquema proposto mostrou-se adequado para manter as condições ideais para o crescimento da levedura (*Sacharomyces cerevisiae*), i.e., temperatura constante durante o processo fermentativo (SILVA *et al.*, 1999).

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. De acordo com o conhecimento prático oriundo das unidades industriais, a concentração de etanol dentro do fermentador deve estar em torno de 40 kg/m^3 pois, nesta faixa, o etanol apresenta bai-

¹Segundo ANDRIETTA (1994), os trocadores de calor e as centrífugas são os equipamentos que mais oneram a produção do etanol.

xo efeito inibitório para as leveduras que produzem o álcool mas é altamente inibitório para a maioria dos microorganismos contaminantes (SILVA *et al.*, 1999).

7.2.1 Modelagem Matemática

A modelagem matemática do processo considerado neste capítulo consiste em balanços de massa e energia representados por equações algébricas e diferenciais. Todos os equipamentos, com exceção do fermentador, são modelados assumindo a condição de estado pseudo-estacionário. Esta aproximação é adequada porque a dinâmica do fermentador é muito lenta quando comparada com as dinâmicas das demais unidades.

Os parâmetros cinéticos utilizados na modelagem do fermentador foram determinados experimentalmente, levando-se em conta a sua dependência com a temperatura. Os experimentos foram realizados por ATALA *et al.* (2001) em uma faixa de temperatura ($28 - 40^{\circ}\text{C}$) que cobre uma ampla região operacional. Além disso, COSTA *et al.* (2001) utilizaram um modelo intrínseco (MONBOUQUETTE, 1987; MONBOUQUETTE, 1992), que leva em consideração a fração volumétrica ocupada pela biomassa e as concentrações de etanol intra e extra-celular. A modelagem do processo proposta por COSTA *et al.* (2001) considerou ainda a queda de viabilidade celular ao longo da fermentação, de modo que a concentração de biomassa (microorganismos) foi dividida em duas partes: uma ativa (células viáveis) e outra inativa (células mortas).

Para se determinarem as concentrações e as vazões de entrada para o fermentador, fez-se um balanço de massa global no processo assumindo as seguintes aproximações (ANDRIETTA, 1994):

- 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, X_R , é fixa. Para que X_R seja mantida fixa, ajusta-se a vazão de água, F_W , que dilui o creme de leveduras.
- A vazão de reciclo de células, cujo valor é determinado pela razão de reciclo de microorganismos, $R_{microrg}$, é mantida constante através da variação da vazão de purga, F_p . A purga serve para retirar produtos secundários acumulados no fermentador e também permite uma constante renovação celular.

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

Utilizando as aproximações descritas acima e considerando a hipótese de estado pseudo-estacionário para todas as unidades, com exceção do fermentador, obtém-se o modelo matemático do processo, representado pelas equações descritas a seguir.

A vazão de alimentação do fermentador, F , é descrita da seguinte forma:

$$F = F_0 + F_{LR} + F_R \quad (7.1)$$

A vazão de reciclo de células é dada por:

$$F_R = F R_{microrg} \quad (7.2)$$

A equação 7.1 pode ser reescrita substituindo o valor de F_R dado pela equação 7.2:

$$F = \frac{F_0 + F_{LR}}{(1 - R_{microrg})} \quad (7.3)$$

O tempo de residência do fermentador é dado por:

$$\tau = \frac{V}{F} \quad (7.4)$$

onde V é o volume do fermentador.

O balanço de biomassa (microorganismos) para a centrífuga resulta em

$$FX = F_E X_E + F_c X_c \quad (7.5)$$

onde X representa a concentração de microorganismos e os índices $_E$ e $_c$ estão associados às correntes que seguem para o *flash* e para a cuba, respectivamente.

O balanço global em torno da centrífuga permite determinar a vazão que segue para a cuba:

$$F_c = F - F_E \quad (7.6)$$

que combinada com a equação 7.5, fornece a vazão de alimentação do tanque *flash*:

$$F_E = \frac{F(X - X_c)}{X_E - X_c} \quad (7.7)$$

7.2 Descrição do Processo

A vazão de alimentação da cuba é obtida fazendo-se o balanço de biomassa para esta unidade:

$$F_{c_1} = \frac{F_R X_R}{X_c} \quad (7.8)$$

A vazão de água de diluição que alimenta a cuba é dada por:

$$F_W = F_R - F_{c_1} \quad (7.9)$$

A concentração de substrato na corrente de reciclo é determinada através de um balanço de massa para o substrato na cuba:

$$S_R = \frac{F_{c_1} S}{F_R} \quad (7.10)$$

e, de forma análoga, determina-se a concentração de etanol na mesma corrente:

$$P_R = \frac{F_{c_1} P}{F_R} \quad (7.11)$$

Os balanços de massa no tanque *flash* fornecem as seguintes quantidades:

$$F_E = F_V + F_L \quad (7.12)$$

$$F_E X_{E_i} = F_V y_i + F_L x_i \quad (7.13)$$

e o equilíbrio líquido-vapor da mistura etanol-água é calculado com a seguinte expressão:

$$k_{eq_i} = \frac{y_i}{x_i} = \gamma_i \frac{p_i^{sat}}{p} \quad (7.14)$$

onde o valor de p_i^{sat} é dado pela equação de Antoine e o valor de γ_i é calculado com o modelo NRTL² (SILVA, 1998).

As concentrações de substrato, etanol e microorganismos presentes na corrente que retorna para o fermentador, vindas do tanque *flash*, são calculadas da seguinte forma:

$$S_{LR} = \frac{F_E S}{F_L} \quad (7.15)$$

²*Non-Randon Two-Liquid* (RENON e PRAUSNITZ, 1968).

$$P_{LR} = \frac{F_E P - F_V P_V}{F_L} \quad (7.16)$$

$$X_{LR} = \frac{F_E X_E}{F_L} \quad (7.17)$$

A vazão de reciclo do *flash* (que determina a vazão de reagentes e produto que retornam para o fermentador) é dada por:

$$F_{LR} = R_{flash} F_L \quad (7.18)$$

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

As concentrações de substrato, etanol e microorganismos na corrente de alimentação do fermentador são dadas por:

$$S_F = \frac{F_0 S_0 + F_{LR} S_{LR} + F_R S_R}{F} \quad (7.19)$$

$$P_F = \frac{F_R P_R + F_{LR} P_{LR}}{F} \quad (7.20)$$

$$X_F = \frac{F_R X_R + F_{LR} X_{LR}}{F} \quad (7.21)$$

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:

$$T_R = \frac{F_W T_W + F_{c1} T}{F_R} \quad (7.22)$$

$$T_F = \frac{F_0 T_0 + F_R T_R + F_{LR} T_{LR}}{F} \quad (7.23)$$

7.2 Descrição do Processo

Assumindo que o volume do reator é constante, as equações de balanço de massa e energia para o fermentador, utilizando o modelo intrínseco, são escritas da seguinte forma:

$$\frac{dX_v}{dt} = r_x - r_d - \frac{F}{V}(X_v - X_{vF}) \quad (7.24)$$

$$\frac{dX_d}{dt} = r_d - \frac{F}{V}(X_d - X_{dF}) \quad (7.25)$$

$$\frac{d \left[\left(1 - \frac{X_t}{\rho} \right) SV \right]}{dt} = F(S_F - S) - r_s V \quad (7.26)$$

$$\frac{d \left[\left(1 - \frac{X_t}{\rho} \right) PV + \frac{X_t}{\rho} \Gamma PV \right]}{dt} = r_p V + F(P_F - P) \quad (7.27)$$

$$\frac{dT}{dt} = \frac{F}{V}(T_F - T) + \frac{\Delta H}{\rho_m Cp} r_s \quad (7.28)$$

onde X_v , X_d , X_t , S e P são as concentrações de células viáveis, células mortas, células totais, substrato e produto, respectivamente. A temperatura no fermentador é representada por T ; as variáveis X_{vF} , X_{dF} , S_F , P_F e T_F são as concentrações de células viáveis, células mortas, substrato, produto e temperatura na alimentação, respectivamente. O calor de reação é representado por ΔH , ρ_m é a densidade do meio fermentativo, Cp é o calor específico, F é a vazão de alimentação, V é o volume do fermentador, ρ é a relação da massa seca pelo volume de massa úmida e Γ é a relação entre as concentrações de etanol intra e extra-celular. Os valores das constantes físicas da equação de balanço de energia (7.28) são exibidos na tabela 7.1 ((SILVA *et al.*, 1999)).

Tabela 7.1: Constantes físicas da equação de energia.

ΔH	$2,167.10^3 \text{ J/kg de ART}$
ρ_m	1000 kg/m^3
Cp	$4,183.10^3 \text{ J/kgK}$

As equações para as taxas cinéticas de crescimento, morte, consumo de substrato e formação de produto (r_x , r_d , r_s e r_p , respectivamente), são dadas pelas equações (7.29) a (7.32) e os parâmetros cinéticos presentes nestas equações foram determinados experimentalmente por ATALA *et al.* (2001) (tabela 7.2).

$$r_x = \mu_{max} \frac{S}{K_s + S} \exp(-K_i S) \left(1 - \frac{X_t}{X_{max}}\right)^m \left(1 - \frac{P}{P_{max}}\right)^n X_v \quad (7.29)$$

$$r_d = K_{dT} X_v \exp(K_{dP} P) \quad (7.30)$$

$$r_s = \frac{r_x}{Y_x} + m_x X_v \quad (7.31)$$

$$r_p = Y_{px} r_x + m_p X_v \quad (7.32)$$

7.2 Descrição do Processo

Tabela 7.2: Valores dos 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}	$-0,4421 T^2 + 26,41 T - 279,75$
Y_X	$2,704 \exp(-0,1225 T)$
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

7.3 Estratégias de Identificação para o Processo Extrativo de Fermentação Alcoólica

Uma vez que o estudo de caso apresentado neste capítulo é multi-variável (três entradas manipuladas e três saídas controladas), decidiu-se que todos os modelos utilizados para identificar o processo serão do tipo MIMO. O processo extrativo de fermentação alcoólica também apresenta variáveis que podem ser consideradas como perturbações medidas, porém, diferente da estratégia adotada no capítulo 6, estas variáveis não serão incorporadas explicitamente nos modelos identificados. Esta estratégia foi utilizada com o intuito de se obterem modelos com menos entradas, o que diminui consideravelmente a dimensão do mapeamento a ser aproximado, minimizando o problema³ relacionado com a explosão de parâmetros e dados necessários para identificar o sistema (maldição da dimensionalidade, discutida na seção 4.3.3) . Além disso, o uso destes modelos nas estratégias de controle propostas na seção 7.4, servirão para verificar o desempenho dos controladores frente a perturbações não modeladas.

Ao longo desta seção serão discutidos todos os passos necessários para identificar o processo extrativo de fermentação alcoólica utilizando os modelos neurais MLP e UGL. Um modelo linear do tipo ARX também será empregado na identificação deste processo para auxiliar na avaliação da não linearidade do processo, além de servir como parâmetro de comparação para o desempenho dos modelos não lineares.

7.3.1 Determinação das Variáveis de Entrada, Saída e Perturbação

SILVA *et al.* (1999) utilizaram os métodos do planejamento fatorial e análise da superfície de resposta (BOX *et al.*, 1978; BARROS NETO *et al.*, 2001) gerada a partir de dados de simulação para determinar as variáveis de processo mais relevantes. Neste estudo foram determinadas as condições ótimas de operação, assim como a influência das variáveis de entrada sobre as saídas de interesse deste processo, seu rendimento e produtividade. A análise dos resultados obtidos provou que, operando o processo

³Associado, principalmente, com modelos com estrutura não hierárquica.

7.3 Estratégias de Identificação para o Processo Extrativo de Fermentação Alcoólica

extrativo dentro da faixa determinada, é possível obter índices de produtividade três vezes maior do que os fornecidos pelos processos fermentativos contínuos tradicionais.

Com o objetivo de determinar a melhor estrutura de controle para este processo, além de complementar o estudo realizado por SILVA *et al.* (1999), COSTA *et al.* (2001) estudaram o seu comportamento dinâmico em malha aberta. Tomando como base as variáveis de processo determinadas previamente (SILVA, 1998; SILVA *et al.*, 1999), COSTA *et al.* (2001) utilizaram o planejamento fatorial e a análise da superfície de resposta 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 (variáveis manipuladas e possíveis perturbações). De acordo com estes estudos, as principais variáveis de interesse para este processo são:

7.3.1.1 Variáveis de Entrada

- Vazão da corrente de alimentação: F_0 [m^3/h];
- Razão de reciclo de microorganismos: $R_{microorg}$ [*adimensional*];
- Razão de reciclo do tanque *flash*: R_{flash} [*adimensional*].

7.3.1.2 Variáveis de Saída

- Concentração de etanol no fermentador: P [kg/m^3];
- Concentração de substrato no fermentador: S [kg/m^3];
- Concentração de microorganismos no fermentador: $X = X_t = X_v + X_d$ [kg/m^3].

7.3.1.3 Variáveis Perturbação

- Concentração de substrato na corrente de alimentação: S_0 [kg/m^3];
- Temperatura da corrente de alimentação: T_0 [$^{\circ}C$].

7.3.2 Geração dos Dados para Identificação do Processo

O simulador do processo extrativo de fermentação alcoólica foi utilizado para gerar um conjunto de dados contendo os sinais de entrada e saída referente a 2000 horas de operação do processo. Conforme destacado no início desta seção, o conjunto de dados foi produzido variando-se apenas as variáveis manipuladas. As variáveis perturbação, S_0 e T_0 , foram mantidas constantes em seus estados estacionários de referência ($180 \text{ kg}/\text{m}^3$ e $30 \text{ }^\circ\text{C}$, respectivamente).

Diferente da estratégia adotada no *Estudo de Caso 1* (capítulo 6), os sinais de entrada (variáveis manipuladas) utilizados para gerar os dados para o treinamento e validação deste processo foram projetados de acordo com a metodologia descrita no item **II** da seção 4.3.4 (página 84). Neste caso, a amplitude e a frequência dos sinais de entrada são variadas aleatoriamente, gerando sinais de entrada semelhantes àqueles ilustrados na figura 4.3 (página 85). Assim, cada uma das entradas foi manipulada de acordo com uma sequência de degraus com amplitude variando dentro do seu respectivo intervalo operacional: $F_0 = [60, 150] \text{ m}^3/\text{h}$; $R_{flash} = [0.2, 0.7]$ e $R_{microrg} = [0.2, 0.7]$.

7.3.3 Determinação do Período de Amostragem

O período de amostragem escolhido para este processo foi determinado em função do tempo médio de análise de um cromatógrafo (equipamento utilizado para medir as concentrações de ART e etanol), já que a medida da concentração celular (que é feita com um turbidímetro) pode ser obtida em intervalos de tempo muito pequenos. Segundo ANDRIETTA (1994), um tempo de amostragem de $T = 12 \text{ min}$ é adequada para este processo.

O procedimento de amostragem das 2000 horas de simulação utilizando este tempo de amostragem, forneceu um conjunto de dados com 10000 padrões. Este conjunto foi dividido em duas partes: A primeira, contendo 9000 padrões, foi utilizada na etapa de treinamento dos modelos (MLP, UGL e DMC) e a segunda parte, contendo os restantes 1000 padrões, foi utilizada na etapa de validação dos modelos.

7.3 Estratégias de Identificação para o Processo Extrativo de Fermentação Alcoólica

7.3.4 Determinação da Estrutura dos Modelos

A determinação da estrutura adequada para os modelos propostos neste capítulo foi baseada em um estudo preliminar - realizado por COSTA *et al.* (2001) - sobre o comportamento dinâmico do processo extrativo de fermentação alcoólica através da determinação do grau de interação entre suas variáveis de entrada e de saída. Desse modo, foi possível estabelecer quais são as variáveis de entrada que mais influenciam o comportamento dinâmico das variáveis de saídas.

A seleção dos regressores para os modelos MLP e UGL (seções 7.3.6 e 7.3.7, respectivamente), por sua vez, teve como base os estudos sobre a modelagem neural do mesmo processo apresentados posteriormente por COSTA *et al.* (2002a), COSTA *et al.* (2002b), e MELEIRO *et al.* (2002c). Nestes trabalhos, os autores utilizaram redes neurais do tipo *Functional Link Network* (FLN) para identificar o processo. Em todos os casos, foram utilizadas representações MISO de modelos neurais FLN para descrever o comportamento dinâmico das variáveis controladas de interesse (P e S). Cada modelo neural MISO teve suas variáveis de entrada e saída determinadas a partir do estudo realizado por COSTA *et al.* (2001) e os regressores determinados através de procedimentos de tentativa e erro, avaliando o desempenho do modelo obtido em previsões a um passo e de longo alcance.

Diferente da estratégia adotada naqueles estudos, optou-se nesta tese pela construção de modelos neurais (MLP e UGL) com representação MIMO, fazendo a devida adaptação na estrutura dos regressores. Assim, o vetor de regressão para cada variável é dado por: $[F_0(k-1)]$, $[R_{flash}(k-1)]$, $[R_{microrg}(k-1)]$, $[S(k-1)]$, $[P(k-1)]$ e $[X(k-1)]$. A estrutura geral dos modelos neurais utilizados neste capítulo é ilustrada na figura 7.2.

7.3.5 Estimação e Validação dos Modelos

O arquivo de dados entrada-saída correspondente a 2000 horas de simulação do processo foi separado em duas partes; a primeira com os dados correspondentes às 1800 primeiras horas foi utilizada no treinamento e a parte referente às 200 horas finais

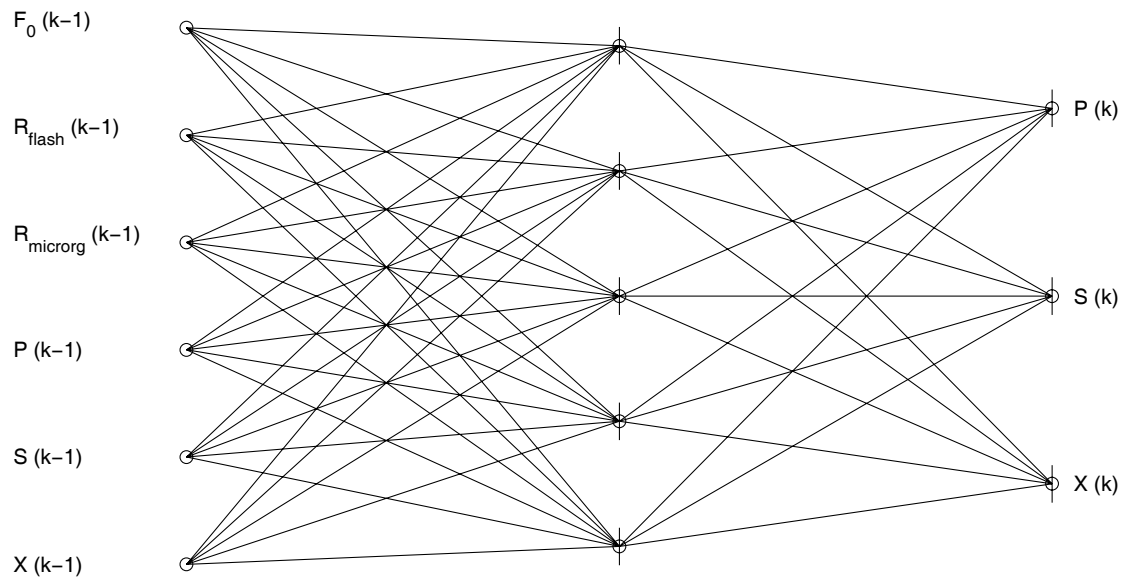


Figura 7.2: Estrutura dos modelos neurais do tipo NNARX.

foi destinada à validação dos modelos. O critério de parada foi o mesmo descrito na seção 6.3.5.

7.3 Estratégias de Identificação para o Processo Extrativo de Fermentação Alcoólica

7.3.6 Identificação Utilizando Modelos Neurais MLP

Apresenta-se nesta seção o resultado da modelagem do processo extrativo de fermentação alcoólica fornecido por um modelo não linear representado por uma rede neural *feedforward* do tipo Perceptron Multicamadas (MLP).

A rede neural selecionada para modelar este processo apresenta uma estrutura geral semelhante à estrutura mostrada na figura 7.2, com $n_e = 6$ entradas e $n_s = 3$ saídas. O modelo neural MLP utilizado nesta seção foi treinado com um algoritmo de otimização que utiliza o método do *Gradiente Conjugado Escalonado Modificado* - GCEM⁴ (DE CASTRO e VON ZUBEN, 1998a), que permite o cálculo exato da informação de 2ª ordem com custo computacional substancialmente reduzido.

De forma análoga ao procedimento empregado na seção 6.3.7, vários modelos neurais com diferentes quantidades de neurônios escondidos, n_{le} , foram testados para verificar que modelo forneceria o melhor resultado. A avaliação do desempenho dos modelos neurais foi feita através da comparação dos erros finais de aproximação EQM (Erro Quadrático Médio) e EQMN (Erro Quadrático Médio Normalizado, dado pela equação 6.27), e também através da análise visual das simulações para previsão um passo adiante e recursiva. As tabelas 7.3 e 7.4 mostram os EQM e EQMN para vários valores de n_{le} .

Tabela 7.3: EQM de modelagem para os dados de validação - Modelos MLP.

	Previsão 1 Passo				Simulação Recursiva		
n_{le}	S	P	X		S	P	X
3	0.0271	0.0155	0.0140		0.2072	0.0920	0.0936
5	0.0205	0.0140	0.0073		0.0691	0.0640	0.0397
10	0.0183	0.0109	0.0070		0.0394	0.0349	0.0325
15	0.0174	0.0101	0.0075		0.0308	0.0321	0.0347
20	0.0174	0.0103	0.0080		0.0373	0.0274	0.0307

Conforme já destacado na seção 6.3.7, os resultados que devem ser avaliados com mais atenção são aqueles relacionados com a simulação recursiva, já que estes

⁴Mesmo algoritmo utilizado na seção 6.3.7.

**ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO
ALCOÓLICA CONTÍNUA**

Tabela 7.4: EQMN de modelagem para os dados de validação - Modelos MLP.

	Previsão 1 Passo				Simulação Recursiva		
n_{l_e}	S	P	X		S	P	X
3	1.4303	0.0568	0.0666		20.6617	2.3829	4.4520
5	0.5895	0.0586	0.0182		3.1202	0.8498	0.6226
10	0.3451	0.0579	0.0067		1.8788	1.6443	0.5012
15	0.0918	0.0428	0.0074		0.8573	0.5359	0.4314
20	0.1338	0.0490	0.0063		0.2599	0.5270	0.2548

modelos serão utilizados no projeto de controladores preditivos. A análise dos dados fornecidos pelas tabelas mostra uma clara (e esperada) tendência de redução dos erros EQM e EQMN na medida em que mais neurônios são acrescentados na camada intermediária, n_{l_e} . Contudo, a partir de $n_{l_e} = 10$ neurônios, o ganho na qualidade da aproximação não justifica a adição de neurônios extras. O mesmo comportamento é verificado quando se faz a análise visual dos resultados da simulação dinâmica fornecida pelos modelos neurais. Desse modo, a rede neural escolhida foi a que utilizou $n_{l_e} = 10$ neurônios escondidos. O número total de parâmetros a serem estimados para este modelo neural é dado por $p = (n_e + 1) n_{l_e} + (n_{l_e} + 1) n_{l_s} = 103$, onde n_{l_e} e n_{l_s} representam o número de neurônios das camadas de entrada e saída, respectivamente. As figuras 7.3, 7.4 e 7.5 apresentam o desempenho do modelo neural para $n_{l_e} = 10$.

7.3 Estratégias de Identificação para o Processo Extrativo de Fermentação Alcoólica

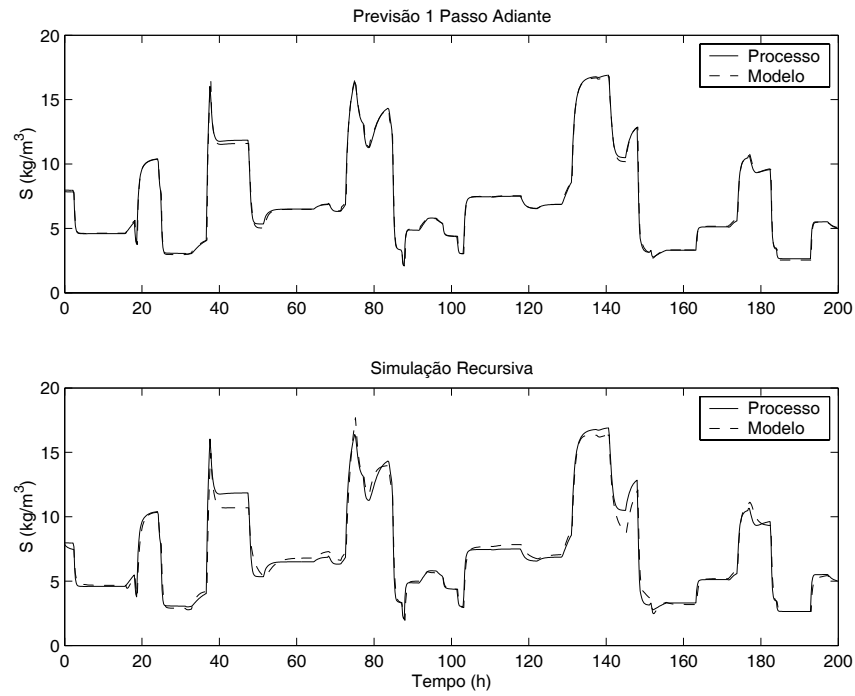


Figura 7.3: Desempenho do modelo MLP com $n_{l_e} = 10$ para a variável S .

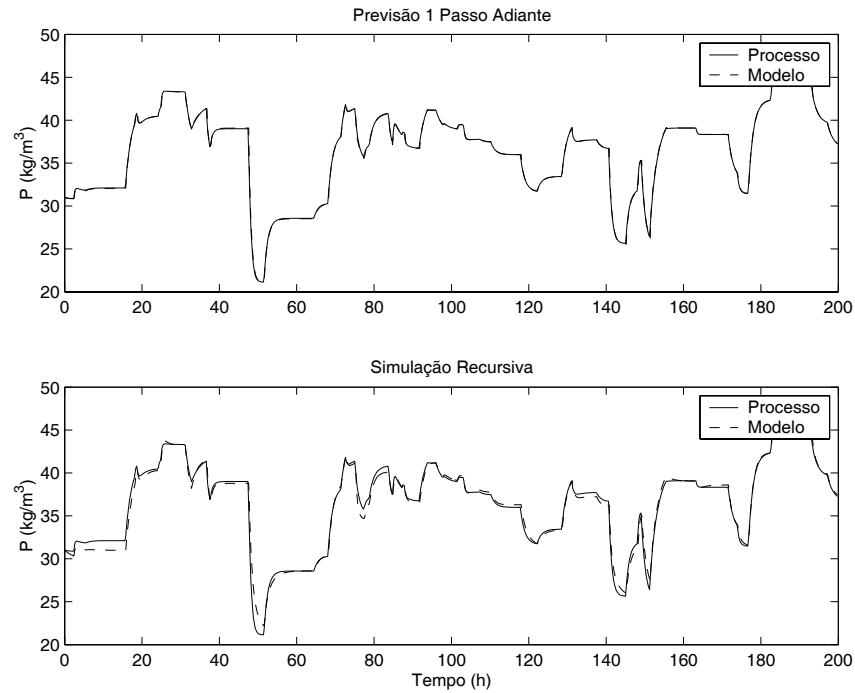


Figura 7.4: Desempenho do modelo MLP com $n_{l_e} = 10$ para a variável P .

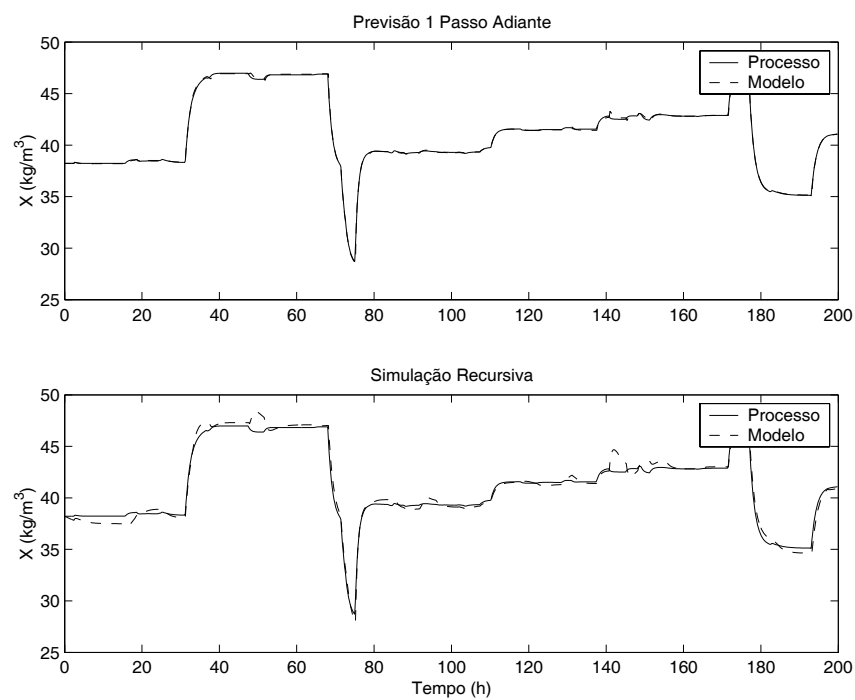


Figura 7.5: Desempenho do modelo MLP com $n_{l_e} = 10$ para a variável X .

7.3 Estratégias de Identificação para o Processo Extrativo de Fermentação Alcoólica

7.3.7 Identificação Utilizando o Modelo Neural Construtivo UGL

Conforme mencionado na seção 6.3.8, a aplicação dos modelos neurais convencionais na modelagem de sistemas não lineares apresenta como principal inconveniente a dificuldade na determinação do número adequado de neurônios da camada interna, n_{le} . Pelo fato de se tratar de um método de tentativa e erro, este é um procedimento que, via de regra, demanda bastante tempo. Além disso, existe ainda a reconhecida sensibilidade aos parâmetros de inicialização apresentada pela maioria dos algoritmos de treinamento desenvolvidos para estes tipos de modelos.

O modelo UGL utilizado para identificar este processo apresenta a mesma estrutura selecionada para o modelo MLP (figura 7.2), ou seja, uma rede neural com estrutura do tipo NNARX (seção 4.4.2) com $n_e = 6$ entradas e $n_s = 3$ saídas. O modelo neural UGL foi treinado com o algoritmo de aprendizado construtivo (VON ZUBEN, 1996) utilizando funções de ativação paramétricas na forma de expansão truncada. Este algoritmo, que utiliza funções de Hermite (de ordem máxima $P = 6$) como funções básicas, selecionou automaticamente $n_{le} = 5$ neurônios escondidos. Apresenta-se a seguir o resultado da modelagem do processo extrativo de fermentação alcoólica fornecido pelo modelo neural construtivo UGL.

As tabelas 7.5 e 7.6 apresentam os erros de modelagem apresentados pelo modelo neural UGL na predição do comportamento dinâmico das variáveis S , P e X em simulações um passo adiante e recursiva. A análise dos resultados apresentados pelos modelos neurais UGL ($n_{le} = 5$) e MLP ($n_{le} = 10$), pode ser feita comparando-se os erros EQM e EQMN apresentados nas respectivas tabelas. O resultado revela que o modelo neural UGL apresentou, de modo geral, resultados equivalentes aos apresentados pelo modelo neural MLP, com a vantagem de necessitar de menos parâmetros além da determinação do número de neurônios ser feita automaticamente.

Tabela 7.5: EQM de modelagem para os dados de validação - Modelo UGL.

Variável de Saída	EQM Prev. 1 Passo	EQM Sim. Recursiva
S	0.0239	0.0899
P	0.0143	0.0645
X	0.0086	0.0326

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

Tabela 7.6: EQMN de modelagem para os dados de validação - Modelo UGL.

Variável de Saída	EQMN Prev. 1 Passo	EQMN Sim. Recursiva
S	0.5668	3.8763
P	0.0931	1.2406
X	0.0395	2.5813

A avaliação do comportamento dinâmico do modelo neural UGL feita através da análise visual das simulações um passo adiante e recursiva (figuras 7.6, 7.7 e 7.8), comprova a sua eficiência e seu grande potencial para aplicações na modelagem de processos não lineares multi-variáveis.

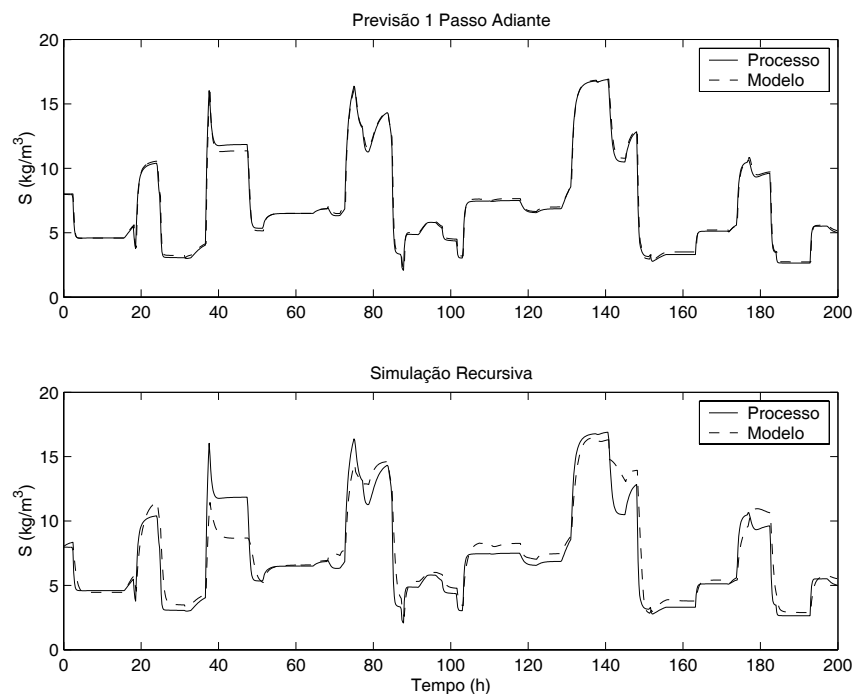


Figura 7.6: Desempenho do modelo UGL com $n_{l_e} = 5$ para a variável S .

As figuras 7.9 a 7.13 exibem as funções de ativação de cada neurônio para as três saídas do processo (S , P e X). Estas figuras ilustram a habilidade do algoritmo de aprendizado construtivo de selecionar diferentes funções de ativação para cada neurônio de modo a garantir a melhor aproximação possível para um dado mapeamento entrada-saída.

7.3 Estratégias de Identificação para o Processo Extrativo de Fermentação Alcoólica

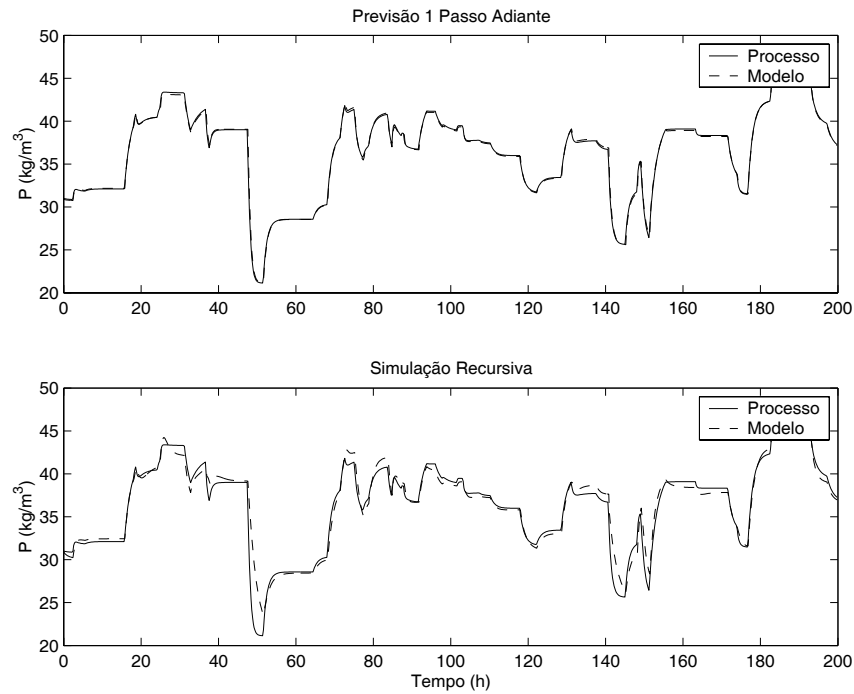


Figura 7.7: Desempenho do modelo UGL com $n_{l_e} = 5$ para a variável P .

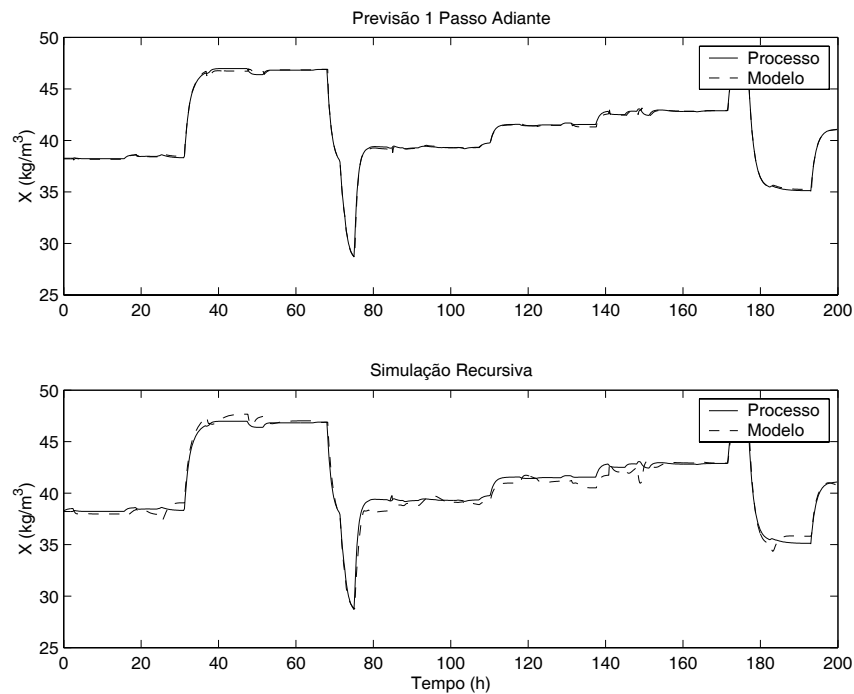


Figura 7.8: Desempenho do modelo UGL com $n_{l_e} = 5$ para a variável X .

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

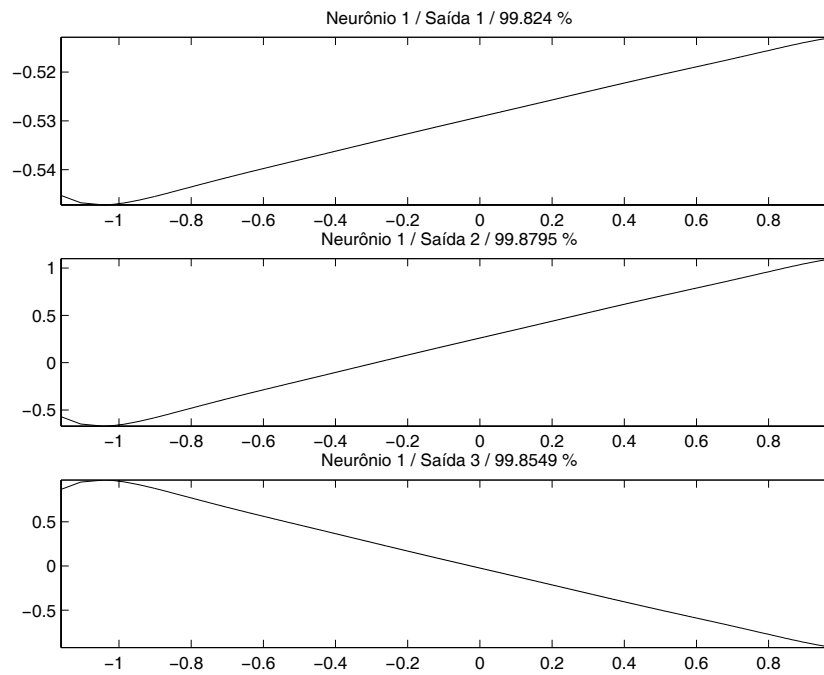


Figura 7.9: Funções de ativação do neurônio 1.

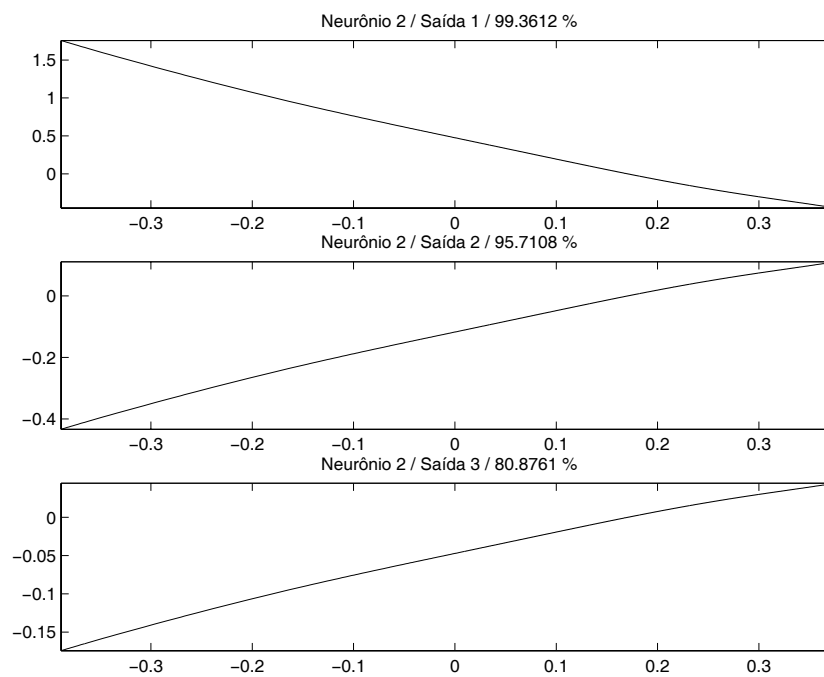


Figura 7.10: Funções de ativação do neurônio 2.

7.3 Estratégias de Identificação para o Processo Extrativo de Fermentação Alcoólica

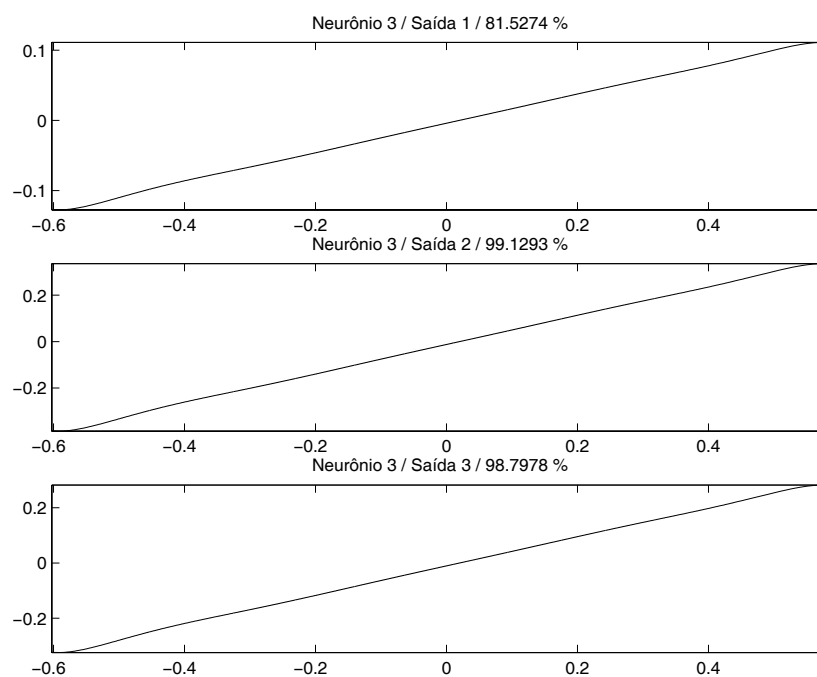


Figura 7.11: Funções de ativação do neurônio 3.

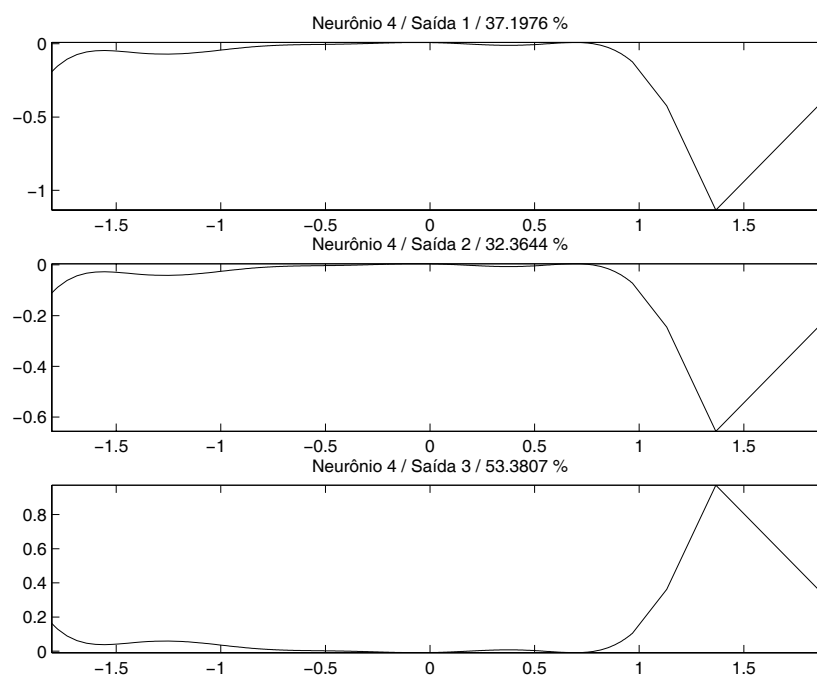


Figura 7.12: Funções de ativação do neurônio 4.

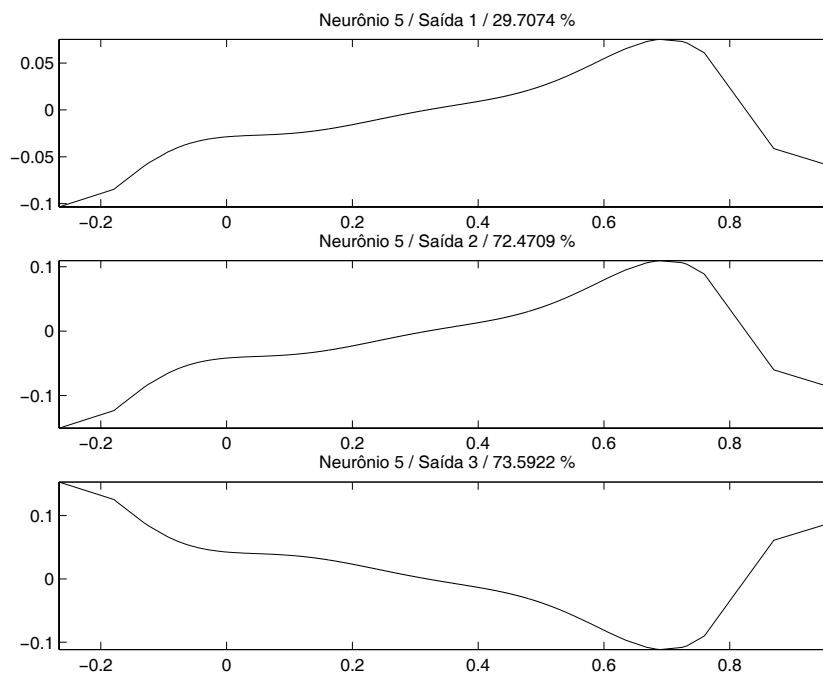


Figura 7.13: Funções de ativação do neurônio 5.

Uma das grandes vantagens da determinação automática do número ótimo de neurônios proporcionado pelo algoritmo de aprendizado construtivo, é a obtenção de modelos neurais mais parcimoniosos e, conseqüentemente, com capacidade de generalização potencialmente maior. No caso específico do modelo neural UGL identificado para o processo extrativo de fermentação alcoólica, o número de parâmetros do modelo a serem estimados é igual a $p = (n_e + 1) n_{l_e} + (n_{l_e} + 1) n_{l_s} = 53$.

7.3 Estratégias de Identificação para o Processo Extrativo de Fermentação Alcoólica

7.3.8 Identificação Utilizando o Modelo Linear ARX

A exemplo do procedimento adotado no *Estudo de Caso 1* (seção 6.3.9), decidiu-se modelar o processo extrativo de fermentação alcoólica utilizando um modelo linear ARX MIMO, cuja realização pode ser representada pela seguinte equação:

$$\begin{aligned} \mathbf{y}(k) = & -A_1\mathbf{y}(k-1) - \dots - A_{n_y}\mathbf{y}(k-n_y) \\ & + B_1\mathbf{u}(k-1) + \dots + B_{n_u}\mathbf{u}(k-n_u) + \boldsymbol{\nu}(k) \end{aligned} \quad (7.33)$$

onde $A_i \in \mathbb{R}^{n_s \times n_s}$, $B_i \in \mathbb{R}^{n_s \times n_e}$, n_u é o máximo atraso entre os regressores de entrada e n_y é o máximo atraso entre os regressores de saída. Os vetores de saída, entrada e ruído branco, \mathbf{y} , \mathbf{u} e $\boldsymbol{\nu}$, respectivamente, são definidos da seguinte forma:

$$\begin{aligned} \mathbf{y}(k) &= [y_1(k) \ y_2(k) \ \dots \ y_{n_s}(k)]^T \\ \mathbf{u}(k) &= [u_1(k) \ u_2(k) \ \dots \ u_{n_e}(k)]^T \\ \boldsymbol{\nu}(k) &= [\nu_1(k) \ \nu_2(k) \ \dots \ \nu_{n_s}(k)]^T \end{aligned} \quad (7.34)$$

Pode-se definir as seguintes matrizes de polinômios (seção 3.3.3):

$$\begin{aligned} \mathbf{A}(q^{-1}) &= I + A_1q^{-1} + \dots + A_{n_y}q^{-n_y}; \\ \mathbf{B}(q^{-1}) &= B_1q^{-1} + \dots + B_{n_u}q^{-n_u}; \end{aligned}$$

resultando na seguinte expressão:

$$\mathbf{A}(q^{-1})\mathbf{y}(k) = \mathbf{B}(q^{-1})\mathbf{u}(k) + \boldsymbol{\nu}(k) \quad (7.35)$$

que pode ser reescrita da seguinte forma:

$$A_0\mathbf{y}(k) = -A_1\mathbf{y}(k-1) + B_1\mathbf{u}(k-1) \quad (7.36)$$

onde

$$A_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{y}(k) = \begin{bmatrix} S(k) \\ P(k) \\ X(k) \end{bmatrix}$$

$$A_1 = \begin{bmatrix} -0.9811 & 0.0423 & 0.0099 \\ 0.0376 & -0.9013 & -0.0220 \\ -0.0062 & -0.0087 & -0.8608 \end{bmatrix} \quad \mathbf{y}(k-1) = \begin{bmatrix} S(k-1) \\ P(k-1) \\ X(k-1) \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0.0095 & -0.0219 & -0.0293 \\ 0.0044 & -0.0998 & -0.0639 \\ -0.0010 & 0.1471 & 0.0027 \end{bmatrix} \quad \mathbf{u}(k-1) = \begin{bmatrix} F_0(k-1) \\ R_{flash}(k-1) \\ R_{microrg}(k-1) \end{bmatrix}$$

Apresenta-se a seguir os resultados da simulação do processo com o modelo linear ARX MIMO utilizando os dados de validação. Pode-se verificar através das figuras 7.14, 7.15 e 7.16 que, assim como observado no *Estudo de Caso 1*, o modelo linear não apresentou um desempenho satisfatório para as simulações recursivas. A explicação para este comportamento foi discutida no final da seção 6.3.9 e também se aplica aqui.

7.3 Estratégias de Identificação para o Processo Extrativo de Fermentação Alcoólica

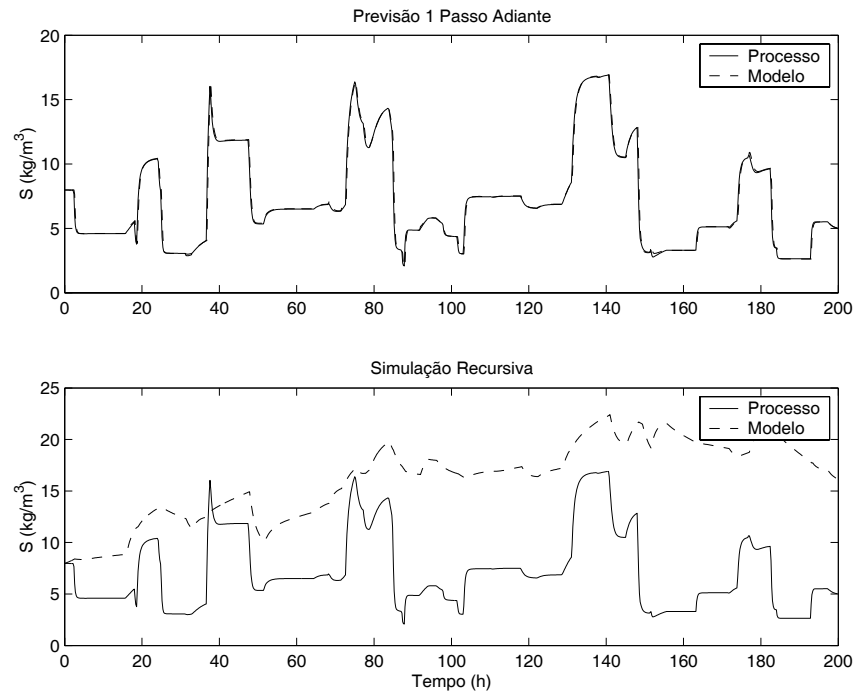


Figura 7.14: Desempenho do modelo linear ARX para a variável S .

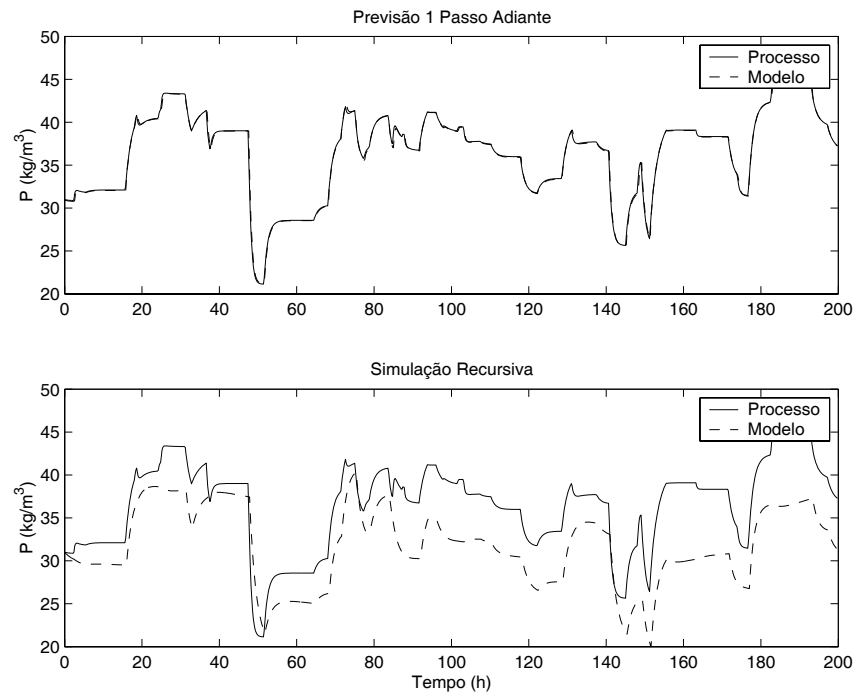


Figura 7.15: Desempenho do modelo linear ARX para a variável P .

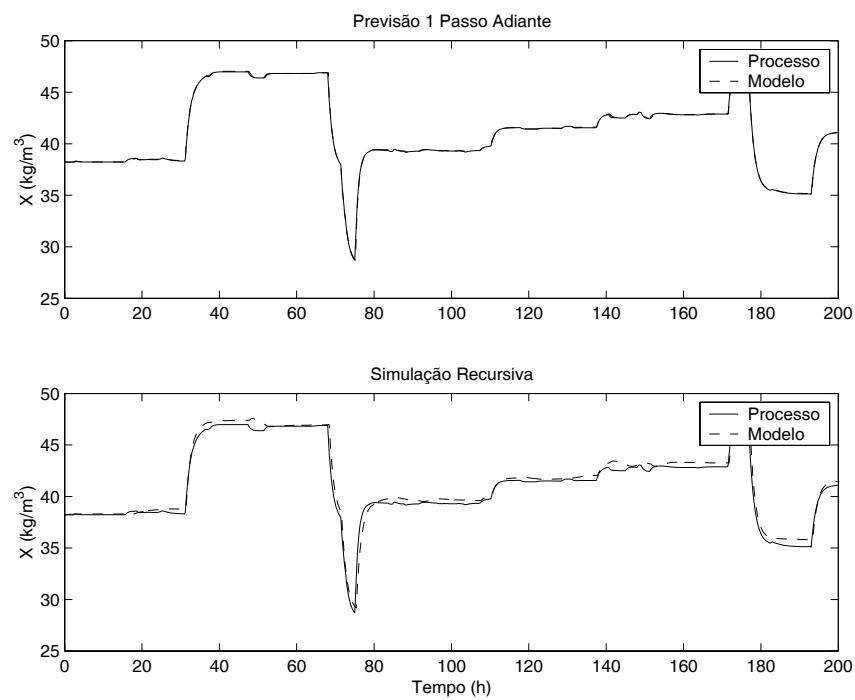


Figura 7.16: Desempenho do modelo linear ARX para a variável X .

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

Nesta seção são apresentados o projeto de controladores preditivos não lineares (baseados nos modelos identificados nas seções 7.3.6, 7.3.7 e 7.3.8) e os resultados referentes à sua aplicação no controle do processo extrativo de fermentação alcoólica. Os controladores apresentados a seguir são equivalentes àqueles apresentados nas seções 6.4.3, 6.4.4 e 6.4.5, porém, o processo estudado neste capítulo é multi-variável, possuindo três variáveis controladas, três variáveis manipuladas e duas variáveis perturbação (medidas). Devido a estas características, o *Estudo de Caso 2* apresenta-se como um bom teste para a robustez e a versatilidade deste tipo de controlador.

O primeiro estudo de controle para este processo foi proposto por SILVA (1998), que utilizou um controlador DMC SISO para controlar a concentração de substrato no fermentador, S , manipulando a vazão de entrada da corrente de alimentação, F_0 .

Posteriormente, COSTA *et al.* (2001) utilizaram os conceitos de planejamento fatorial e análise da superfície de resposta, associados à simulação dinâmica, para investigar as interações entre as variáveis manipuláveis e controladas do processo. Com isso, foi possível determinar a melhor estrutura de controle para o processo, ou seja, identificar que variável manipulada seria a mais indicada para controlar uma determinada variável de processo (S , P ou X). Três controladores do tipo DMC SISO foram testados por COSTA *et al.* (2001) para verificar a melhor estrutura para controlar (separadamente) as concentrações de substrato, S , e etanol, P , no fermentador.

Em seguida, COSTA *et al.* (2002a; 2002b) projetaram um controlador preditivo não linear (do tipo MPC) mono-variável baseado em um modelo neural do tipo *Functional Link Network* (FLN) e o utilizaram para controlar S manipulando F_0 .

Dando seqüência ao estudo de controladores preditivos não lineares, MELEIRO *et al.* (2002c) estenderam o projeto do controlador MPC-FLN para considerar o caso multi-variável, com duas variáveis controladas (S e P) e duas variáveis manipuladas (F_0 e R_{flash}). Neste trabalho, o modelo não linear do processo também foi identificado a partir de uma rede neural do tipo FLN, contudo, foram utilizados dois modelos neurais MISO, um para cada variável controlada.

Os controladores utilizados nesta tese consideram o caso mais completo de controle multi-variável para o processo extrativo de fermentação alcoólica. Propõe-se controlar S , P e X , através da manipulação de F_0 , R_{flash} e $R_{microrg}$. Consideram-se os problemas servo e regulatório para todas as variáveis controladas, além de perturbações nas variáveis S_0 e T_0 . Todos os modelos do processo (neurais e linear) utilizados no projeto dos controladores são MIMO.

7.4.1 Formulação Geral do Controlador Preditivo Não Linear Multi-Variável

Considere um processo multi-variável com V_c variáveis controladas e V_m variáveis manipuladas. O objetivo de um controlador preditivo não linear do tipo MPC, é calcular um conjunto de ações de controle futuras que minimize a seguinte função objetivo:

$$J = \sum_{j=1}^{V_c} \sum_{i=N_1}^{N_y} (\hat{y}_{j,k+i}^c - w_{j,k+i})^2 + \sum_{j=1}^{V_m} \lambda_j \sum_{i=1}^{N_u} (\Delta u_{j,k+i-1})^2 \quad (7.37)$$

onde λ é o fator de ponderação (parâmetros de ajuste que penalizam as ações de controle), $\Delta \mathbf{u}$ são os incrementos nas variáveis manipuladas, N_1 é o horizonte inicial, N_y é o horizonte de predição, N_u é o horizonte de controle, \mathbf{w} são as trajetórias de referência (*set-points*) e $\hat{\mathbf{y}}^c$ são as previsões dos modelos (MLP ou UGL) corrigidas. Estas previsões são obtidas recursivamente até um total de N_y previsões futuras para cada variável controlada, V_c , e corrigidas da seguinte forma:

$$\hat{\mathbf{y}}_{k+i}^c = \hat{\mathbf{y}}_{k+i} + \mathbf{d}_k, \quad (i = 1, \dots, N_y) \quad (7.38)$$

Na equação (7.38), $\hat{\mathbf{y}}_{k+i}$ são as previsões feitas pelo modelo para as saídas controladas do processo no instante de amostragem futuro, $k+i$, e \mathbf{d}_k são os termos de correção dados por:

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

$$\mathbf{d}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k \quad (7.39)$$

onde \mathbf{y}_k são as saídas medidas do processo no instante de amostragem presente e $\hat{\mathbf{y}}_k$ são as respectivas predições do modelo (calculadas no instante de amostragem anterior).

O algoritmo de otimização calcula os incrementos, $\Delta \mathbf{u}$, para as variáveis manipuladas de modo a minimizar a função objetivo (7.37). As ações de controle futuras são então derivadas a partir dos incrementos ótimos de controle:

$$\mathbf{u}_{k+i} = \mathbf{u}_{k+i-1} + \Delta \mathbf{u}_{k+i}, \quad (i = 0, \dots, N_u - 1) \quad (7.40)$$

É importante observar que somente as primeiras N_u ações de controle são otimizadas, enquanto que as demais são mantidas constantes, ou seja:

$$\mathbf{u}_{k+i} = \mathbf{u}_{k+N_u-1}, \quad (i = N_u, \dots, N_y - 1) \quad (7.41)$$

O problema de otimização também está sujeito às seguintes restrições:

$$\mathbf{y}_{min} \leq \hat{\mathbf{y}}_{k+i} \leq \mathbf{y}_{max}, \quad (i = 1, \dots, N_y) \quad (7.42)$$

$$\mathbf{u}_{min} \leq \mathbf{u}_{k+i-1} \leq \mathbf{u}_{max}, \quad (i = 1, \dots, N_u) \quad (7.43)$$

$$|\mathbf{u}_{k+i-1} - \mathbf{u}_{k+i-2}| \leq \Delta \mathbf{u}_{max}, \quad (i = 1, \dots, N_u) \quad (7.44)$$

Finalmente, a estratégia do “Horizonte Móvel⁵” é adotada (GARCIA *et al.*, 1989; SOETERBOEK, 1992; CAMACHO e BORDONS, 1999), onde somente a primeira ação de controle (para cada variável manipulada), \mathbf{u}_k , é implementada e o problema de otimização é resolvido novamente em cada instante de amostragem. O problema de

⁵*Receding Horizon.*

otimização (com restrições) é resolvido através da técnica da *Programação Quadrática Sucessiva* - SQP (EDGAR e HIMMELBLAU, 1988).

Para o caso específico da planta de fermentação extrativa, as restrições operacionais impostas às variáveis manipuladas, $\mathbf{u} = [F_0 \ R_{flash} \ R_{microrg}]$, e controladas $\mathbf{y} = [S \ P \ X]$ são dadas por:

- Restrições nas Variáveis Controladas:

$$\begin{aligned} S(k+i) &\geq 0, & (i = 1, \dots, N_y) \\ P(k+i) &\geq 0, & (i = 1, \dots, N_y) \\ X(k+i) &\geq 0, & (i = 1, \dots, N_y) \end{aligned} \quad (7.45)$$

- Restrições nas Variáveis Manipuladas:

$$\begin{aligned} 60 &\leq F_0(k+i) \leq 150, & (i = 0, \dots, N_u - 1) \\ 0.2 &\leq R_{flash}(k+i) \leq 0.7, & (i = 0, \dots, N_u - 1) \\ 0.2 &\leq R_{microrg}(k+i) \leq 0.7, & (i = 0, \dots, N_u - 1) \end{aligned} \quad (7.46)$$

- Restrições nas Ações de Controle:

$$\begin{aligned} |F_0(k+i) - F_0(k+i-1)| &\leq 50, & (i = 0, \dots, N_u - 1) \\ |R_{flash}(k+i) - R_{flash}(k+i-1)| &\leq 0.1, & (i = 0, \dots, N_u - 1) \\ |R_{microrg}(k+i) - R_{microrg}(k+i-1)| &\leq 0.1, & (i = 0, \dots, N_u - 1) \end{aligned} \quad (7.47)$$

Conforme discutido na seção 7.3, existem duas variáveis do tipo perturbação, S_0 e T_0 , que não foram consideradas explicitamente nos modelos identificados (MLP, UGL e ARX). Com o objetivo de verificar o desempenho dos algoritmos de controle propostos em situações onde estas variáveis assumem valores diferentes de seus valores nominais, todas as simulações da planta em malha fechada (seções 7.4.2, 7.4.3 e 7.4.4) foram submetidas às mesmas oscilações nos valores das variáveis perturbação. Os valores considerados para estas variáveis estão representados na figura 7.17.

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

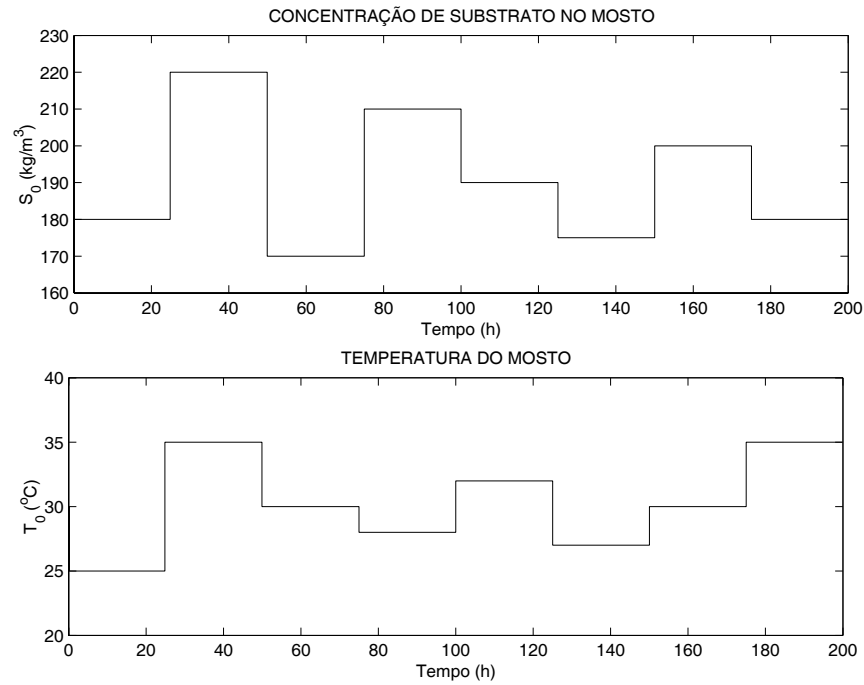


Figura 7.17: Perturbações agindo sobre o processo extrativo de fermentação alcoólica.

A seguir são apresentados os resultados do controle da planta de fermentação extrativa utilizando os controladores MPC-MLP (seção 7.4.2), MPC-UGL (seção 7.4.3) e DMC (seção 7.4.4). Todas as estratégias de controle são testadas, simultaneamente, em problemas servo e regulador. Em cada simulação em malha fechada, será imposta uma trajetória que deverá ser seguida por uma das variáveis controladas (problema servo), enquanto as outras duas deverão permanecer em seus estados estacionários de referência (problema regulador).

7.4.2 Controle Preditivo Não Linear Baseado no Modelo MLP

Apresenta-se nesta seção o desempenho do controlador preditivo não linear baseado no modelo neural MLP identificado na seção 7.3.6. Os parâmetros do controlador foram sintonizados com os seguintes valores: $N_1 = 1$, $N_y = 9$, $N_u = 2$ e $\lambda = [0,001 \ 0,1 \ 0,1]^T$.

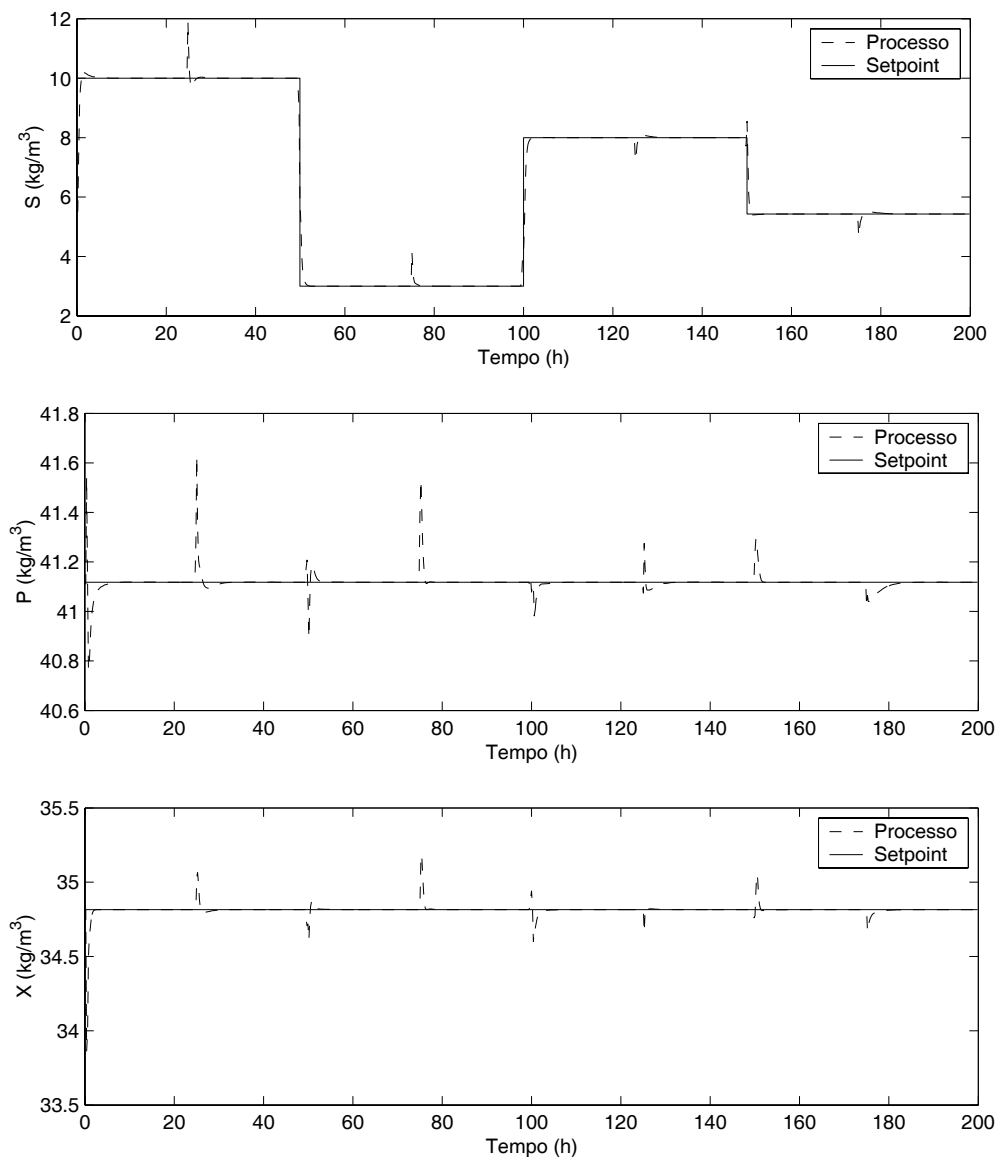


Figura 7.18: Controlador MPC-MLP - Problemas servo para a variável S e regulatório para as variáveis P e X .

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

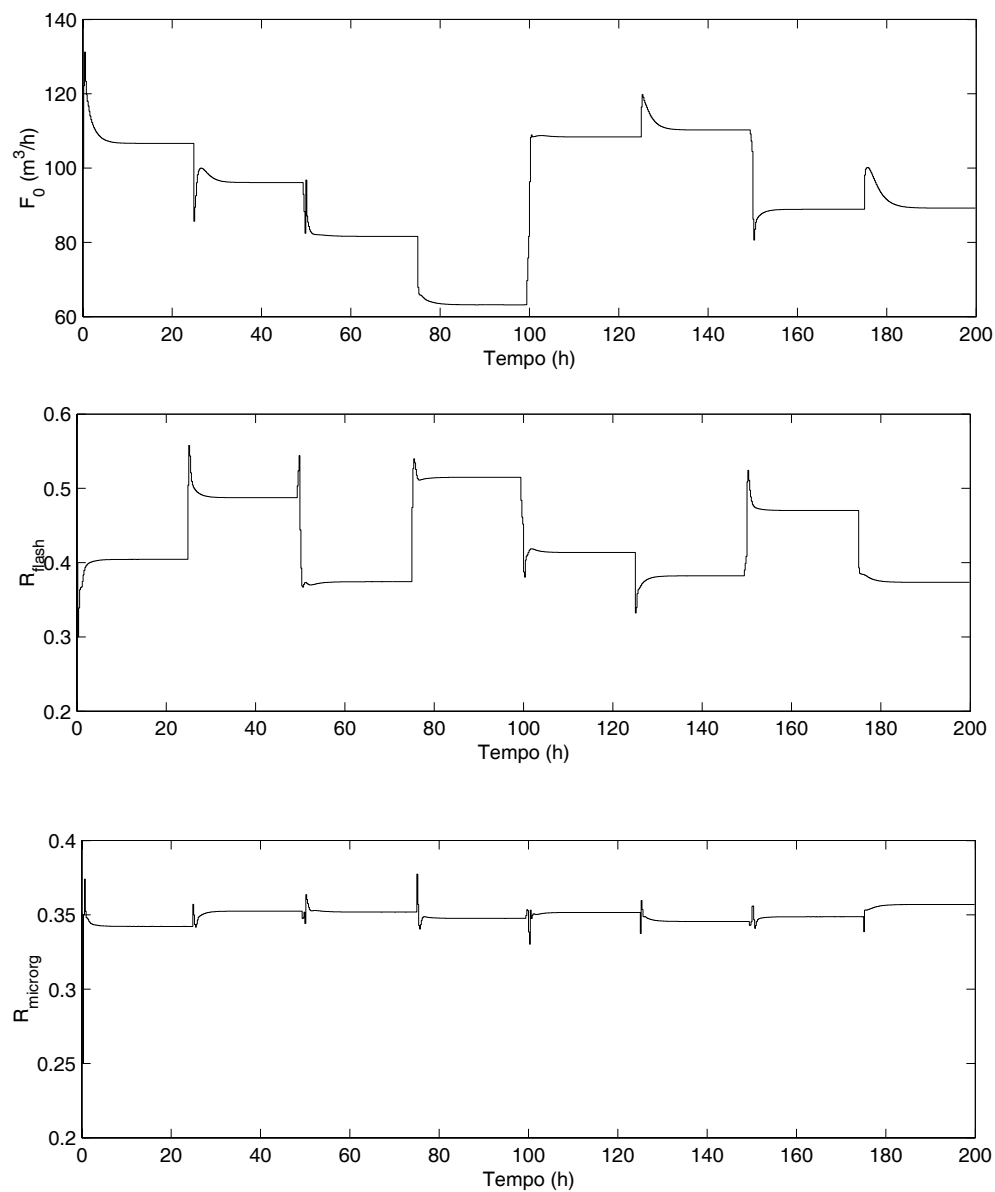


Figura 7.19: Controlador MPC-MLP - Variáveis manipuladas referentes à figura 7.18.

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

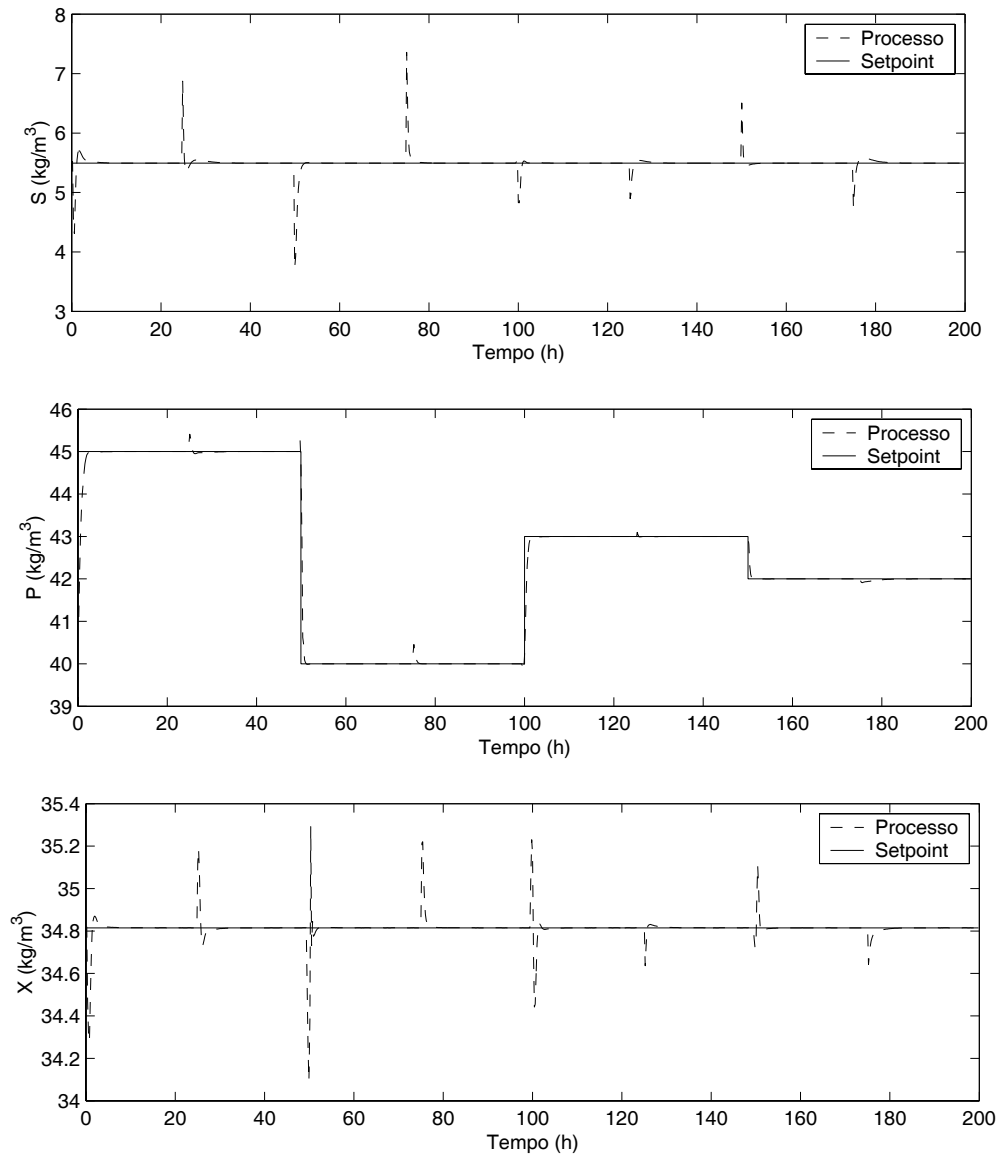


Figura 7.20: Controlador MPC-MLP - Problemas servo para a variável P e regulatório para as variáveis S e X .

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

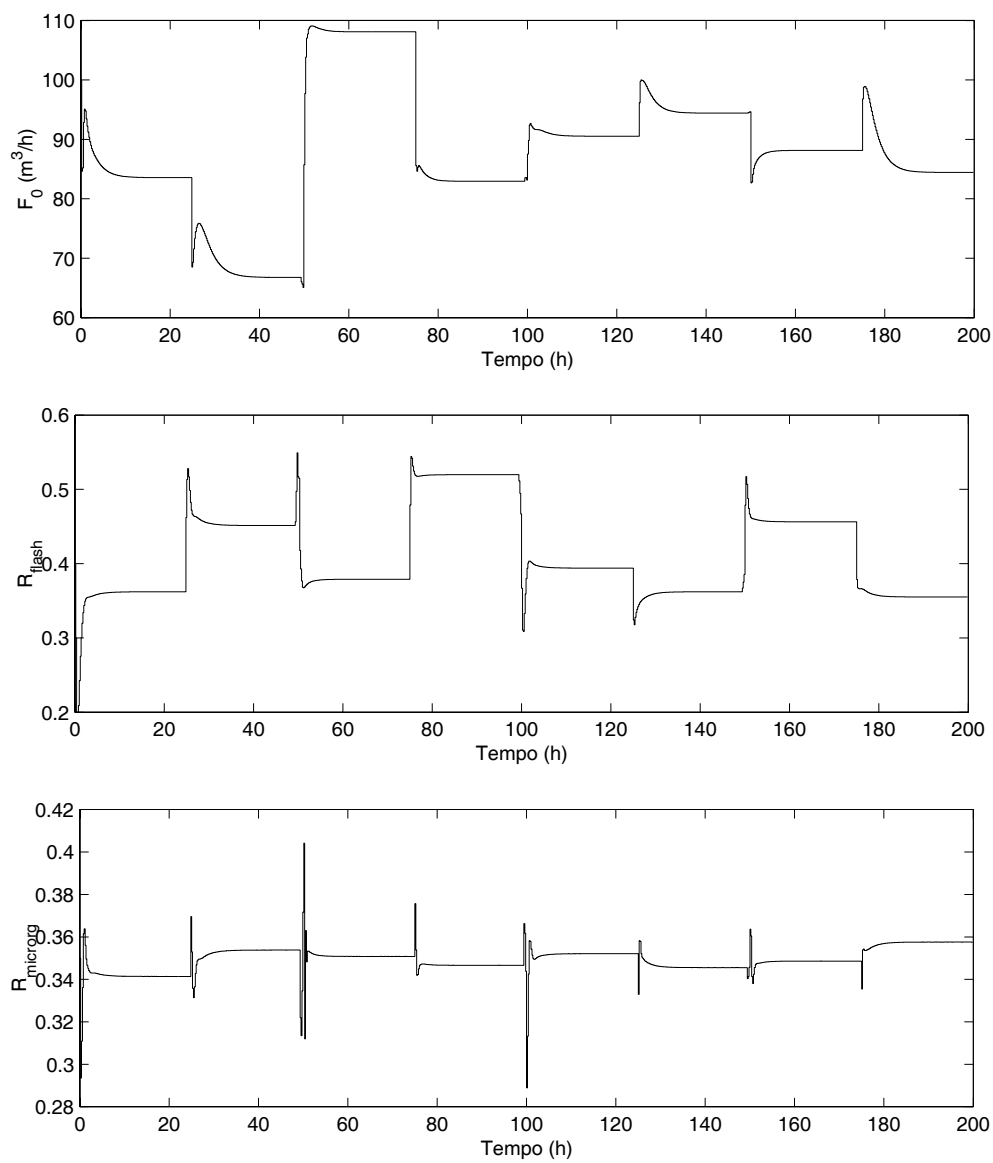


Figura 7.21: Controlador MPC-MLP - Variáveis manipuladas referentes à figura 7.20.

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

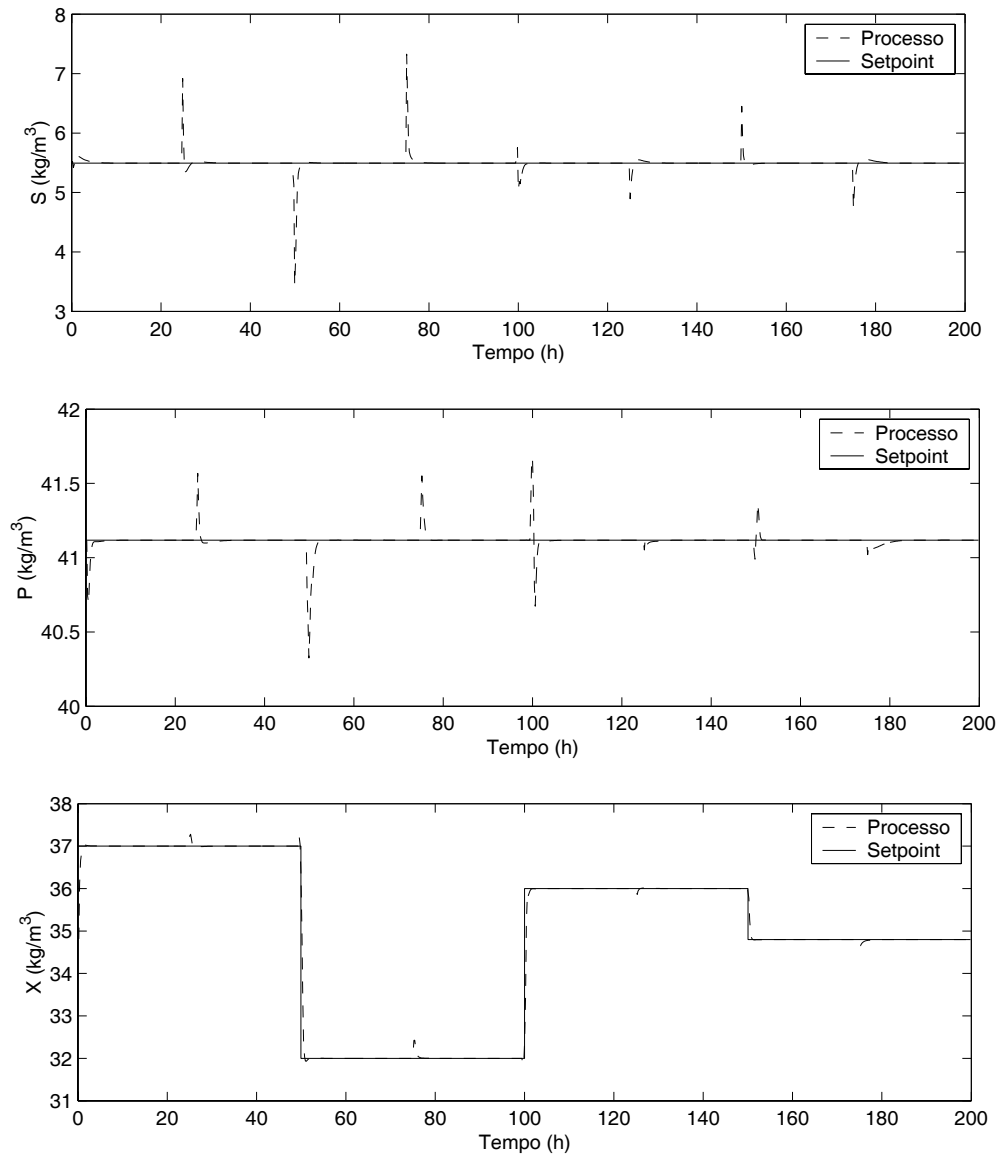


Figura 7.22: Controlador MPC-MLP - Problemas servo para a variável X e regulatório para as variáveis S e P .

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

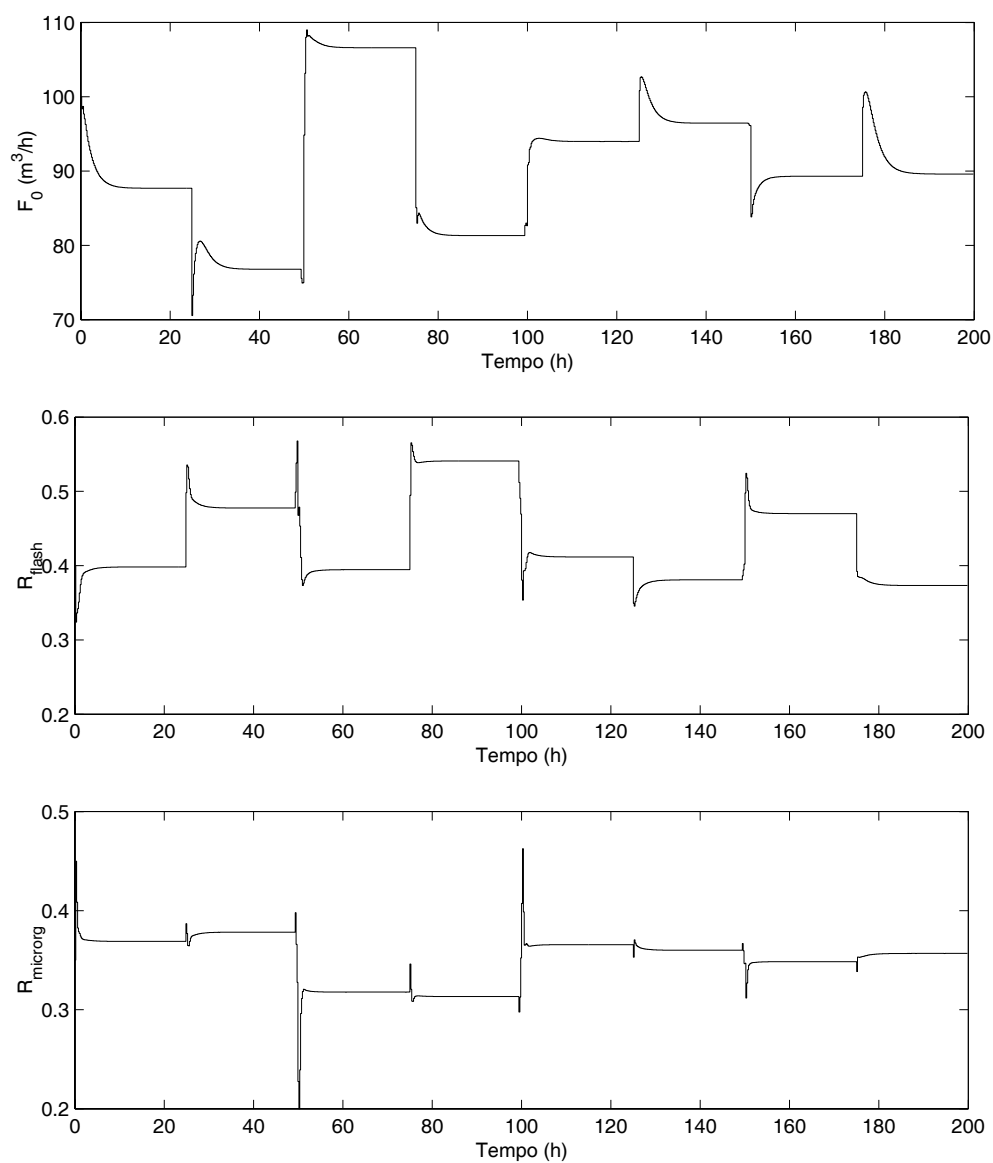


Figura 7.23: Controlador MPC-MLP - Variáveis manipuladas referentes à figura 7.22.

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

As figuras 7.18, 7.20 e 7.22, mostram o desempenho do controlador multi-variável MPC-MLP em simulações do processo em malha fechada. Os resultados obtidos demonstram que o controlador foi capaz de desempenhar, simultaneamente, as funções servo e regulatória de forma bastante eficiente. Deve-se destacar que em todas as simulações, as duas variáveis perturbação - cuja influência sobre o comportamento dinâmico do processo não foi considerada nos modelos utilizados pelos controladores - tiveram seus valores variados de acordo com os perfis exibidos na figura 7.17.

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

7.4.3 Controle Preditivo Não Linear Baseado no Modelo UGL

Nesta seção são apresentados os resultados quando o modelo UGL é utilizado no algoritmo do controlador preditivo multi-variável MPC-UGL. Os parâmetros para este controlador foram ajustados da seguinte forma: $N_1 = 1$, $N_y = 9$, $N_u = 2$ e $\lambda = [0,001 \ 0,2 \ 0,5]^T$.

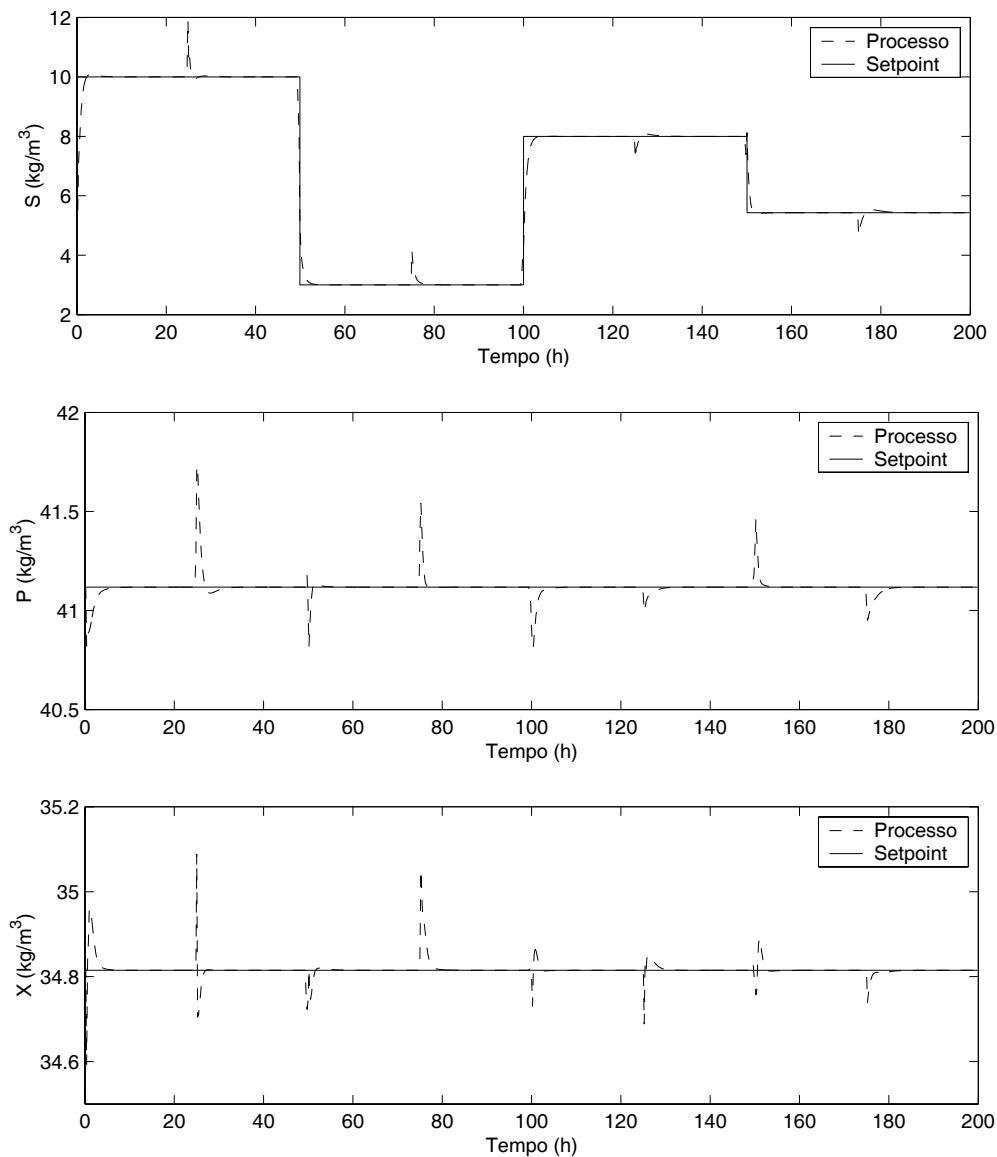


Figura 7.24: Controlador MPC-UGL - Problemas servo para a variável S e regulatório para as variáveis P e X .

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

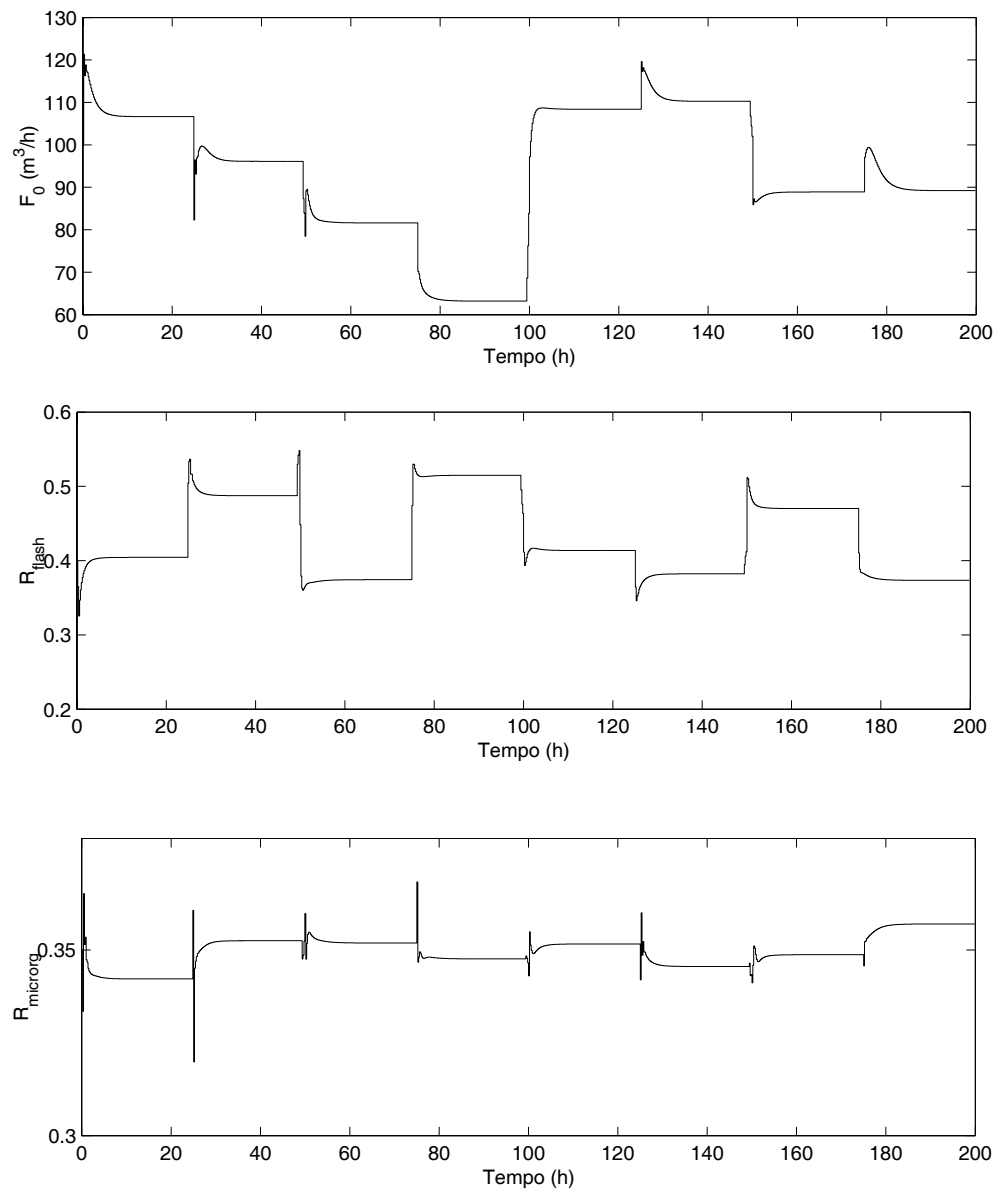


Figura 7.25: Controlador MPC-UGL - Variáveis manipuladas referentes à figura 7.24.

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

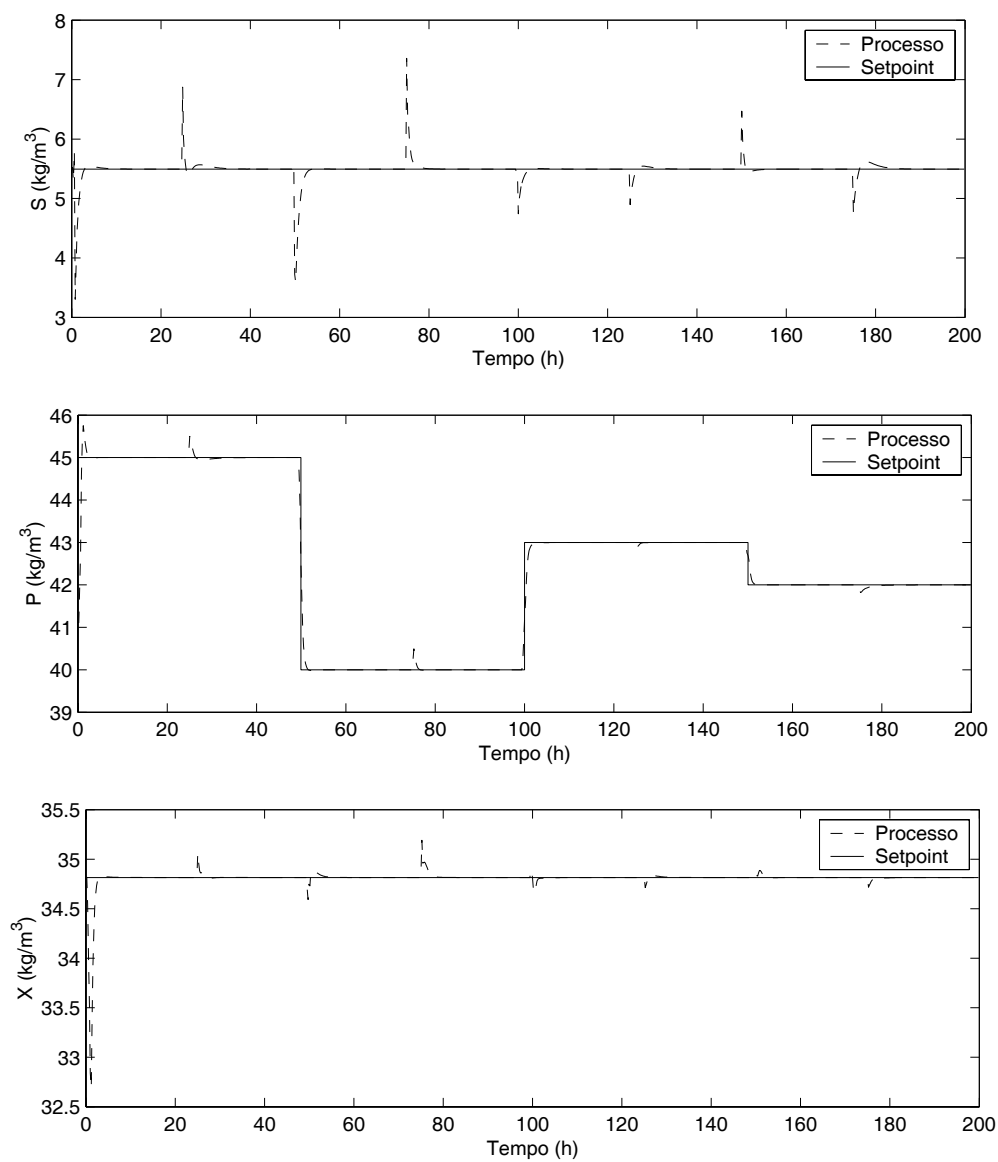


Figura 7.26: Controlador MPC-UGL - Problemas servo para a variável P e regulatório para as variáveis S e X .

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

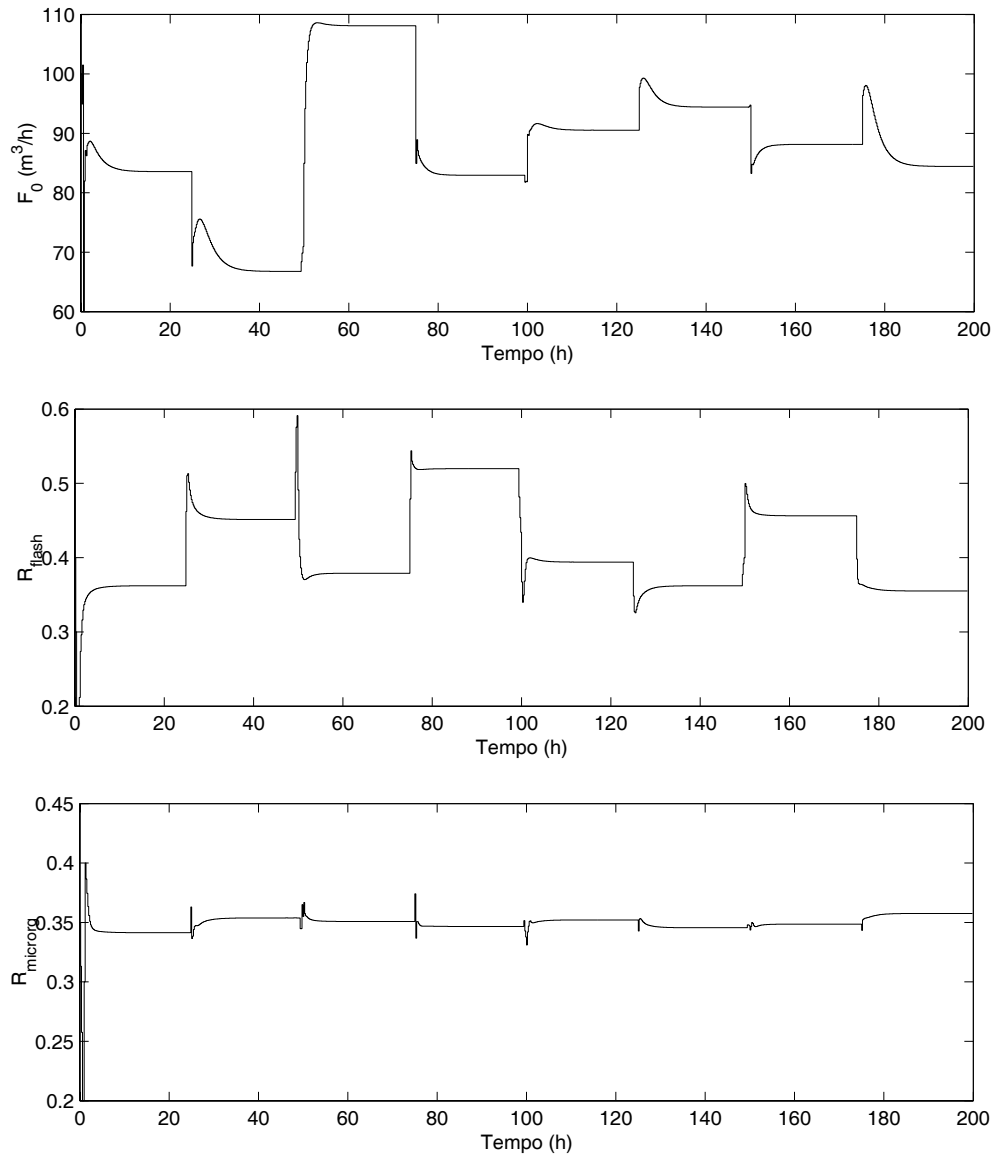


Figura 7.27: Controlador MPC-UGL - Variáveis manipuladas referentes à figura 7.26.

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

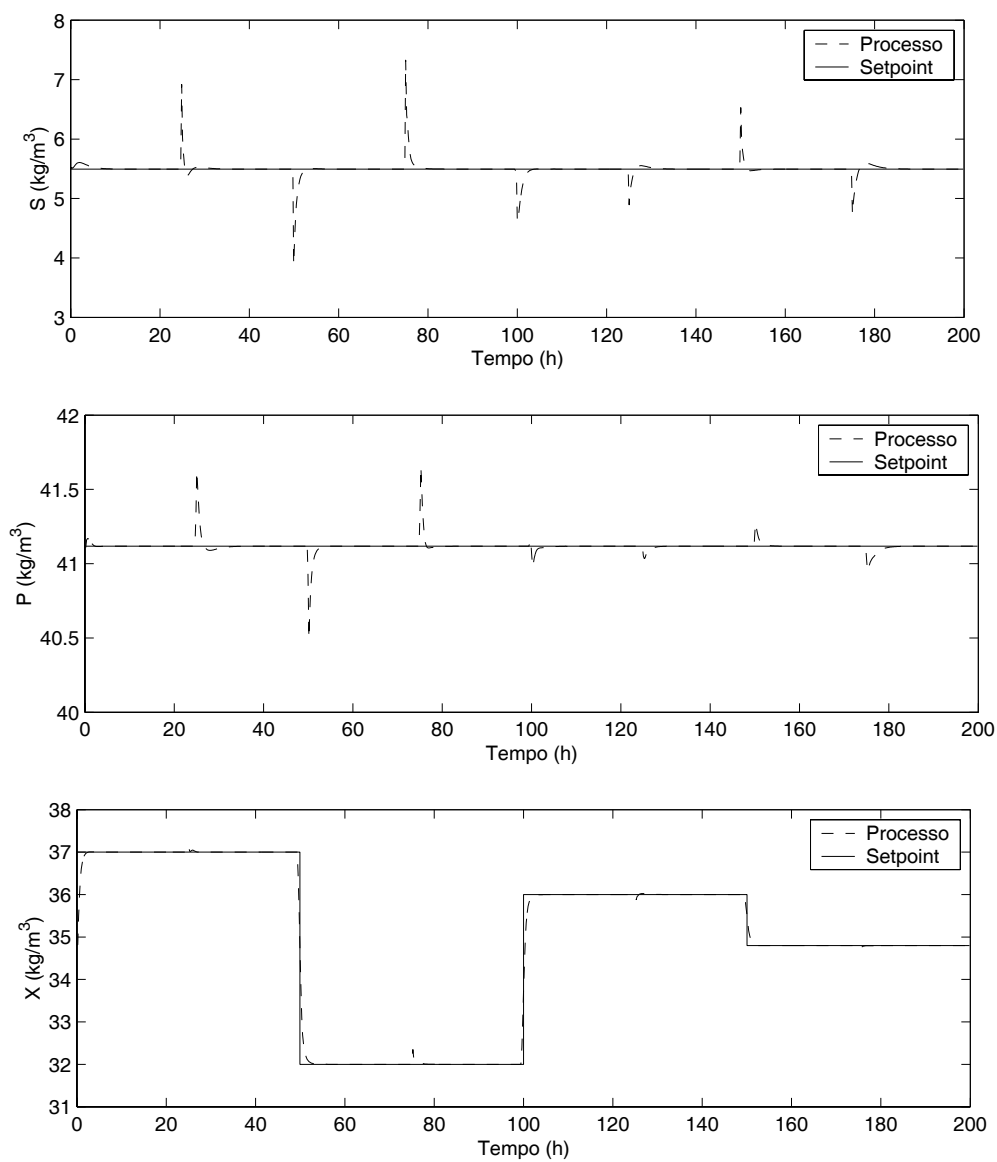


Figura 7.28: Controlador MPC-UGL - Problemas servo para a variável X e regulatório para as variáveis S e P .

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

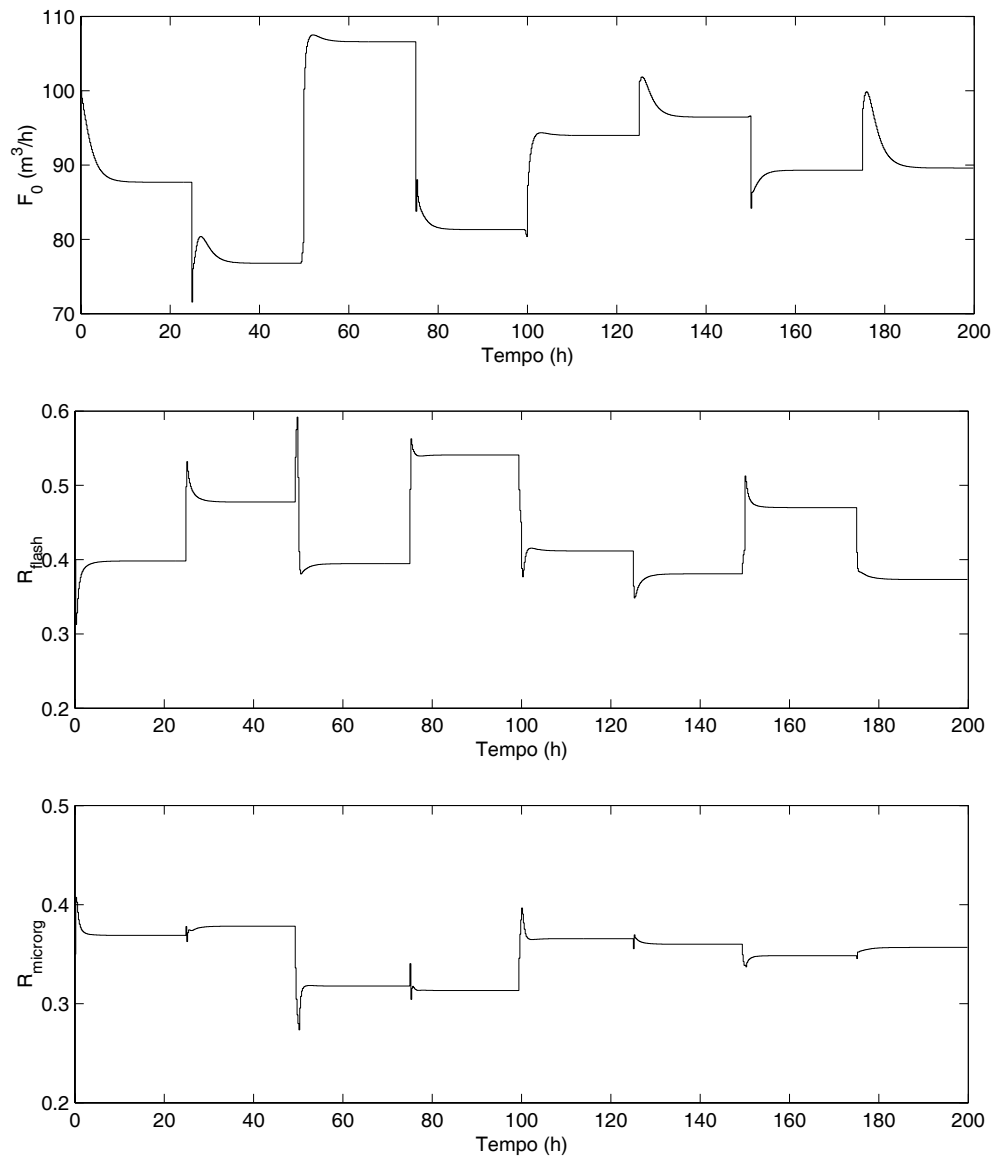


Figura 7.29: Controlador MPC-UGL - Variáveis manipuladas referentes à figura 7.28.

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

Analisando-se as figuras 7.24, 7.26 e 7.28, verifica-se que o desempenho do controlador MPC-UGL multi-variável é bastante semelhante àquele apresentado pelo controlador MPC-MLP. Pode-se também observar que as ações de controle impostas por ambos os controladores foram bastante similares, apresentando variações suaves durante a maior parte da simulação.

7.4.4 Controle Preditivo Linear: O Controlador por Matriz Dinâmica - DMC

Pelos mesmos motivos destacados na seção 6.4.5, apresentam-se nesta seção os resultados da aplicação de um controlador DMC MIMO ao processo de fermentação extrativa, operando sob as mesmas condições descritas anteriormente neste capítulo.

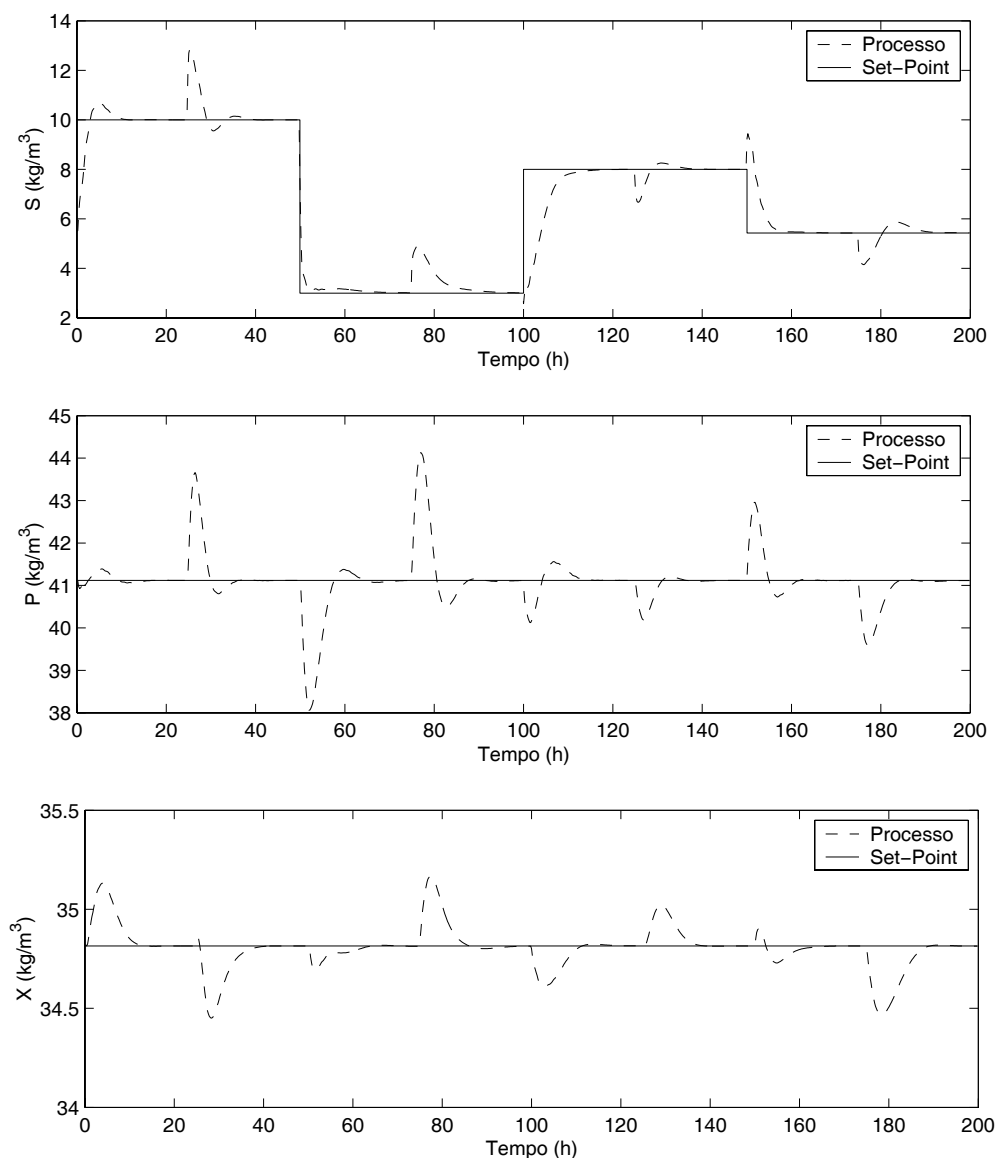


Figura 7.30: Controlador DMC - Problemas servo para a variável S e regulatório para as variáveis P e X .

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

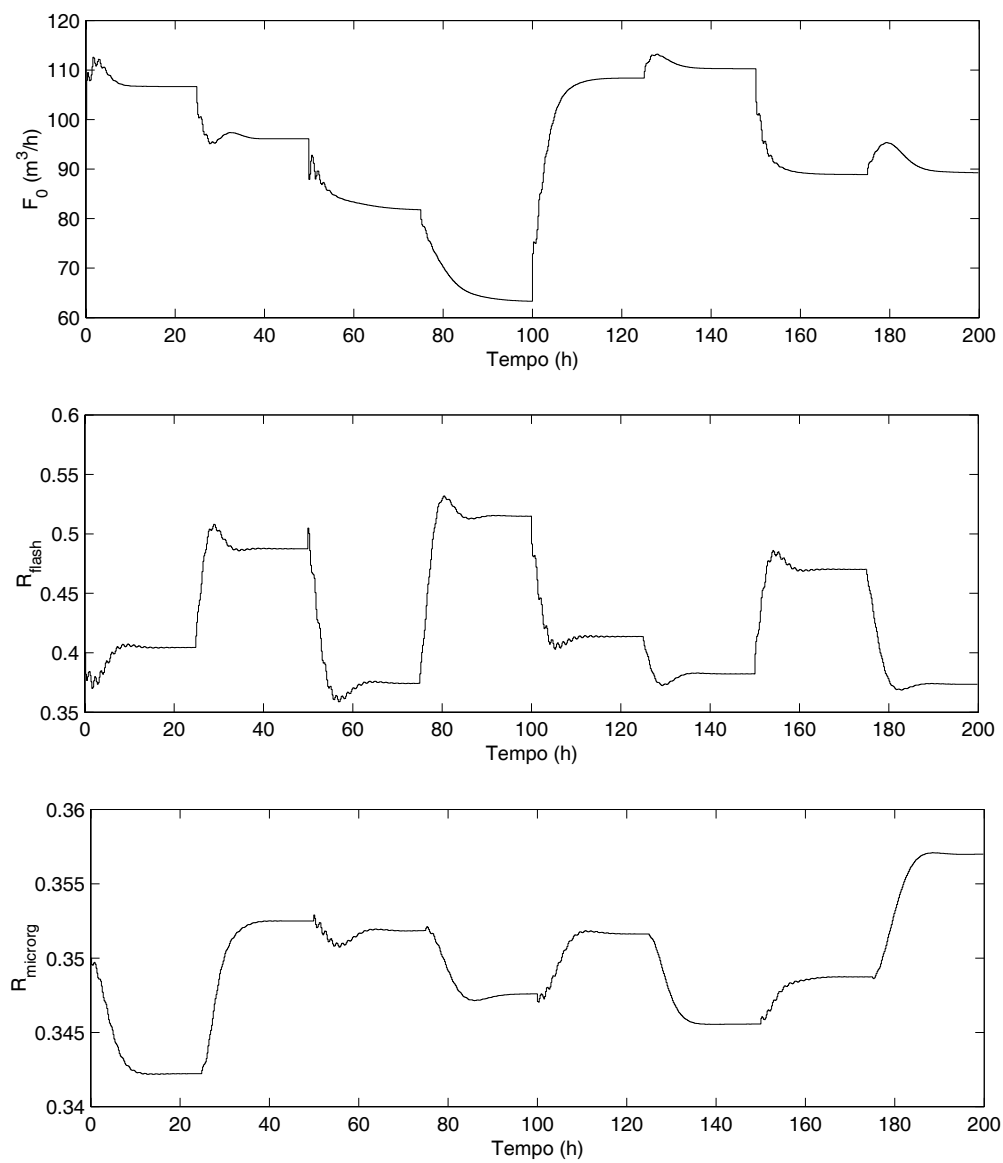


Figura 7.31: Controlador DMC - Variáveis manipuladas referentes à figura 7.30.

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

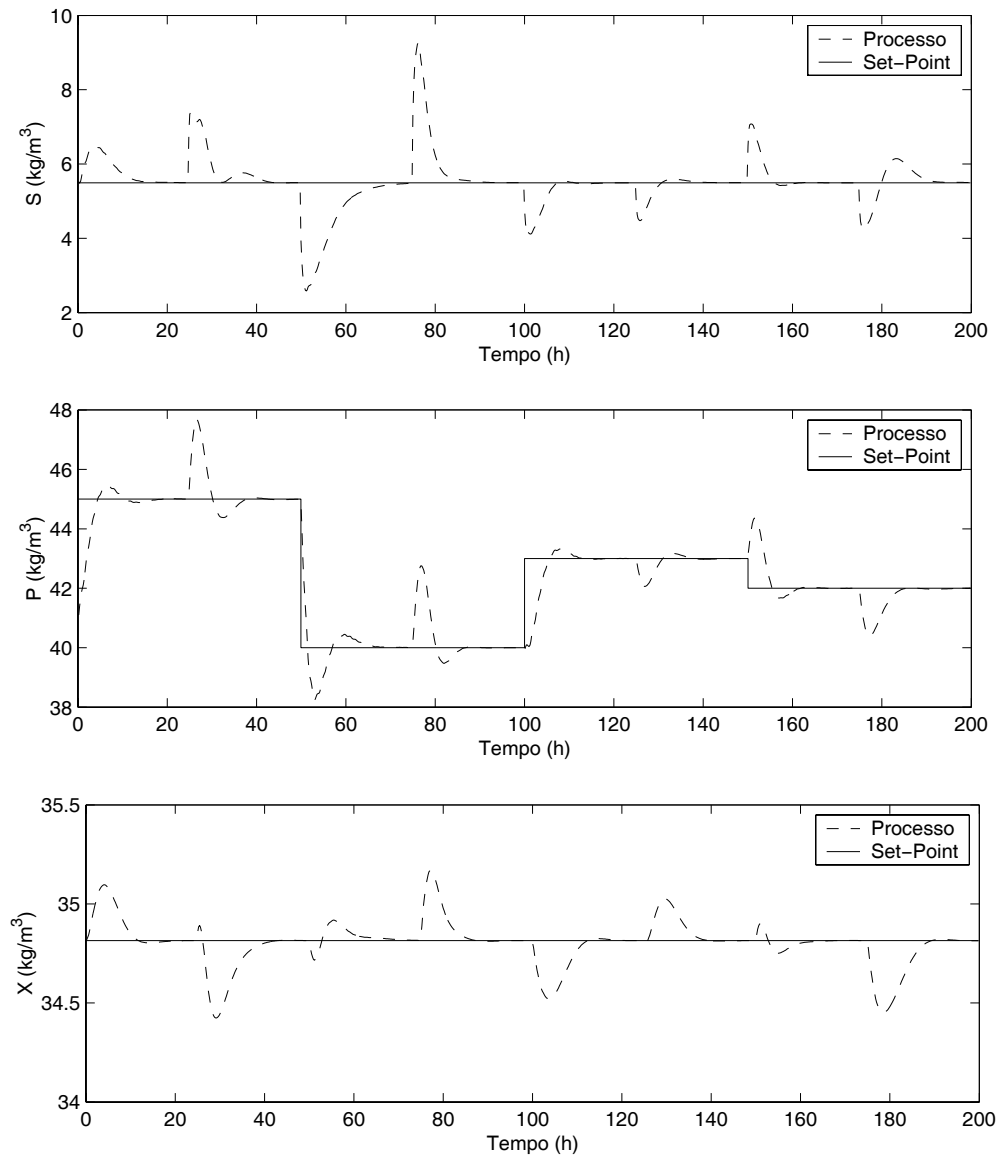


Figura 7.32: Controlador DMC - Problemas servo para a variável P e regulatório para as variáveis S e X .

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

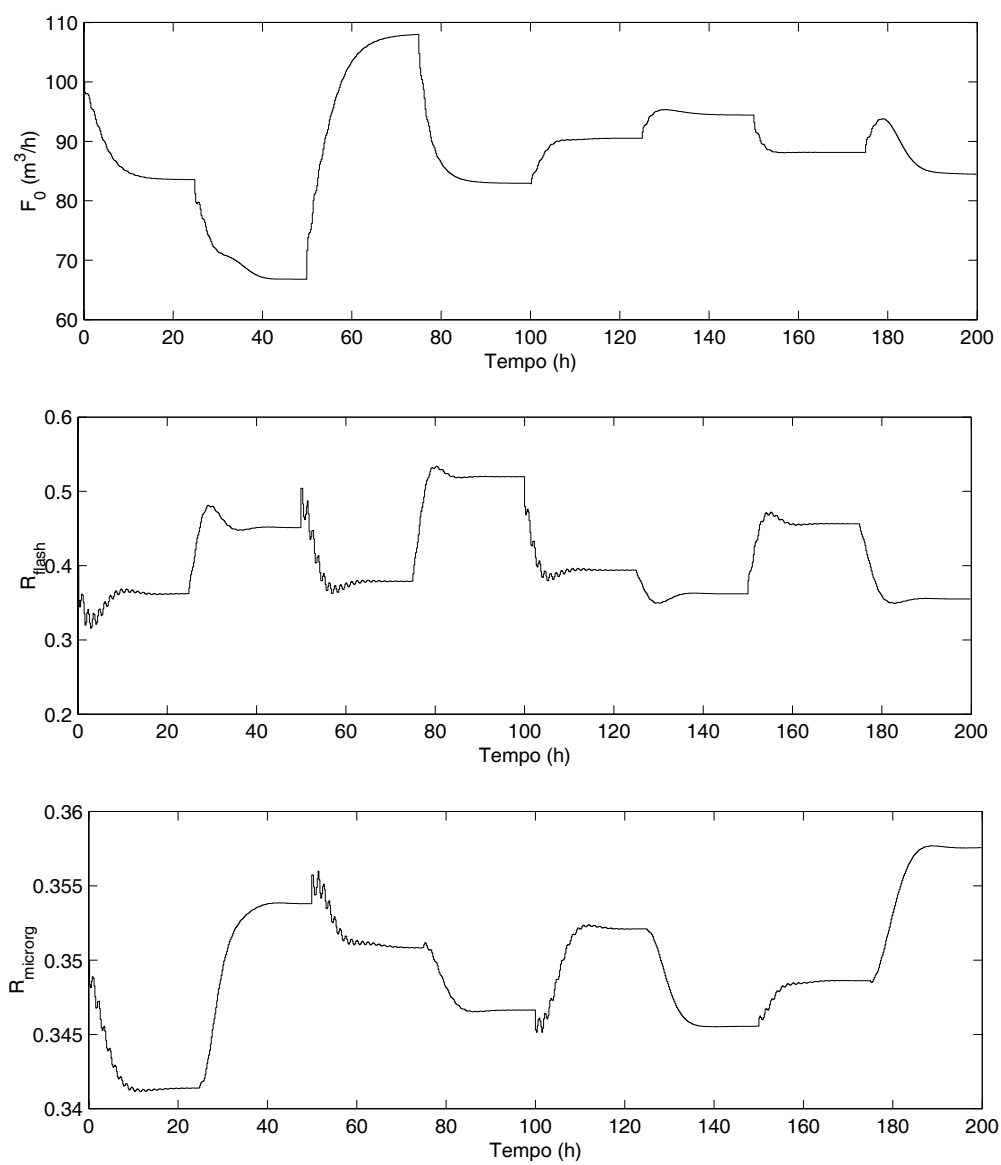


Figura 7.33: Controlador DMC - Variáveis manipuladas referentes à figura 7.32.

ESTUDO DE CASO 2: PROCESSO EXTRATIVO DE FERMENTAÇÃO ALCOÓLICA CONTÍNUA

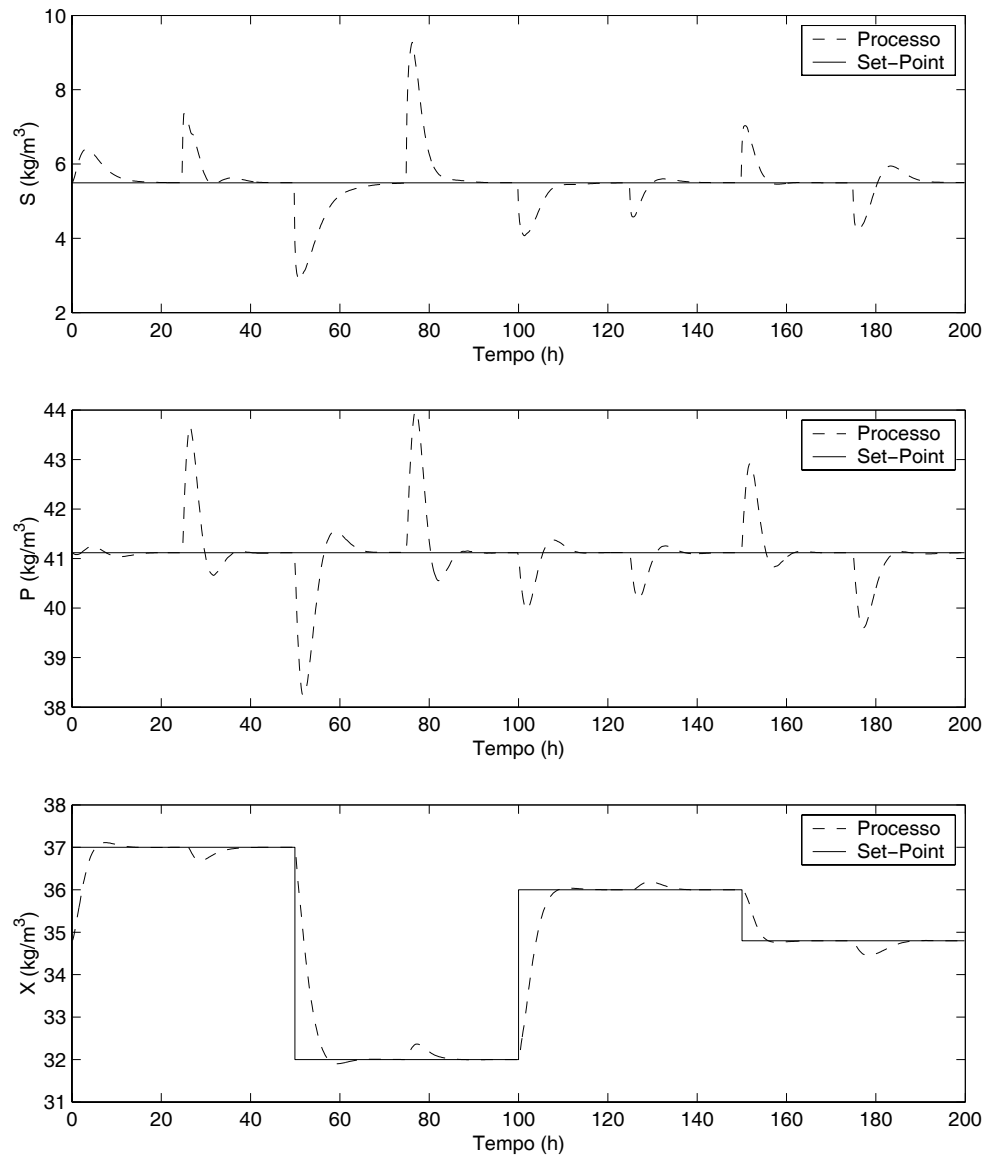


Figura 7.34: Controlador DMC - Problemas servo para a variável X e regulatório para as variáveis S e P .

7.4 Estratégias de Controle para o Processo Extrativo de Fermentação Alcoólica: Caso Multi-Variável

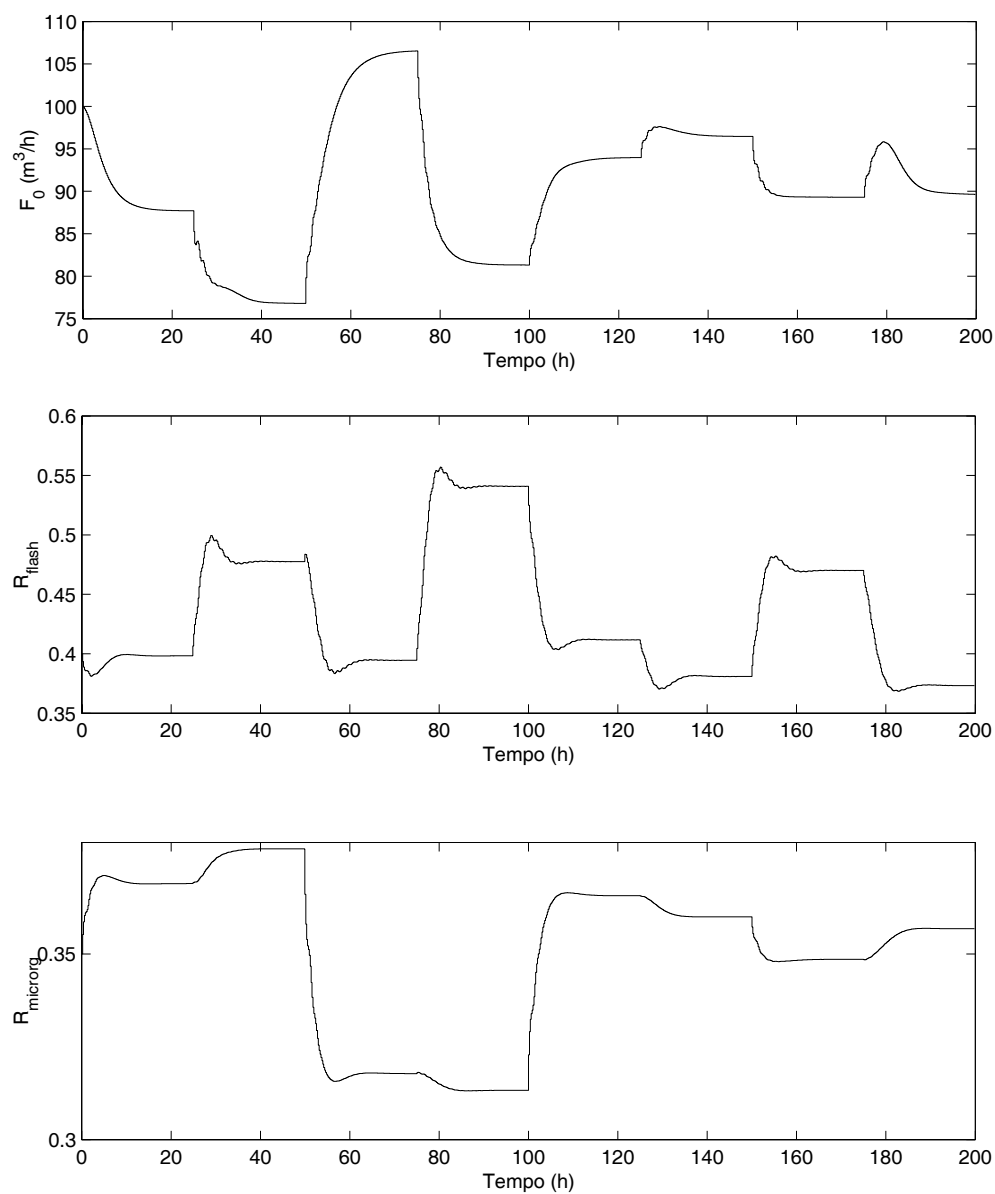


Figura 7.35: Controlador DMC - Variáveis manipuladas referentes à figura 7.34.

Os parâmetros do controlador DMC multi-variável foram sintonizados da seguinte forma: $N_m = 25$ (horizonte de modelo), $N_1 = 1$ (horizonte inicial), $N_y = 5$ (horizonte de predição), $N_u = 1$ (horizonte de controle) e $\lambda = [0, 35 \ 50 \ 50]^T$ (fatores de ponderação).

Analisando-se as figuras 7.30, 7.32 e 7.34, conclui-se que o controlador DMC também foi capaz de realizar o controle multi-variável processo de forma adequada. Contudo, assim como foi observado no *Estudo de Caso 1* (seção 6.4.5), o processo respondeu de forma mais lenta quando comparado com os resultados apresentados pelos controladores não lineares (MPC-MLP e MPC-UGL).

7.5 Conclusões

Ao longo deste capítulo, foram apresentados os resultados da generalização das técnicas de identificação e controle - empregadas no capítulo 6 - para o caso multi-variável. A principal conclusão é que, assim como no caso mono-variável, as técnicas não lineares de identificação e controle propostas nesta tese mostraram-se capazes de lidar com problemas encontrados tipicamente na indústria química, tais como processos não lineares, ruídos nos instrumentos de medida, sensores com tempos de análise relativamente longos, etc.

Com relação à identificação do processo, mostrou-se uma vez mais a eficiência das técnicas não lineares na modelagem de processos dinâmicos. Ressalta-se novamente a necessidade da verificação do desempenho dos modelos através da simulação recursiva quando se pretende utilizá-los para representar o comportamento dinâmico de um processo. Os modelos neurais mostraram-se bastante adequados para este tipo de tarefa, merecendo destaque o algoritmo neural com aprendizado construtivo (UGL), que desponta como uma alternativa muito promissora para a geração de modelos não lineares já que, além de determinar automaticamente (e de forma ótima) a dimensão e a forma das funções de ativação da rede neural, é capaz de gerar modelos mais parcimoniosos e potencialmente mais genéricos.

Capítulo 8

CONSIDERAÇÕES FINAIS

8.1 Conclusões

As principais conclusões sobre o conteúdo dos assuntos abordados nesta tese foram apresentadas em detalhes ao final de cada capítulo. A seguir serão apresentadas as conclusões gerais sobre o trabalho apresentado neste documento.

O principal objetivo desta tese foi apresentar uma contribuição para as áreas de modelagem e controle de processos da indústria química. Os capítulos iniciais deste trabalho apresentam resumida e didaticamente, os fundamentos teóricos sobre a identificação de processos. A justificativa para isso é que a identificação é uma tarefa difícil de ser realizada, além de ser a parte que consome mais tempo no projeto de sistemas de controle.

Foi dada maior ênfase aos modelos não lineares por conta de sua capacidade para descrever mais adequadamente o comportamento não linear típico da maioria dos processos químicos. A abordagem neural foi destacada devido ao seu excelente desempenho na modelagem dos mais diversos problemas não lineares.

Escolheram-se dois tipos de modelos não lineares para modelar os processos estudados: *i)* Dois modelos paramétricos, representados por uma rede neural *feedforward* do tipo perceptron multicamada (MLP) e um modelo nebuloso com estrutura hierárquica e; *ii)* Um modelo não paramétrico, representado por uma rede neural *feedforward*

com estrutura definida automaticamente por um algoritmo de aprendizado construtivo (UGL), utilizando funções de ativação distintas para cada neurônio da camada interna.

Nas duas abordagens, a ênfase foi na obtenção de modelos com dimensão reduzida, evitando assim o principal obstáculo para a maior utilização dos modelos não lineares do tipo caixa-preta na descrição do comportamento dinâmico de processos da indústria química. O modelo UGL destacou-se na tarefa de fornecer modelos mais parcimoniosos, o que pode ser explicado pelo fato de sua formulação considerar o projeto de redes neurais de forma iterativa, adicionando neurônios extras na camada intermediária apenas depois que os neurônios previamente adicionados tenham aproximado a função da melhor maneira possível. Este procedimento mostrou-se capaz de fornecer redes neurais com dimensão ótima e, portanto, com capacidade de generalização potencialmente superior aos outros modelos não lineares.

Os processos estudados foram escolhidos pela complexidade inerente aos processos biotecnológicos, fato que os classifica como estudos de caso bastante interessantes para testar novas estratégias de identificação e controle. Além disso, o etanol vem apresentando uma demanda crescente no cenário energético nacional, o que faz com que os resultados obtidos possam despertar algum interesse prático em um futuro próximo.

O controle preditivo baseado em modelo foi a estratégia de controle escolhida devido ao seu comprovado sucesso em aplicações teóricas e práticas a vários processos da indústria química. Pelo fato do controle por matriz dinâmica (DMC) ser um dos algoritmos de controle mais estudados e aplicados industrialmente, optou-se por implementá-lo em ambos os processos estudados, analisando o seu desempenho frente aos controladores não lineares propostos. No entanto, o foco das aplicações de controle foi o desenvolvimento de uma estrutura básica que pudesse ser adaptada para projetar controladores baseados em modelos lineares e não lineares, para processos mono e multi-variáveis.

Verificou-se através dos resultados apresentados nos capítulos 6 e 7, que as estratégias não lineares de identificação e de controle propostas nesta tese apresentaram resultados muito bons.

8.2 Perspectivas Futuras

Algumas sugestões para trabalhos futuros são apresentadas a seguir:

- Desenvolver uma estratégia de controle que considere explicitamente a possibilidade da incorporação das abordagens fenomenológica e empírica, obtendo o melhor que cada uma delas pode oferecer. Esta estratégia “híbrida” deve ser complementar e não mutuamente exclusiva.
- Testar outras funções ortogonais nos modelos neurais construtivos. As *Wavelets*, por exemplo, apresentaram resultados interessantes na modelagem de processos químicos quando utilizadas como funções de ativação de redes neurais *feedforward* tradicionais. Estes modelos são chamados genericamente de *WaveNets*.
- Desenvolver modelos com estrutura hierárquica nebulosa (ou neural) considerando múltiplas saídas.
- Desenvolver controladores adaptativos não lineares multi-variáveis baseados no conceito neural construtivo. Estes controladores teriam aplicações diretas em processos variantes no tempo, como por exemplo a maioria dos processos da indústria de química fina, que geralmente operam em regime de batelada.
- Explorar as vantagens dos modelos neurais construtivos em uma formulação de controle baseada no conceito do GPC, que são bastante adequados para a filosofia de controle adaptativo clássico.
- Estudos aprofundados sobre a estabilidade e o desenvolvimento de métodos de otimização não linear mais rápidos e eficientes.
- Aplicar as técnicas de identificação e controle propostas nesta tese em estratégias de controle do tipo *Plant Wide Control*. Problemas deste tipo exigem muito mais das técnicas de controle e otimização adotadas, uma vez que o ótimo individual nem sempre (ou quase nunca) é o ótimo geral. Este tipo de estudo tem potencial para implementar melhorias na maioria das plantas químicas. Um exemplo típico são as refinarias de petróleo.

Índice de Autores

- AGUIRRE, L. A., 36, 37, 39, 43, 48, 49,
52, 54, 55, 75, 79, 88, 103, 105–
107, 110, 169, 191, 289
- ALVES, D. M. G., 152, 153, 303
- AMARAL, W. C., 36, 45–47, 67–72, 74,
82, 83, 173, 174, 177, 289, 291,
292, 298, 300
- AMORIM, H. V., 152, 153, 303
- ANDERSON, J. A., 7, 289
- ANDRIETTA, M. G. S., 196, 303
- ANDRIETTA, S. R., 151–153, 156, 158,
160, 166, 196, 198, 201, 211, 225–
227, 236, 289, 290, 301, 303
- ARBIB, M. A., 39, 297
- ARKUN, Y., 143, 300
- ASSIS, A. J., 160, 290
- ATALA, D. I. P., 223, 224, 227, 232, 235,
237, 253, 290, 293
- BÄRMANN, F., 6, 10, 17, 20, 23–25, 290
- BADGWELL, T. A., 118, 119, 121, 122,
126, 128, 135, 144, 215, 301
- BAILEY, J. E., 161, 290
- BAKSHI, B. R., 12, 302
- BARRON, A. R., 77, 290
- BARROS NETO, B., 234, 290
- BASSO, L. C., 152, 153, 303
- BATTITI, R., 20, 290
- BAUGHMAN, D. R., 197, 290
- BENVENISTE, A., 77, 296
- BEQUETTE, B. W., 118, 121, 291
- BIEGLER-KÖNIG, F., 6, 10, 17, 20, 23–
25, 290
- BIEGLER, L. T., 143, 297, 300
- BIENENSTOCK, E., 10, 295
- BILLINGS, S. A., 96, 107, 291, 293
- BLANCH, H. W., 223, 298
- BORDONS, C., 122, 147, 200, 255, 291
- BOX, G. E. P., 234, 291
- BRDYŚ, M. A., 100, 291
- BROOME, P. W., 44, 291
- BRUNS, R. E., 234, 290
- CAJUEIRO, D. O., 80, 291
- CAMACHO, E. F., 122, 147, 200, 255,
291
- CAMPELLO, R. J. G. B., 45–47, 55, 56,
67–74, 82, 83, 173, 174, 177, 186,
188, 291, 292, 298, 300
- CARREIRA-PERPIÑÁN, M. A., 12, 13,
15, 292
- CAVALCANTE JÚNIOR, F. L., 66, 292
- CHAN, L.-W., 115, 297
- CHEN, S., 96, 107, 291, 293
- CHEN, W., 83, 293
- CHRISTEN, M., 223, 293

- CLARKE, D. W., 119, 121, 122, 196, 293
- COOK, P. A., 100, 299
- COSTA, A. C., 223, 224, 227, 232, 235,
237, 253, 290, 293, 298
- COWAN, C. F., 96, 293
- CRAIG, J. C., 223, 302
- CUTLER, C. R., 119, 121, 127, 128, 136,
215, 293
- CYBENKO, G., 77, 293
- CYSEWSKI, G. R., 223, 294
- DE CASTRO, L. N., 8–11, 19, 112, 113,
115, 181, 239, 294
- DECHECHI, E. C., 158, 160, 168, 175,
198, 201, 211, 294
- DELYON, B., 77, 296
- DEMUTH, H. B., 100, 197, 295
- DENKER, J. S., 10, 297
- DOURSAT, R., 10, 295
- EDGAR, T. F., 145, 200, 256, 294
- FABRO, J. L., 153, 156, 290
- FALB, P. L., 39, 297
- FINN, R. K., 223, 301
- FIORIM, F., 153, 156, 290
- FONSECA, E. F., 166, 294
- FRIEDMAN, J. H., 12, 14, 294
- GAMBARO, V. D. V., 153, 156, 290
- GARCIA, C. E., 120, 122, 143, 147, 196,
200, 255, 294, 295, 299
- GATTU, G., 143, 295
- GEMAN, S., 10, 295
- GERULES, M., 197, 300
- GOLDBERG, N., 223, 302
- GOLDEMBERG, J., 152, 154, 155, 295,
299
- GOMA, G., 223, 298
- GOODWIN, G. C., 82, 298
- GRANT, P. M., 96, 293
- HAGAN, M. T., 100, 197, 295
- HANSEN, L. K., 7, 11, 38, 39, 41, 44, 49,
50, 77–79, 81–83, 86, 89, 90, 92,
93, 95, 100, 102, 106, 107, 114,
115, 197, 295, 299
- HASSIBI, B., 10, 295
- HASTIE, T. J., 11, 21, 26, 301
- HAYKIN, S., 10, 67, 82, 295
- HEBB, D. O., 6, 295
- HENRIQUE, H. M., 115, 295
- HENSON, M. A., 122, 141, 142, 196, 296
- HERNÁNDEZ, E., 143, 300
- HIMMELBLAU, D. M., 145, 200, 256, 294
- HINTON, G. E., 6, 301
- HOPFIELD, J. J., 6, 296
- HORNIK, K., 10, 77, 296
- HUNTER, J. S., 234, 291
- HUNTER, W. G., 234, 291
- HUNT, K. J., 197, 304
- HUSSAIN, M. A., 197, 296, 299
- HWANG, J.-N., 21, 26, 296, 297
- ISHIDA, K., 223, 296
- IYODA, E. M., 10, 11, 19, 112, 294, 296
- JALMARSON, H., 77, 296
- JAMALUDDIN, H. B., 107, 291
- JOHNSON, D., 197, 300
- JUDITSKY, A., 77, 296
- JÁCOME, C. R. F., 36, 52, 289

ÍNDICE DE AUTORES

- KALMAN, R. E., 39, 297
KANTER, I., 87, 297
KISNER, R. A., 68, 83, 301
KOHONEN, T., 6, 297
KOSKO, B., 10, 67, 82, 297
KURANTZ, M. J., 223, 302
LAY, S.-R., 21, 26, 296, 297
LE CUN, Y., 10, 87, 297
LEE, J. H., 120–122, 196, 215, 299
LEE, P. L., 128, 299
LEMONS, J. M., 69, 300
LEUNG, C.-S., 115, 297
LIMA, C. A. M., 9, 12, 29, 297
LIMA, E. L., 115, 295
LIU, Y. A., 197, 290
LI, W. C., 143, 297
LJUNG, L., 36, 53, 77, 79, 191, 296, 297
MACEDO, I., 154, 295
MACIEL FILHO, R., 47, 67, 68, 70, 72, 74, 82, 174, 177, 186, 188, 223, 224, 227, 232, 235, 237, 253, 290, 292, 293, 298
MAECHLER, M., 21, 26, 296
MAIORELLA, B. L., 223, 298
MARTIN, R. D., 21, 26, 296
MAUGERI FILHO, F., 153, 156, 160, 225, 289, 290, 301
MAUGERI, F., 221, 223–227, 231, 232, 234, 235, 237, 253, 290, 293, 302
McCULLOCH, W. S., 6, 298
MELEIRO, L. A. C., 47, 67, 68, 70, 72, 74, 82, 174, 177, 186, 188, 224, 237, 253, 292, 293, 298
MIDDLETON, R. H., 82, 298
MILLS, P. M., 79, 197, 298
MINIER, M., 223, 293, 298
MINSKY, M. L., 6, 7, 298
MOHTADI, C., 119, 121, 196, 293
MONBOUQUETTE, H. G., 227, 299
MORARI, M., 120–122, 147, 196, 200, 215, 255, 295, 299
MOREIRA, J. R., 152, 154, 155, 299
MORSHEDI, A. M., 143, 294
MUJTABA, I. M., 197, 299
MØLLER, M. F., 8, 298
NARENDRA, K. S., 92, 96–98, 197, 299
NETTO, M. L. A., 6, 9, 11, 17, 21, 24, 26, 27, 29–33, 115, 303
NEWELL, R. B., 128, 299
NG, G. W., 77, 100, 197, 299
NORTON, J. P., 79, 299
NØRGAARD, M., 7, 11, 38, 39, 41, 44, 49, 50, 77–79, 81–83, 86, 89, 90, 92, 93, 95, 100, 106, 107, 114, 115, 197, 299
OLIVEIRA, G. H. C., 45–47, 300
OLIVEIRA, J. V., 69, 300
OLIVEIRA, N. M. C., 143, 300
OLLIS, D. F., 161, 290
PALMER JR., C. L. S., 153, 156, 290
PAPON, J., 119, 121, 301
PAPPERT, S. A., 6, 7, 298
PARTHASARATHY, K., 92, 96–98, 197, 299
PEARLMUTTER, B. A., 9, 300
PEDRYCZ, W., 65, 174, 300

- PETERSON, T., 143, 300
- PICHÉ, S., 197, 300
- PINHEIRO, E., 10, 11, 19, 294
- PINTO, J. M., 134, 300
- PITTS, W., 6, 298
- POULSEN, N. K., 7, 11, 38, 39, 41, 44, 49, 50, 77–79, 81–83, 86, 89, 90, 92, 93, 95, 100, 106, 107, 114, 115, 197, 299
- PRAUSNITZ, J. M., 229, 301
- PRECHELT, L., 115, 300
- PRETT, D. M., 120, 122, 147, 196, 200, 255, 295, 299
- PSALTIS, D., 100, 300
- QIN, S. J., 118, 119, 121, 122, 126, 128, 135, 144, 215, 301
- RAJU, G. U., 68, 83, 301
- RAMAKER, B. L., 119, 121, 127, 128, 136, 215, 293
- RAMALINGHAN, A., 223, 301
- RAULT, A., 119, 121, 301
- RAVN, O., 7, 11, 38, 39, 41, 44, 49, 50, 77–79, 81–83, 86, 89, 90, 92, 93, 95, 100, 106, 107, 114, 115, 197, 299
- RAWLINGS, J. B., 122, 126, 142, 301
- RENON, H., 223, 229, 293, 301
- RICHALET, J., 119, 121, 301
- RODRIGUES, G. G., 36, 52, 289
- RODRIGUES, M. I., 153, 156, 160, 221, 223–227, 231, 234, 235, 301, 302
- ROOSEN, C. B., 11, 21, 26, 301
- ROSENBLATT, F., 6, 301
- ROSENFELD, E., 7, 289
- RUMELHART, D. E., 6, 301
- SALAMON, P., 102, 295
- SAYYAR-RODSARI, B., 197, 300
- SCARMINIO, I. S., 234, 290
- SCHIMERT, J., 21, 26, 296
- SCHORK, F. J., 143, 300
- SEBORG, D. E., 115, 118, 119, 122, 141, 142, 295, 296, 302
- SHAW, I. S., 59–61, 302
- SHIMIZU, K., 223, 296
- SIDERIS, A., 100, 300
- SILVA, F. L. H., 221, 223–227, 229, 231, 234, 235, 253, 302
- SIMÕES, M. G., 59–61, 302
- SJÖBERG, J., 63, 64, 77, 296, 302
- SOETERBOEK, R., 122–124, 147, 200, 255, 302
- SOLLA, S. A., 10, 87, 297
- SOURIRAJAN, S., 223, 302
- STINCHCOMBE, M., 10, 77, 296
- STOICA, P., 36, 40, 41, 79, 83, 106, 107, 302
- STORK, D. G., 10, 295
- STUETZLE, W., 12, 14, 294
- STUPIELLO, J. P., 152, 153, 156, 160, 289, 290
- SUGENO, M., 66, 302
- SUM, P.-F., 115, 297
- SÖDERSTRÖM, T., 36, 40, 41, 79, 83, 106, 107, 302
- TADÉ, M. O., 79, 197, 298
- TAKAGI, T., 66, 302

ÍNDICE DE AUTORES

TAYLOR, F., 223, 302
TESTUD, J. L., 119, 121, 301
TONG, H., 81, 302
TUFFS, P. S., 119, 121, 196, 293
UTOJO, U., 12, 302
VAN DER SMAGT, P. P., 20, 303
VIEGAS, M. C., 196, 303
VON ZUBEN, F. J., 6, 8–11, 15–17, 19,
21, 24, 26, 27, 29–33, 67, 70,
74, 112, 115, 181, 185, 186, 188,
239, 243, 292, 294, 298, 303
VOON, W. S. F., 107, 291
WANG, L.-X., 58, 63, 68, 83, 293, 303
WHEALS, A. E., 152, 153, 303
WHITE, H., 10, 77, 296
WIENER, N., 44, 304
WILKE, C. R., 223, 294, 298
WILLIAMS, R. J., 6, 301
WITTENMARK, B., 172, 290
WONG, K.-W., 115, 297
YAMAMURA, A. A., 100, 300
YOU, S.-S., 21, 297
ZADEH, L. A., 56, 304
ZAFIRIOU, E., 143, 295
ZANG, Q., 77, 296
ZHOU, J., 68, 83, 301
ZHU, Q. M., 107, 291
ZOMAYA, A. Y., 79, 197, 298
ŻBIKOWSKI, R., 197, 304
ÅSTRÖM, K. J., 172, 290

Referências Bibliográficas

- AGUIRRE, L. A. (2000). *Introdução à Identificação de Sistemas - Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*, Editora UFMG.
- AGUIRRE, L. A., RODRIGUES, G. G. e JÁCOME, C. R. F. (1998). Identificação de sistemas não-lineares utilizando modelos NARMAX polinomiais - uma revisão e novos resultados, *SBA Controle & Automação* **9**(2): 90–106.
- AMARAL, W. C. (1980). *Identificação Recursiva de Sistemas Multivariáveis*, Tese de Doutorado, Universidade Estadual de Campinas - Faculdade de Engenharia - Departamento de Engenharia Elétrica - UNICAMP, Campinas - SP - Brasil.
- ANDERSON, J. A. e ROSENFELD, E. (1988). *Neurocomputing: Foundation of Research*, MIT Press.
- ANDRIETTA, S. R. (1994). *Modelagem, Simulação e Controle de Fermentação Alcoólica Contínua em Escala Industrial*, Tese de Doutorado, Universidade Estadual de Campinas - Faculdade de Engenharia de Alimentos - FEA - UNICAMP, Campinas - SP - Brasil.
- ANDRIETTA, S. R. e MAUGERI FILHO, F. (1994). *Advances in Bioprocess Engineering*, Kluwer Academic Publishers, Capítulo: Optimum Design of a Continuous Fermentation Unit of an Industrial Plant for Alcohol Production, pp. 47–52.
- ANDRIETTA, S. R. e STUPIELLO, J. P. (1990a). Simulação e modelagem para processos de fermentação alcoólica (I) batelada alimentada, *STAB Açúcar, Álcool e Subprodutos* **8**: 36–40.

- ANDRIETTA, S. R. e STUPIELLO, J. P. (1990b). Simulação e modelagem para processos de fermentação alcoólica (II) contínua, *STAB Açúcar, Álcool e Subprodutos* **9**: 45–51.
- ANDRIETTA, S. R., STUPIELLO, J. P., GAMBARO, V. D. V., FIORIM, F., FABRO, J. L., PALMER JR., C. L. S. e MAUGERI FILHO, F. (1994). Fermentação contínua - projeto, controle e aspectos operacionais (Usina Guarani), *STAB Açúcar, Álcool e Subprodutos* **12**(5): 26–30.
- ASSIS, A. J. (1996). *Projeto de Controladores Adaptativos Auto-Ajustáveis*, Dissertação de Mestrado, Universidade Estadual de Campinas - Faculdade de Engenharia Química - FEQ - UNICAMP, Campinas - SP - Brasil.
- ÅSTRÖM, K. J. e WITTENMARK, B. (1997). *Computer-Controlled Systems*, 3rd edn, Prentice Hall.
- ATALA, D. I. P., COSTA, A. C., MACIEL FILHO, R. e MAUGERI, F. (2001). Kinetics of ethanol fermentation with high biomass concentration considering the effect of temperature, *Applied Biochemistry and Biotechnology* **91-93**: 353–365.
- BAILEY, J. E. e OLLIS, D. F. (1986). *Biochemical Engineering Fundamentals*, 2nd edn, McGraw-Hill, New York.
- BÄRMANN, F. e BIEGLER-KÖNIG, F. (1992). On a class of efficient learning algorithms for neural networks, *Neural Networks* **5**(1): 139–144.
- BARRON, A. R. (1993). Universal approximation bounds for superpositions of a sigmoid function, *IEEE Transactions on Information Theory* **39**: 930–945.
- BARROS NETO, B., SCARMINIO, I. S. e BRUNS, R. E. (2001). *Como Fazer Experimentos*, Editora da UNICAMP.
- BATTITI, R. (1992). First- and second-order methods for learning: Between steepest descent and Newton's method, *Neural Computation* **4**(2): 141–166.
- BAUGHMAN, D. R. e LIU, Y. A. (1995). *Neural Networks in Bioprocessing and Chemical Engineering*, Academic Press Inc.

REFERÊNCIAS BIBLIOGRÁFICAS

- BEQUETTE, B. W. (1991). Nonlinear control of chemical processes: A review, *Ind. Eng. Chem. Res.* **30**: 1391–1413.
- BILLINGS, S. A., JAMALUDDIN, H. B. e CHEN, S. (1992). Properties of neural networks with applications to modelling non-linear dynamical systems, *International Journal of Control* **55**(1): 193–224.
- BILLINGS, S. A. e VOON, W. S. F. (1986). Correlation based model validity tests for non-linear models, *International Journal of Control* **44**(1): 235–244.
- BILLINGS, S. A. e ZHU, Q. M. (1994). Nonlinear model validation using correlation tests, *International Journal of Control* **60**(6): 1107–1120.
- BOX, G. E. P., HUNTER, W. G. e HUNTER, J. S. (1978). *Statistics for Experimenters*, John Wiley and Sons.
- BRDYŚ, M. A. (1995). *Advanced Control with MATLAB and SIMULINK*, Ellis Horwood, Capítulo: Neural Networks in Identification and Control.
- BROOME, P. W. (1965). Discrete orthonormal sequences, *Journal of the Association for Computing Machinery* **12**(2): 151–168.
- CAJUEIRO, D. O. (2000). *Controle adaptativo paralelo usando redes neurais com identificação explícita da dinâmica não modelada*, Dissertação de Mestrado, Instituto Tecnológico da Aeronáutica - ITA, São José dos Campos - SP - Brasil.
- CAMACHO, E. F. e BORDONS, C. (1999). *Model Predictive Control*, Springer-Verlag.
- CAMPELLO, R. J. G. B. (2002). *Arquiteturas e Metodologias para Modelagem e Controle de Sistemas Complexos Utilizando Ferramentas Clássicas e Modernas*, Tese de Doutorado, Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação - FEEC - UNICAMP, Campinas - SP - Brasil.
- CAMPELLO, R. J. G. B. e AMARAL, W. C. (1999a). Equivalência entre modelos nebulosos e redes neurais, *Proc. IV Simpósio Brasileiro de Automação Inteligente*, São Paulo - Brasil, pp. 208–212.

- CAMPELLO, R. J. G. B. e AMARAL, W. C. (1999b). Otimização irrestrita de modelos nebulosos e neurais com estrutura em cascata, *Proc. IV Simpósio Brasileiro de Automação Inteligente*, São Paulo - Brasil, pp. 449–454.
- CAMPELLO, R. J. G. B. e AMARAL, W. C. (2000a). Modelos relacionais nebulosos: Aspectos numéricos e lingüísticos, *Proc. Congresso Brasileiro de Automática*, Florianópolis - SC - Brasil, pp. 1896–1901.
- CAMPELLO, R. J. G. B. e AMARAL, W. C. (2000b). Optimization of hierarchical neural fuzzy models, *Proc. IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Como - Italy, p. CD.
- CAMPELLO, R. J. G. B. e AMARAL, W. C. (2001). Modeling and linguistic knowledge extraction from systems using fuzzy relational models, *Fuzzy Sets and Systems* **121**: 113–126.
- CAMPELLO, R. J. G. B. e AMARAL, W. C. (2002). Hierarchical fuzzy relational models: Linguistic interpretation and universal approximation, *Proc. 11th IEEE International Conference on Fuzzy Systems*, Honolulu - USA, p. CD.
- CAMPELLO, R. J. G. B., MELEIRO, L. A. C., AMARAL, W. C. e MACIEL FILHO, R. (2001). Identification of a Bioprocess Using Orthonormal Function Based Models, *Proc. 6th World Congress on Chemical Engineering*, Melbourne - Austrália, p. CD.
- CAMPELLO, R. J. G. B., VON ZUBEN, F. J., AMARAL, W. C., MELEIRO, L. A. C. e MACIEL FILHO, R. (2002). Hierarchical fuzzy models within the framework of orthonormal basis function and their application to bioprocess control, *Submetido para Chemical Engineering Science*.
- CARREIRA-PERPIÑÁN, M. A. (1996). A review of dimension reduction techniques (disponível em: <http://www.dcs.shef.ac.uk/~miguel/papers/cs-96-09.html> - acesso em outubro de 2002), *Technical Report CS-96-09*, Dept. of Computer Science, University of Sheffield - UK.
- CAVALCANTE JÚNIOR, F. L. (2000). *Controle preditivo utilizando um modelo nebuloso*, Dissertação de Mestrado, Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação - FEEC - UNICAMP.

REFERÊNCIAS BIBLIOGRÁFICAS

- CHEN, S., BILLINGS, S. A., COWAN, C. F. e GRANT, P. M. (1990). Practical identification of NARMAX models using radial basis functions, *Int. j. Control* **52**: 1327–1350.
- CHEN, W. e WANG, L.-X. (2000). A note on universal approximation by hierarchical fuzzy systems, *Information Sciences* **123**: 241–248.
- CHRISTEN, M., MINIER, M. e RENON, H. (1990). Ethanol extraction by supported liquid membrane during fermentation, *Biotechnol. Bioengineering* **3**: 116–123.
- CLARKE, D. W. (1994). *Advances in Model Based Predictive Control*, Oxford University Press.
- CLARKE, D. W., MOHTADI, C. e TUFFS, P. S. (1987a). Generalized predictive control 1. The basic algorithm, *Automatica* **23**: 137–148.
- CLARKE, D. W., MOHTADI, C. e TUFFS, P. S. (1987b). Generalized predictive control 2. Extensions and interpretations, *Automatica* **23**: 149–160.
- COSTA, A. C., ATALA, D. I. P., MAUGERI, F. e MACIEL FILHO, R. (2001). Factorial design and simulation for the optimization and determination of control structures for an extractive alcoholic fermentation, *Process Biochemistry* **37**: 125–137.
- COSTA, A. C., MELEIRO, L. A. C. e MACIEL FILHO, R. (2002a). Controle preditivo não linear de um processo de fermentação alcoólica extrativa, *Anais do XIV Congresso Brasileiro de Engenharia Química*, Natal - RN - Brasil, pp. CD-ROM.
- COSTA, A. C., MELEIRO, L. A. C. e MACIEL FILHO, R. (2002b). Non-linear predictive control of an extractive alcoholic fermentation process, *Process Biochemistry* **38**(5): 743–750.
- CUTLER, C. R. e RAMAKER, B. L. (1979). Dynamic matrix control - A computer control algorithm, *AIChE 86th National Meeting, Houston - Texas - USA*.
- CUTLER, C. R. e RAMAKER, B. L. (1980). Dynamic matrix control - A computer control algorithm, *Proc. Joint Automatic Control Conference - Paper WP5-B, San Francisco - CA - USA*.
- CYBENKO, G. (1989). Approximations by superposition of a sigmoidal function, *Mathematics of Control Signals and Systems* **2**: 303–314.

- CYSEWSKI, G. R. e WILKE, C. R. (1977). Rapid ethanol fermentation using vacuum and cell recycle, *Biotechnol. Bioengineering* **19**: 1125–1143.
- DE CASTRO, L. N. (1998). *Análise e Síntese de Estratégias de Aprendizizado para Redes Neurais Artificiais*, Dissertação de Mestrado, Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação - FEEC - UNICAMP, Campinas - SP - Brasil.
- DE CASTRO, L. N., IYODA, E. M., PINHEIRO, E. e VON ZUBEN, F. J. (1999). Redes neurais construtivas: Uma abordagem comparativa, *Proceedings of the IVth Brazilian Conference on Neural Networks*, ITA, São José dos Campos - SP - Brazil, pp. 102–107.
- DE CASTRO, L. N. e VON ZUBEN, F. J. (1998a). Optimised Training Techniques for Feedforward Neural Networks, *Technical Report DCA-RT 03/98*, Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação - FEEC - UNICAMP.
- DE CASTRO, L. N. e VON ZUBEN, F. J. (1998b). Uma alternativa simples e robusta para inicialização de pesos em redes neurais multicamadas, *Anais do V Simpósio Brasileiro de Redes Neurais*, Vol. 2, pp. 97–102.
- DECHECHI, E. C. (1998). *Controle Avançado Preditivo Adaptativo “DMC Multivariável Adaptativo”*, Tese de Doutorado, Universidade Estadual de Campinas - Faculdade de Engenharia Química - FEQ - UNICAMP, Campinas - SP - Brasil.
- EDGAR, T. F. e HIMMELBLAU, D. M. (1988). *Optimization of Chemical Processes*, McGraw-Hill.
- FONSECA, E. F. (1998). *Controle Preditivo Baseado em Redes Neurais de Fermentação Alcoólica Contínua*, Dissertação de Mestrado, DEQ - UFPE.
- FRIEDMAN, J. H. e STUETZLE, W. (1981). Projection pursuit regression, *Journal of the American Statistical Association* **76**(376): 817–823.
- GARCIA, C. E. e MORSHEDI, A. M. (1986). Quadratic programming solution of dynamic matrix control (QDMC), *Chemical Engineering Communications* **46**: 73–87.

REFERÊNCIAS BIBLIOGRÁFICAS

- GARCIA, C. E., PRETT, D. M. e MORARI, M. (1989). Model predictive control: Theory and practice - a survey, *Automatica* **25**(3): 335–348.
- GATTU, G. e ZAFIRIOU, E. (1992). Nonlinear quadratic dynamic matrix control with state estimation, *Industrial & Engineering Chemistry Research* **31**(4): 1096–1104.
- GATTU, G. e ZAFIRIOU, E. (1995). Observer based nonlinear quadratic dynamic matrix control for state space and input-output models, *Canadian Journal of Chemical Engineering* **73**: 883–895.
- GEMAN, S., BIENENSTOCK, E. e DOURSAT, R. (1992). Neural networks and the bias/variance dilemma, *Neural Computation* **4**(1): 1–58.
- GOLDEMBERG, J. e MACEDO, I. (1994). The brazilian alcohol program - an overview, *Energy for Sustainable Development* **1**(1): 17–22.
- HAGAN, M. T. e DEMUTH, H. B. (1999). Neural networks for control, *Proceedings of the American Control Conference*, San Diego, California.
- HANSEN, L. K. e SALAMON, P. (1990). Neural network ensembles, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**: 993–1001.
- HASSIBI, B. e STORK, D. G. (1993). Second order derivatives for network pruning: Optimal brain surgeon, in S. J. Hanson, J. D. Cowan e C. L. Giles (eds), *Advances in Neural Information Processing Systems*, Vol. 5, Morgan Kauffman, San Mateo, CA, pp. 598–605.
- HAYKIN, S. (1999). *Neural Networks: A Comprehensive Foundation*, 2nd edn, Prentice Hall.
- HEBB, D. O. (1949). *The Organization of Behaviour*, Wiley.
- HENRIQUE, H. M. (1999). *Uma Contribuição ao Estudo de Redes Neurais Aplicadas ao Controle de Processos*, Tese de Doutorado, Universidade Federal do Rio de Janeiro - UFRJ - COPPE - PEQ, Rio de Janeiro - RJ - Brasil.
- HENRIQUE, H. M., LIMA, E. L. e SEBORG, D. E. (2000). Model structure determination in neural network models, *Chemical Engineering Science* **55**(22): 5457–5469.

- HENSON, M. A. (1998). Nonlinear model predictive control: Current status and future directions, *Computers and Chemical Engineering* **23**: 187–202.
- HENSON, M. A. e SEBORG, D. E. (1997). *Nonlinear Process Control*, Prentice Hall.
- HOPFIELD, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities, *J. Proc. of the National Academy Sciences of the USA* **79**: 2554–2558.
- HORNIK, K., STINCHCOMBE, M. e WHITE, H. (1989). Multi-layer feedforward networks are universal approximators, *Neural Networks* **2**(5): 359–366.
- HORNIK, K., STINCHCOMBE, M. e WHITE, H. (1990). Universal approximation of an unknown function and its derivatives using multilayer feedforward networks, *Neural Networks* **3**(5): 551–560.
- HUSSAIN, M. A. (1999). Review of the applications of neural networks in chemical process control - simulation and on-line implementation, *Artificial Intellig. Engineering* **13**: 55–68.
- HWANG, J.-N., LAY, S.-R., MAECHLER, M., MARTIN, R. D. e SCHIMERT, J. (1994). Regression modeling in back-propagation and projection pursuit learning, *IEEE Transactions on Neural Networks* **5**(3): 342–353.
- ISHIDA, K. e SHIMIZU, K. (1996). Novel repeated batch operation for flash fermentation system: Experimental data and mathematical modelling, *Journal Chem. Technol. Biotechnol.* **66**: 340–346.
- IYODA, E. M. (2000). *Inteligência Computacional no Projeto Automático de Redes Neurais Híbridas e Redes Neurofuzzy Heterogêneas*, Dissertação de Mestrado, Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação - FEEC - UNICAMP, Campinas - SP - Brasil.
- JUDITSKY, A., JALMARSON, H., BENVENISTE, A., DELYON, B., LJUNG, L., SJÖBERG, J. e ZANG, Q. (1995). Nonlinear black-box models in system identification: Mathematical foundations, *Automatica* **31**(12): 1725–1750.

REFERÊNCIAS BIBLIOGRÁFICAS

- KALMAN, R. E., FALB, P. L. e ARBIB, M. A. (1969). *Topics in Mathematical System Theory*, McGraw-Hill.
- KOHONEN, T. (1982). Self-organized formulation of topologically correct feature maps, *Biological Cybernetics* **43**: 59–69.
- KOSKO, B. (1992). *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall.
- KOSKO, B. (1997). *Fuzzy Engineering*, Prentice Hall.
- LAY, S.-R., HWANG, J.-N. e YOU, S.-S. (1994). Extensions to projection pursuit learning networks with parametric smoothers, *Proceedings of IEEE ICNN'94*, Vol. 3, pp. 1325–1330.
- LE CUN, Y., DENKER, J. S. e SOLLA, S. A. (1990). Optimal brain damage, in D. S. Touretzky (ed.), *Advances in Neural Information Processing Systems II*, Morgan Kaufman, San Mateo, CA, pp. 598–605.
- LE CUN, Y., KANTER, I. e SOLLA, S. A. (1991). Eigenvalues of covariance matrices: application to a neural-network learning, *Physical Review Letters* **66**(18): 2396–2399.
- LEUNG, C.-S., WONG, K.-W., SUM, P.-F. e CHAN, L.-W. (2001). A pruning method for the recursive least squares algorithm, *Neural Networks* **14**(2): 147–174.
- LI, W. C. e BIEGLER, L. T. (1989). Multistep Newton-type control strategies for constrained nonlinear processes, *Chemical Engineering Research & Design* **67**: 562–577.
- LIMA, C. A. M. (2000). *Emprego de Teoria de Agentes no Desenvolvimento de Dispositivos Neurocomputacionais Híbridos e Aplicação ao Controle e Identificação de Sistemas Dinâmicos*, Dissertação de Mestrado, Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação - FEEC - UNICAMP, Campinas - SP - Brasil.
- LJUNG, L. (1999). *System Identification: Theory for The User*, 2nd edn, Prentice Hall.

- MAIORELLA, B. L., BLANCH, H. W. e WILKE, C. R. (1984). Biotechnology report - economic evaluation of alternative ethanol fermentation processes, *Biotechnol. Bioengineering* **26**: 1003–1025.
- MCCULLOCH, W. S. e PITTS, W. (1943). A logical calculus of ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics* **5**: 115–133.
- MELEIRO, L. A. C., CAMPELLO, R. J. G. B., AMARAL, W. C. e MACIEL FILHO, R. (2002a). Application of hierarchical neural fuzzy models to modeling and control of a bioprocess, *Submetido para Computers and Chemical Engineering* .
- MELEIRO, L. A. C., CAMPELLO, R. J. G. B., MACIEL FILHO, R. e VON ZUBEN, F. J. (2002b). Identification of a multivariate fermentation process using constructive learning, *Proc. SBRN'2002 - VII Brazilian Symposium on Neural Networks*, IEEE Computer Society.
- MELEIRO, L. A. C., COSTA, A. C. e MACIEL FILHO, R. (2002c). Non-linear multivariable predictive control of an alcoholic fermentation process using functional link networks, *Submetido para Process Biochemistry* .
- MELEIRO, L. A. C., MACIEL FILHO, R., CAMPELLO, R. J. G. B. e AMARAL, W. C. (2001). *Application of Neural Networks and Other Learning Technologies in Process Engineering*, Imperial College Press, Capítulo: Hierarchical Neural Fuzzy Models as a Tool for Process Identification: A Bioprocess Application.
- MIDDLETON, R. H. e GOODWIN, G. C. (1990). *Digital Control and Estimation: A Unified Approach*, Prentice Hall, Englewood Cliffs.
- MILLS, P. M., ZOMAYA, A. Y. e TADÉ, M. O. (1996). *Neuro-Adaptive Process Control - A Practical Approach*, John Wiley & Sons.
- MINIER, M. e GOMA, G. (1982). Ethanol production by extractive fermentation, *Biotechnol. Bioengineering* **24**: 1565–1579.
- MINSKY, M. L. e PAPPERT, S. A. (1969). *Perceptrons*, MIT Press.
- MØLLER, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks* **6**: 525–533.

REFERÊNCIAS BIBLIOGRÁFICAS

- MONBOUQUETTE, H. G. (1987). Models for high cell density bioreactors must consider biomass volume fraction: Cell recycle example, *Biotechnology Bioengineering* **39**: 498–503.
- MONBOUQUETTE, H. G. (1992). Modeling high-biomass-density cell recycle fermentors, *Biotechnology Bioengineering* **29**: 1075–1080.
- MORARI, M., GARCIA, C. E., LEE, J. H. e PRETT, D. M. (1994). *Model Predictive Control*, Prentice Hall.
- MORARI, M. e LEE, J. H. (1999). Model predictive control: Past, present and future, *Computers and Chemical Engineering* **23**: 667–682.
- MOREIRA, J. R. e GOLDEMBERG, J. (2002). O Programa do Álcool (disponível em: <http://www.mct.gov.br/clima/comunicold/alcohol.htm> - acesso em 27 maio 2002), *Technical report*, Universidade de São Paulo - Instituto de Eletrotécnica e Energia IEE/USP.
- MUJTABA, I. M. e HUSSAIN, M. A. (eds) (2001). *Application of Neural Networks and Other Learning Technologies in Process Engineering*, Imperial College Press.
- NARENDRA, K. S. e PARTHASARATHY, K. (1990). Identification and control of dynamic systems using neural networks, *IEEE Trans. on Neural Networks* **1**(1): 4–27.
- NEWELL, R. B. e LEE, P. L. (1989). *Applied Process Control - A Case Study*, Prentice Hall.
- NG, G. W. (1997). *Application of Neural Networks to Adaptive Control of Nonlinear System*, Research Studies Press LTD.
- NG, G. W. e COOK, P. A. (1998). On-line adaptive control of non-linear plants using neural networks with applications to liquid level control, *Int. J. Adapt. Control Signal Process.* .
- NØRGAARD, M., RAVN, O., POULSEN, N. K. e HANSEN, L. K. (2000). *Neural Networks for Modelling and Control of Dynamic Systems*, Springer-Verlag.
- NORTON, J. P. (1986). *An Introduction to Identification*, Academic Press.

- OLIVEIRA, G. H. C., CAMPELLO, R. J. G. B. e AMARAL, W. C. (1999). Fuzzy models within orthonormal basis function framework, *Proc. 8th IEEE Internat. Conference on Fuzzy Systems*, Seoul/Korea, pp. 957–962.
- OLIVEIRA, J. V. e LEMOS, J. M. (1997). Simplifying fuzzy relational structures for adaptive control, *Proc. 7th IFSA World Congress*, Prague - Czech Republic, pp. 330–335.
- OLIVEIRA, N. M. C. e BIEGLER, L. T. (1995). An extension of Newton-type algorithms for nonlinear process control, *Automatica* **31**(2): 281–286.
- PEARLMUTTER, B. A. (1994). Fast exact multiplication by the hessian, *Neural Computation* **6**(1): 147–160.
- PEDRYCZ, W. (1995). *Fuzzy Sets Engineering*, CRC Press.
- PEDRYCZ, W. (ed.) (1996). *FUZZY MODELLING - Paradigms and Practice*, Kluwer Academic Publishers.
- PETERSON, T., HERNÁNDEZ, E., ARKUN, Y. e SCHORK, F. J. (1992). A nonlinear DMC algorithm and its applications to a semibatch polymerization reactor, *Chemical Engineering Science* **47**(4): 737–753.
- PICHÉ, S., SAYYAR-RODSARI, B., JOHNSON, D. e GERULES, M. (2000). Nonlinear model predictive control using neural networks, *IEEE Control Systems Magazine* pp. 53–62.
- PINTO, J. M. (1990). *Controle por matriz dinâmica em coluna de destilação*, Dissertação de Mestrado, PEQ - COPPE - UFRJ, Rio de Janeiro - RJ - Brasil.
- PRECHELT, L. (1996). Comparing adaptive and non-adaptive connection pruning with pure early stopping, *Progress in Neural Information Processing* pp. 46–52.
- PRECHELT, L. (1997). Connection pruning with static and adaptive pruning schedules, *Neurocomputing* **16**: 49–611.
- PSALTIS, D., SIDERIS, A. e YAMAMURA, A. A. (1988). A multilayer neural network controller, *IEEE Control Systems Magazine* **8**: 17–21.

REFERÊNCIAS BIBLIOGRÁFICAS

- QIN, S. J. e BADGWELL, T. A. (1997). An overview of industrial model predictive control technology, *in* J. C. KANTOR, C. E. GARCIA e B. CARNAHAN (eds), *Chemical Process Control*, Vol. 93 of *AIChE Symposium Series*, pp. 232–256.
- QIN, S. J. e BADGWELL, T. A. (1998). An overview of nonlinear model predictive control applications, *Proc. IFAC Workshop on Nonlinear Model Predictive Control - Assessment and Future Directions*, Ascona - Switzerland.
- RAJU, G. U., ZHOU, J. e KISNER, R. A. (1991). Hierarchical fuzzy control, *International Journal of Control* pp. 1201–1216.
- RAMALINGHAN, A. e FINN, R. K. (1977). The vacuform process: A new approach to fermentation alcohol, *Biotechnol. Bioengineering* **19**: 583–589.
- RAWLINGS, J. B. (2000). Tutorial overview of model predictive control, *IEEE Control Systems Magazine* (SPECIAL SECTION - Industrial Process Control): 38–51.
- RENON, H. e PRAUSNITZ, J. M. (1968). Local compositions in thermodynamic excess functions for liquid mixtures, *AIChE Journal* **14**(1): 135–144.
- RICHALET, J., RAULT, A., TESTUD, J. L. e PAPON, J. (1976). Algorithmic control of industrial processes, *Proc. of the 4th IFAC Symposium on Identification and System Parameter Estimation*, pp. 1119–1167.
- RODRIGUES, M. I., ANDRIETTA, S. R. e MAUGERI FILHO, F. (1992). Simulação da produtividade e rendimento em fermentação alcoólica contínua para reatores operando em condições ideais e não-ideais de mistura, *STAB Açúcar, Alcool e Subprodutos* pp. 35–41.
- ROOSEN, C. B. e HASTIE, T. J. (1994). Automatic smoothing spline projection pursuit, *Journal of Computational and Graphical Statistics* **3**: 235–248.
- ROSENBLATT, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain, *Psychology Review* **65**: 386–408.
- RUMELHART, D. E., HINTON, G. E. e WILLIAMS, R. J. (1986). *Parallel Distributed Processing*, MIT Press, Capítulo: Learning internal representations by error propagation.

- SÖDERSTRÖM, T. e STOICA, P. (1989). *System Identification*, Prentice Hall.
- SEBORG, D. E. (1999). *Advances in Control*, Springer-Verlag, Capítulo: A Perspective on Advanced Strategies for Process Control (Revisited).
- SHAW, I. S. e SIMÕES, M. G. (1999). *Controle e Modelagem Fuzzy*, Editora Edgard Blücher LTDA.
- SILVA, F. L. H. (1998). *Modelagem, Simulação e Controle de Fermentação Alcoólica Contínua Extrativa*, Tese de Doutorado, Universidade Estadual de Campinas - Faculdade de Engenharia de Alimentos - FEA - UNICAMP, Campinas - SP - Brasil.
- SILVA, F. L. H., RODRIGUES, M. I. e MAUGERI, F. (1999). Dynamic modelling, simulation and optimization of an extractive continuous alcoholic fermentation process, *Journal of Chemical Technology and Biotechnology* **74**: 176–182.
- SJÖBERG, J. (1995). *Non-Linear System Identification with Neural Networks*, Tese de Doutorado, Universidade de Linköping - Dept. de Engenharia Elétrica, Suécia.
- SOETERBOEK, R. (1992). *Predictive Control - A Unified Approach*, Prentice Hall.
- SOURIRAJAN, S. (1987). An approach to reverse osmosis/ultrafiltration. membrane design and development, *Membrane* **12**: 367–379.
- TAKAGI, T. e SUGENO, M. (1985). Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Systems, Man and Cybernetics* **SMC-15**: 116–132.
- TAYLOR, F., KURANTZ, M. J., GOLDBERG, N. e CRAIG, J. C. (1998). Kinetics of continuous fermentation and stripping of ethanol, *Biotechnology Letters* **20**: 67–72.
- TONG, H. (1990). *Nonlinear Time Series, a Dynamical Systems Approach*, Clarendon Press - Oxford, UK.
- UTOJO, U. e BAKSHI, B. R. (1995). *Neural Networks in Bioprocessing and Chemical Engineering*, Academic Press Inc., Capítulo: Connections between Neural Networks and Multivariate Statistical Methods: An Overview.

REFERÊNCIAS BIBLIOGRÁFICAS

- VAN DER SMAGT, P. P. (1994). Minimisation methods for training feedforward neural networks, *Neural Networks* **7**(1): 1–11.
- VIEGAS, M. C., ANDRIETTA, S. R. e ANDRIETTA, M. G. S. (2002). Use of tower reactors for continuous ethanol production, *Brazilian Journal of Chemical Engineering* **19**(2): 167–173.
- VON ZUBEN, F. J. (1996). *Modelos Paramétricos e Não-Paramétricos de Redes Neurais Artificiais e Aplicações*, Tese de Doutorado, Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação - FEEC - UNICAMP, Campinas - São Paulo - Brasil.
- VON ZUBEN, F. J. e NETTO, M. L. A. (1995). Unit-growing learning optimizing the solvability condition for model-free regression, *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 2, pp. 795–800.
- VON ZUBEN, F. J. e NETTO, M. L. A. (1996). Implementação Computacional de Algoritmos de Treinamento para Redes Neurais Paramétricas e Não-Paramétricas, *Technical Report DCA-RT 01/96*, Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação - DCA - FEEC - UNICAMP.
- VON ZUBEN, F. J. e NETTO, M. L. A. (1997). Projection pursuit and the solvability condition applied to constructive learning, *Proceedings of the International Joint Conference on Neural Networks*, Vol. 2, Houston - USA, pp. 1062–1067.
- WANG, L.-X. (1994). *Adaptive Fuzzy Systems and Control - Design and Stability Analysis*, Prentice Hall.
- WANG, L.-X. (1998). Universal approximation by hierarchical fuzzy systems, *Fuzzy Sets and Systems* **93**: 223–230.
- WANG, L.-X. (1999). Analysis and design of hierarchical fuzzy systems, *IEEE Transactions on Fuzzy Systems* **7**: 617–624.
- WHEALS, A. E., BASSO, L. C., ALVES, D. M. G. e AMORIM, H. V. (1999). Fuel ethanol after 25 years, *TIBTECH* **17**: 482–487.

REFERÊNCIAS BIBLIOGRÁFICAS

- WIENER, N. (1956). *Modern Mathematics for the Enginner*, McGraw-Hill, New York,
Capítulo: The Theory of Prediction.
- ZADEH, L. A. (1965). Fuzzy sets, *Information and Control* **8**: 338–353.
- ŻBIKOWSKI, R. e HUNT, K. J. (eds) (1996). *Neural Adaptive Control Technology*, Vol. 15
of *World Scientific Series in Robotics and Intelligent Systems*, World Scientific.