

Universidade Estadual de Campinas - UNICAMP
Faculdade de Engenharia Elétrica e de Computação - FEEC
Depto. de Engenharia de Computação e Automação Industrial - DCA

Arquiteturas e Metodologias para Modelagem e Controle de Sistemas Complexos Utilizando Ferramentas Clássicas e Modernas

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da
Universidade Estadual de Campinas, como parte dos requisitos exigidos para a
obtenção do título de DOUTOR EM ENGENHARIA ELÉTRICA.

Ricardo José Gabrielli Barreto Campello

Mestre em Engenharia Elétrica - FEEC/UNICAMP

Banca Examinadora

Wagner Caradori do Amaral

Orientador

Gérard Favier

João César Moura Mota

Marcelo Carvalho Minhoto Teixeira

João Marcos Travassos Romano

Fernando Antonio Campos Gomide

Fernando José Von Zuben

Campinas, Julho de 2002

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

C153a Campello, Ricardo José Gabrielli Barreto
Arquiteturas e metodologias para modelagem e
controle de sistemas complexos utilizando ferramentas
clássicas e modernas / Ricardo José Gabrielli Barreto
Campello. --Campinas, SP: [s.n.], 2002.

Orientador: Wagner Caradori do Amaral.
Tese (doutorado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Modelos matemáticos. 2. Identificação de
sistemas. 3. Sistemas de comando e controle.
4. Otimização matemática. 5. Sistemas difusos. 6. Redes
neurais (Computação). 7. Algoritmos genéticos.
8. Volterra, Séries de. I. Amaral, Wagner Caradori do.
II. Universidade Estadual de Campinas. Faculdade de
Engenharia Elétrica e de Computação. III. Título.

*À Adriana
e à memória dos meus avós
José e Alda Gabrielli.*

Agradecimentos

Presto meus sinceros agradecimentos a todos os amigos, colegas, professores e funcionários da FEEC/UNICAMP, que direta ou indiretamente contribuíram para a realização deste trabalho. Uma menção particular é devida aos que seguem:

- Ao Prof. Wagner Caradori do Amaral, pela forte amizade e confiança, bem como pelo incentivo e por todo o apoio profissional e pessoal ao longo de sete anos de orientação.
- Aos professores Fernando J. Von Zuben, Fernando A. C. Gomide, Paulo A. V. Ferreira e Maurício C. de Oliveira, pelas sugestões, críticas e consultorias oportunas.
- Aos amigos Maurício Carvalho, Rodrigo Gonçalves e Luiz Meleiro, pelos incontáveis cafés, raras cervejas (pura falta de tempo...) e companheirismo perseverante em tantos longos dias de trabalho.

Agradeço também:

- Ao Prof. Gérard Favier, pelas importantes contribuições e por sua simpática recepção na França durante minha visita ao laboratório I3S da Universidade de Nice - Sophia Antipolis.
- Aos professores Gustavo H. C. Oliveira, da PUC-PR, e Rubens Maciel Filho, da Faculdade de Eng. Química, por suas colaborações em parte dos desenvolvimentos.
- À FAPESP, CAPES, FUNCAMP e a todo o povo brasileiro pelo suporte financeiro a este programa de doutorado e à divulgação dos seus resultados científicos no Brasil e no exterior.

Finalmente, quero expressar minha profunda gratidão aos meus pais, pela paciência e por minha educação, e à família da Dri, por toda a consideração e carinho dos seus pais, irmãos, tios e primos.

“Uma vez perguntaram a Darwin sobre a preguiça e ele fingiu que procurava um lápis embaixo da mesa”.

Luis Fernando Veríssimo, A Grande Mulher Nua, 1975.

Resumo

Este trabalho enfoca o problema de modelagem de sistemas complexos através da utilização integrada de ferramentas matemáticas e computacionais tanto bem estabelecidas quanto recentemente desenvolvidas na literatura. As principais contribuições concentram-se em problemas ainda em aberto relacionados aos sistemas *fuzzy*, mas uma vez que a maioria desses problemas são compartilhados com algumas arquiteturas de redes neurais artificiais, essas também inserem-se no presente contexto. Consideram-se inicialmente os problemas de estimação paramétrica ótima e extração de conhecimento lingüístico a partir de modelos relacionais *fuzzy*, importantes nos demais desenvolvimentos do trabalho. Esses problemas são formalizados e solucionados conjuntamente através de um paradigma matemático e lingüístico original. Em seguida considera-se o principal problema associado aos sistemas *fuzzy* e neurais, que diz respeito à dimensionalidade desses sistemas, através de uma estratégia de representação denominada na literatura de estratégia hierárquica. Várias questões ainda em aberto a respeito dessa estratégia são abordadas a partir da proposta, desenvolvimento e análise de diferentes arquiteturas de modelos hierárquicos. Considera-se finalmente a questão da identificação de sistemas dinâmicos através de modelos *fuzzy* e neurais com estruturas baseadas em funções ortonormais. A fundamentação matemática dos modelos propostos sustenta-se nos bem conhecidos modelos de Volterra desenvolvidos em bases ortonormais de funções, denominados Wiener-Volterra. Conseqüentemente, esses modelos também inserem-se no escopo do trabalho, que inclui um conjunto de novos resultados relacionados à sua otimização. Finalmente, desenvolve-se uma estratégia de controle preditivo baseada na linearização dos modelos de previsão propostos e aplica-se essa estratégia ao controle de dois processos químicos.

Palavras-chave: Sistemas Não Lineares, Sistemas *Fuzzy*, Redes Neurais Artificiais, Algoritmos Genéticos, Séries de Volterra, Funções de Base Ortonormal, Modelagem Qualitativa, Otimização, Estimação, Aproximação Universal de Funções, Identificação e Controle de Processos.

Abstract

This work is concerned with the problem of modeling complex systems by combining classical and modern mathematical and computational tools. The main contributions focus on open questions related to the fields of fuzzy systems and artificial neural networks. Initially, the problems of the optimal estimation and linguistic interpretation of fuzzy relational models are considered. These problems are formulated and solved using a unified approach based on an original mathematical and linguistic paradigm. Next, the “curse of dimensionality” problem of fuzzy and neural systems is dealt with by adopting a modeling strategy based on the use of hierarchical model structures previously introduced in the literature. Two novel hierarchical neural fuzzy models are proposed to avoid the drawbacks associated with the existing structures. Another subject of interest is the identification of dynamic systems using fuzzy and neural models within the framework of orthonormal basis functions. The development of these models rely on mathematical foundations based on the well-known Wiener-Volterra approach. Therefore, Wiener-Volterra models are also investigated and a set of new results with respect to their optimization is presented. At last, a predictive control scheme based on the linearization of the proposed models is developed and applied to the control of two chemical processes.

Keywords: Nonlinear Systems, Fuzzy Logic, Artificial Neural Networks, Genetic Algorithms, Volterra Series, Orthonormal Basis Functions, Qualitative Modeling, Optimization, Estimation, Universal Approximation, System Identification and Control.

Lista de Siglas e Abreviações

- **ART:** Açúcares Redutores Totais.
- **ANOVA:** *ANalysis Of VAriance.*
- **BIBO:** *Bounded-Input/Bounded Output.*
- **BLUE:** *Best Linear Unbiased Estimation.*
- **CARIMA:** *Controlled Auto-Regressive Integrating Moving Average.*
- **CO₂:** Dióxido de Carbono.
- **CSTR:** *Continuous Stirred Tank Reactor.*
- **EQM:** Erro Quadrático Médio.
- **E/S:** Entrada/Saída.
- **FIR:** *Finite Impulse Response.*
- **FCM:** *Fuzzy C-Means.*
- **FBO:** Funções de Base Ortonormal.
- **GPC:** *Generalized Predictive Control(ler).*
- **GA:** *Genetic Algorithm.*
- **IRC:** *Intersection Rule Configuration.*
- **LMS:** *Least Mean Squares.*

- **MISO:** *Multi-Input/Single-Output.*
- **MIMO:** *Multi-Input/Multi-Output.*
- **MLP:** *Multi-Layer Perceptron.*
- **MPC ou MBPC:** *Model-Based Predictive Control(ler).*
- **nD:** *n-Dimensional.*
- **(N)ARX:** *(Nonlinear) Auto-Regressive with eXogenous inputs.*
- **NFIR ou NLFIR:** *NonLinear Finite Impulse Response.*
- **NXOR:** *Negative eXclusive OR.*
- **NLMA:** *NonLinear Moving Average.*
- **NAMW:** *Number Average Molecular Weight.*
- **Norma-t:** Norma triangular.
- **QP:** *Quadratic Programming.*
- **RBF:** *Radial Basis Function.*
- **RNA:** Rede Neural Artificial.
- **RLS:** *Recursive Least Squares.*
- **SISO:** *Single-Input/Single-Output.*
- **SOM:** *Self-Organized Map.*
- **SQP:** *Sequential Quadratic Programming.*
- **TS:** Takagi-Sugeno.
- **UoD:** *Universe of Discourse.*
- **URC:** *Union Rule Configuration.*
- **XOR:** *eXclusive OR.*

Lista de Figuras

3.1	Exemplos de partições <i>fuzzy</i> (7 conjuntos referenciais) que satisfazem o critério de equivalência de informações.	29
3.2	Conjuntos referenciais <i>fuzzy</i> das variáveis de entrada e saída.	36
3.3	Dados estocásticos XOR (símbolos) e valor de contorno 10% das funções de densidade de probabilidade originais. (círculos).	45
3.4	Conjuntos <i>Fuzzy</i> Gaussianos: <i>Muito Pequeno</i> , <i>Pequeno</i> , <i>Médio</i> , <i>Grande</i> , <i>Muito Grande</i> (y) ; <i>Negativo Grande</i> , <i>Negativo Pequeno</i> , <i>Zero</i> , <i>Positivo Pequeno</i> e <i>Positivo Grande</i> (w).	46
3.5	Saídas do sistema (linha cheia) e do modelo relacional (linha tracejada): previsão 1 passo-à-frente (acima) e série sintética (abaixo).	47
3.6	Regras <i>fuzzy</i> e trajetórias lingüísticas: Modelo completo.	48
3.7	Saída do sistema (linha cheia) e do modelo lingüístico extraído (linha tracejada): EQM= 0.4325.	49
3.8	Conjunto de regras simplificado ($\xi = 0.05$): Modelo incompleto.	49
4.1	Conjuntos referenciais de um modelo <i>fuzzy</i> com 2 entradas: $\mathcal{X}_1(x_1)$, $i = 1, 2, 3$ (acima) ; $\mathcal{X}_2(x_2)$ $i = 1, 2$ (abaixo).	57
4.2	Funções de ativação (6 neurônios) da rede neural RBF equivalente.	58
4.3	Estrutura não hierárquica.	59
4.4	Relação entre o número de entradas n e o número de regras <i>fuzzy</i> /neurônios M em modelos não hierárquicos (valores típicos de c).	59
4.5	Estrutura Hierárquica.	60
4.6	Relação entre o número de entradas n e o número de regras <i>fuzzy</i> /neurônios M em modelos hierárquicos (valores típicos de c).	61
4.7	Modelo com normalização de E/S.	64
4.8	Distribuição homogênea de conjuntos referenciais <i>fuzzy</i>	65
4.9	Modelo hierárquico com 3 entradas.	66
4.10	Evolução do EQM para dados de estimação (linha cheia) e validação (linha tracejada) durante treinamento de modelos hierárquicos com hierarquia $x_1(k) = u(k - 1)$, $x_2(k) = y(k - 1)$ e $x_3(k) = y(k - 2)$	68
4.11	Saída do sistema (linha cheia) e previsão 1 passo-à-frente de 2 modelos hierárquicos (linha pontilhada): Melhor modelo (acima) ; pior modelo (abaixo).	68

4.12 Saída do sistema (linha cheia) e série sintética de 2 modelos hierárquicos (linha pontilhada): Melhor modelo (acima) ; pior modelo (abaixo). . . .	69
4.13 Evolução do EQM para dados de estimação (linha cheia) e validação (linha tracejada) durante treinamento de modelos hierárquicos com hierarquia $x_1(k) = y(k - 1)$, $x_2(k) = y(k - 2)$ e $x_3(k) = u(k - 1)$	70
4.14 Função Sinc em um domínio compacto do \mathfrak{R}^2	71
4.15 Modelo hierárquico com domínio bidimensional.	71
4.16 Evolução do EQM entre as saídas do mapeamento Sinc e do seu modelo hierárquico para dados de estimação (linha cheia) e validação (linha tracejada) ao longo do procedimento de treinamento desse modelo. . . .	72
4.17 Modelo hierárquico da função Sinc em um domínio compacto do \mathfrak{R}^2 . . .	72
4.18 Superfície de erro de aproximação da função Sinc: $E(x_1, x_2) = f(x_1, x_2) - \hat{y}(x_1, x_2)$	73
4.19 Estrutura hierárquica alternativa.	75
4.20 Estrutura hierárquica alternativa com 4 entradas.	76
4.21 (a) Evolução do EQM ao longo do treinamento; (b) Saída do modelo (linha tracejada) e saídas artificiais (linha cheia) associadas às classes <i>sestosa</i> (-1), <i>versicolor</i> (0) e <i>virginica</i> (+1): Ordem hierárquica B. . . .	78
4.22 (a) Evolução do EQM ao longo do treinamento; (b) Saída do modelo (linha tracejada) e saídas artificiais (linha cheia) associadas às classes <i>sestosa</i> (-1), <i>versicolor</i> (0) e <i>virginica</i> (+1): Ordem hierárquica A. . . .	79
4.23 Evolução do EQM ao longo do treinamento com relação aos dados de estimação (linha cheia) e validação (linha tracejada): Ordem hierárquica A.	80
4.24 Modelo relacional hierárquico com 4 entradas.	81
4.25 Lógica XOR em cascata.	90
4.26 Lógica NXOR em cascata.	90
4.27 Evolução do erro quadrático médio entre a saída real e a saída do modelo para duas inicializações de parâmetros distintas (A e B).	91
4.28 Projeções dos três <i>clusters</i> no componente x_1 do domínio 4D e respectivas aproximações triangulares: Conjuntos <i>fuzzy</i> \mathcal{A}_1 , \mathcal{B}_1 e \mathcal{C}_1 para a entrada x_1 (de cima para baixo).	96
4.29 Projeções dos três <i>clusters</i> no componente x_2 do domínio 4D e respectivas aproximações triangulares: Conjuntos <i>fuzzy</i> \mathcal{C}_2 , \mathcal{A}_2 e \mathcal{B}_2 para a entrada x_2 (de cima para baixo).	97
4.30 Projeções dos três <i>clusters</i> no componente x_3 do domínio 4D e respectivas aproximações triangulares: Conjuntos <i>fuzzy</i> \mathcal{A}_3 , \mathcal{B}_3 e \mathcal{C}_3 para a entrada x_3 (de cima para baixo).	97
4.31 Projeções dos três <i>clusters</i> no componente x_4 do domínio 4D e respectivas aproximações triangulares: Conjuntos <i>fuzzy</i> \mathcal{A}_4 , \mathcal{B}_4 e \mathcal{C}_4 para a entrada x_4 (de cima para baixo).	98

4.32	Representação simplificada de classes bem comportadas em um domínio 2D.	99
4.33	(a) Evolução do EQM ao longo do treinamento; (b) Saída do modelo (linha tracejada) e saídas artificiais (linha cheia) associadas às classes <i>sestosa</i> (-1), <i>versicolor</i> (0) e <i>virginica</i> (+1).	100
4.34	Representação simplificada de classes não convexas e não linearmente separáveis em um domínio 2D.	103
5.1	Diagrama de blocos do modelo NFIR.	117
5.2	Diagrama de blocos do modelo FBO.	118
5.3	Modelo FBO com dinâmica de Laguerre.	120
5.4	Figura esquemática do tanque de líquido em escala de laboratório.	132
5.5	Dados de estimação (esquerda) e validação (direita) do tanque de líquido.	132
5.6	Saída do processo real (linha cheia) e previsão em série sintética do modelo FBO <i>fuzzy</i> (linha tracejada): Dados de validação.	133
5.7	Concentração de etanol na saída do último reator (P_4 [kg/m ³], linha contínua) e saída do modelo em série sintética (linha pontilhada) para a parcela de validação do conjunto de dados I: (a) Modelo linear, $n = 2$; (b) Modelo de Volterra, $n = 4$; (c) Modelo <i>fuzzy</i> relacional/RBF, $n = 4$; (d) Modelo <i>fuzzy</i> TS, $n = 3$	137
5.8	Medidas da concentração de etanol P_4 [kg/m ³]: Parcela de estimação do conjunto de dados II.	139
5.9	Concentração de etanol P_4 [kg/m ³] (linha contínua) e saída do modelo em série sintética (linha pontilhada) para a parcela de validação do conjunto de dados II: (a) Modelo de Volterra, $n = 4$; (b) Modelo <i>fuzzy</i> relacional/RBF, $n = 3$; (c) Modelo <i>fuzzy</i> Takagi-Sugeno, $n = 2$	140
5.10	Modelo FBO com múltiplas entradas.	142
5.11	Arquitetura A de modelo FBO hierárquico com duas entradas e dois estados ortonormais cada.	143
5.12	Arquitetura B de modelo FBO hierárquico com duas entradas e dois estados ortonormais cada.	144
5.13	Arquitetura alternativa de modelo hierárquico FBO com múltiplas entradas.	144
5.14	Evolução do erro quadrático médio entre as saídas do processo e do modelo ao longo do procedimento híbrido de otimização: Dados de estimação (linha contínua) e validação (linha tracejada).	147
5.15	Saída do processo (P_4 [kg/m ³], linha contínua) e série sintética do modelo (linha tracejada) para dados de validação.	147
5.16	Pólo de Laguerre ótimo para o kernel de 2a. ordem como uma função de ρ_j para alguns valores de ρ_i ($i, j = 1, 2$; $i \neq j$).	155
5.17	Kernel de 2a. ordem com $\rho_1 = 0.5$ e $\rho_2 = 0.01$	156
5.18	Kernel de 2a. ordem com $\rho_1 = 0.5$ e $\rho_2 = 0.9$	157

5.19	Comportamento do erro absoluto máximo entre o kernel original h_2 e o kernel aproximado \bar{h}_2 em função do pólo de Laguerre. A linha pontilhada indica a solução ótima $p_2^* = 0.51349$	157
5.20	Kernel aproximado com $p_2 = 0.51349$ e $N_2 = 3$	158
5.21	Superfície de erro entre os kernels h_2 e \bar{h}_2	158
5.22	Evolução do pólo de Laguerre ao longo do algoritmo iterativo para três conjuntos de dados distintos. As linhas contínuas indicam a solução ótima $p^* = 0.79376$	162
5.23	Evolução do erro quadrático médio entre os dados de validação e a série sintética do modelo para cada iteração do algoritmo: Base de Laguerre única (EQM final = 8.4294×10^{-4}).	162
5.24	Acima: Dados de validação e série sintética do modelo final com base de Laguerre única (curvas sobrepostas); Abaixo: Erro de previsão.	163
5.25	Evolução dos pólos p_1 (círculos) e p_2 (triângulos) de Laguerre ao longo do algoritmo. As linhas contínuas indicam as soluções ótimas $p_1^* = 0.81873$ e $p_2^* = 0.51349$	163
5.26	Evolução do erro quadrático médio entre os dados de validação e a série sintética do modelo para cada iteração do algoritmo: Bases de Laguerre distintas (EQM final = 2.3826×10^{-5}).	164
5.27	Acima: Dados de validação e série sintética do modelo final com bases de Laguerre distintas (curvas sobrepostas); Abaixo: Erro de previsão.	164
6.1	Saída do sistema (linha cheia) e série sintética do modelo FBO-TS com $n = 2$ e $p = 0.75$ (linha tracejada) [kg/kmol]: Dados de validação.	176
6.2	Referência e saída do sistema em malha fechada [kg/kmol] (acima); Variável manipulada [m ³ /h] (abaixo).	177
6.3	Acima: Referência e saída do processo em malha fechada (P_4 [kg/m ³]); Abaixo: Variável manipulada (F_a [m ³ /h]).	178
6.4	Saída do processo (P_4 [kg/m ³], linha contínua) e série sintética do modelo (linha tracejada).	178
6.5	Acima: Referência e saída do processo em malha fechada (P_4 [kg/m ³]); Abaixo: Variável manipulada (F_a [m ³ /h]).	179
6.6	Acima: Referência e saída do processo em malha fechada (P_4 [kg/m ³]); Abaixo: Variável manipulada (F_a [m ³ /h]).	180
6.7	Acima: Referência e saída do processo em malha fechada (P_4 [kg/m ³]); Ao centro: Entrada de perturbação (S_0 [kg/m ³]); Abaixo: Variável manipulada (F_a [m ³ /h]).	180
6.8	Acima: Referência e saída do processo em malha fechada (P_4 [kg/m ³]); Ao centro: Entrada de perturbação (S_0 [kg/m ³]); Abaixo: Variável manipulada (F_a [m ³ /h]).	181
6.9	Acima: Referência e saída do processo em malha fechada (P_4 [kg/m ³]); Abaixo: Variável manipulada (F_a [m ³ /h]).	182

A.1	Discretização <i>fuzzy</i> de um valor <i>crisp</i> a' utilizando dois conjuntos referenciais $\mathcal{A}_1(a)$ e $\mathcal{A}_2(a)$	197
B.1	Figura esquemática da planta industrial para produção de etanol.	200

Lista de Tabelas

3.1	Padrões lógicos XOR.	42
3.2	Padrões lógicos XOR (fuzzificados).	42
3.3	Estimativas do modelo para os padrões lógicos XOR.	43
3.4	Padrões lógicos XOR com perturbação estocástica.	44
4.1	Desempenhos de treinamento e simulação de um modelo hierárquico otimizado a partir de cinco inicializações distintas. Hierarquia: $x_1(k) = u(k - 1)$, $x_2(k) = y(k - 1)$ e $x_3(k) = y(k - 2)$	67
4.2	Desempenhos de treinamento e simulação de um modelo hierárquico otimizado a partir de cinco inicializações distintas. Hierarquia: $x_1(k) = y(k - 1)$, $x_2(k) = y(k - 2)$ e $x_3(k) = u(k - 1)$	69
4.3	Evolução do desempenho de classificação do modelo hierárquico a cada 10 épocas de treinamento.	80
4.4	Tabela de Dados (Padrões Lógicos).	90
4.5	Regras lingüísticas referenciais para o problema <i>Iris</i>	98
4.6	Regras lingüísticas extraídas dos dados <i>Iris</i> (2 ^a e 3 ^a maiores representabilidades).	102
4.7	Regras lingüísticas extraídas dos dados <i>Iris</i> (4 ^a , 5 ^a e 6 ^a maiores representabilidades).	102
5.1	Desempenho de simulação do modelo linear para a parcela de validação do conjunto de dados I.	136
5.2	Desempenho de simulação do modelo de Volterra para a parcela de validação do conjunto de dados I.	138
5.3	Desempenho de simulação do modelo <i>fuzzy</i> relacional/RBF para a parcela de validação do conjunto de dados I.	138
5.4	Desempenho de simulação do modelo <i>fuzzy</i> TS para a parcela de validação do conjunto de dados I.	139
5.5	Desempenhos de simulação dos melhores modelos FBO não lineares para a parcela de validação do conjunto de dados II.	141
6.1	Desempenho de modelagem em função do pólo de Laguerre p ($n = 2$).	175
6.2	Desempenho de modelagem em função do número de funções e estados de Laguerre n ($p = 0.75$).	176

Conteúdo

1	Introdução	1
1.1	Organização e Contribuições	4
2	Modelos <i>Fuzzy</i>, Neurais e de Volterra	7
2.1	Modelos <i>Fuzzy</i>	7
2.2	Modelos Neurais	10
2.3	Modelos Neuro- <i>Fuzzy</i>	12
2.4	Modelos de Volterra	13
2.4.1	Modelos Wiener-Volterra	16
2.5	Resumo	21
2.6	Contribuições	22
3	Estimação e Interpretação de Modelos Relacionais <i>Fuzzy</i>	23
3.1	Introdução	23
3.2	Modelos Relacionais <i>Fuzzy</i>	25
3.3	Estimação Ótima de Modelos Relacionais	26
3.3.1	Seleção de Estrutura	27
3.3.2	Problema de Otimização	28
3.3.3	Aspectos Numéricos	31
3.3.4	Procedimento Algorítmico	33
3.4	Interpretação de Modelos Relacionais	34
3.4.1	Caracterização do Problema	34
3.4.2	Extração de Regras Lingüísticas	35
3.4.3	Simplificação de Bases de Regras	40
3.5	Exemplos	41
3.5.1	Exemplo 1: Problema XOR	42
3.5.2	Exemplo 2: Problema XOR com Perturbação Estocástica	44
3.5.3	Exemplo 3: Fornalha de Gás Box-Jenkins	45
3.6	Resumo e Discussões Complementares	50
3.7	Contribuições e Perspectivas	52

4	Modelos Hierárquicos	53
4.1	Introdução	53
4.2	Modelos Hierárquicos com Submodelos Relacionais Simplificados	56
4.2.1	Modelos Relacionais Simplificados e Redes Neurais RBF	56
4.2.2	Estrutura Hierárquica Padrão	58
4.2.3	Problema de Otimização	61
4.2.4	Normalização de Dados e Inicialização de Parâmetros	64
4.2.5	Seleção da Ordem Hierárquica	65
4.2.6	Estruturas Hierárquicas Alternativas	74
4.2.7	Abordagens Algorítmicas Alternativas	75
4.2.8	Exemplo: Problema de Classificação <i>Iris</i>	77
4.3	Modelos Hierárquicos com Submodelos Relacionais Convencionais	81
4.3.1	Estrutura Hierárquica Padrão	81
4.3.2	Interpretação Lingüística	82
4.3.3	Otimização	87
4.3.4	Normalização de Dados e Inicialização de Parâmetros	95
4.3.5	Exemplo: Problema de Classificação <i>Iris</i>	95
4.4	Aproximação Universal	104
4.5	Resumo e Discussões Complementares	112
4.6	Contribuições e Perspectivas	113
5	Modelos Baseados em Funções de Base Ortonormal (FBO)	115
5.1	Introdução	115
5.2	Modelos FBO	116
5.3	Classes de Modelos FBO	120
5.3.1	Modelo Linear	120
5.3.2	Modelo de Volterra	121
5.3.3	Modelo Relacional <i>Fuzzy</i> / Rede RBF	121
5.3.4	Modelo <i>Fuzzy</i> Takagi-Sugeno	122
5.4	Estimação de Parâmetros	123
5.4.1	Análise de Sistemas com Ruídos de Medida	124
5.5	Projeto da Base de Funções Ortonormais	125
5.6	Interpretação dos Modelos FBO	126
5.6.1	Modelos FBO-TS Generalizados	127
5.7	Capacidade de Representação dos Modelos FBO	129
5.8	Modelos FBO: <i>Fuzzy</i> \times Volterra	130
5.8.1	Caso de Estudo: Tanque de Nível Líquido	131
5.9	Aplicação: Planta para Produção de Etanol	134
5.9.1	Geração e Amostragem de Dados	135
5.9.2	Seleção de Estrutura	135
5.9.3	Estimação dos Modelos FBO	136
5.10	Modelos FBO Multivariáveis	141

5.11	Modelos FBO Hierárquicos	142
5.11.1	Aplicação: Planta para Produção de Etanol	145
5.12	Redução de Ordem em Modelos Wiener-Volterra pela Otimização da Base de Laguerre	148
5.12.1	Definição do Problema	148
5.12.2	Desenvolvimento Ótimo do Kernel de Primeira Ordem	149
5.12.3	Desenvolvimento Ótimo do Kernel de m -ésima Ordem	150
5.12.4	Exemplo	154
5.12.5	Desenvolvimento Ótimo dos Kernels com Base de Laguerre Única	158
5.12.6	Estimação Simultânea dos Kernels e dos Pólos de Laguerre Ótimos	160
5.12.7	Resultados de Simulação	161
5.13	Resumo e Discussões Complementares	164
5.14	Contribuições e Perspectivas	166
6	Aplicações em Controle de Processos Não Lineares	169
6.1	Introdução	169
6.2	Controle Preditivo com Base na Linearização de Modelos	170
6.3	Linearização de Modelos <i>Fuzzy</i> FBO	171
6.4	Controle Preditivo de um Processo de Polimerização	173
6.4.1	Identificação do Reator CSTR	174
6.4.2	Controle do Reator CSTR	176
6.5	Controle Preditivo de um Processo Biotecnológico	177
6.6	Resumo e Discussões Complementares	183
6.7	Contribuições e Perspectivas	183
7	Conclusões e Perspectivas	185
7.1	Conclusões	185
7.2	Perspectivas	187
A	Fundamentos em Conjuntos e Sistemas <i>Fuzzy</i>	189
A.1	Modelos Baseados em Regras <i>Fuzzy</i>	189
A.1.1	Modelos Lingüísticos (de Mamdani)	189
A.1.2	Modelos Takagi-Sugeno	193
A.2	Equivalência entre Modelos <i>Fuzzy</i> e Redes Neurais RBF	194
A.3	Discretização <i>Fuzzy</i>	195
A.4	Estrutura de Cognição	197
B	Processo Industrial para Produção de Etanol	199
B.1	Introdução	199
B.2	Planta para Produção de Etanol	200
B.2.1	Variáveis Manipuladas, de Perturbação e de Saída	202

C	Algoritmo Genético	205
C.1	Introdução	205
C.2	Algoritmo	206
D	Provas de Lemas e Teoremas	209
D.1	Prova do Teorema 5.1	209
D.2	Prova do Lema 5.1	216
D.3	Prova do Lema 5.2	216
D.4	Prova do Teorema 5.3	220
E	Controlador Preditivo Generalizado (GPC)	223
E.1	Introdução	223
E.2	O Controlador GPC	223
E.2.1	Previsão via Cálculo Iterativo do Modelo	225
E.2.2	Otimização do Critério de Desempenho	230
E.2.3	Restrições de Entrada e Saída	231
E.2.4	Modelos FBO	231
E.2.5	Propriedades do Controle Preditivo	232
	Bibliografia	233

Capítulo 1

Introdução

Nas últimas duas décadas houve um rápido crescimento no interesse em pesquisas e aplicações das técnicas de *Inteligência Computacional*, basicamente classificadas nas teorias de Conjuntos e Sistemas Nebulosos (*Fuzzy*), Redes Neurais Artificiais (RNAs) e Computação Evolutiva (Algoritmos Genéticos, entre outros). O termo Inteligência Computacional foi motivado para distinguir essas teorias daquela denominada *Inteligência Artificial* clássica, largamente investigada nos anos 70 e que se caracteriza essencialmente por reproduzir ou representar conhecimentos através de conjuntos de regras passíveis de serem executadas em um computador, como por exemplo os bem conhecidos sistemas especialistas.

As técnicas de inteligência computacional têm sido utilizadas em diferentes campos do conhecimento e em diversas aplicações, tais como aproximação de funções, classificação de padrões, previsão de séries temporais, filtragem de sinais e identificação/controlar de processos dinâmicos, particularmente quando métodos convencionais não proporcionam resultados satisfatórios. Sabe-se que cada um desses problemas pode ser resumido à modelagem de um mapeamento que represente de forma adequada as relações relevantes entre as variáveis de interesse do problema em questão. Essas relações, de uma maneira geral não lineares e multivariáveis, possivelmente de grande porte e variantes no tempo, são denominadas *Sistemas Complexos* no presente contexto.

Neste trabalho aborda-se o problema da modelagem de sistemas complexos através da utilização integrada de ferramentas matemáticas e computacionais clássicas e modernas. Por ferramentas clássicas entende-se aquelas bem estabelecidas até os anos 70, tais como estimadores de mínimos quadrados, métodos de otimização não linear, análise convexa, bases de funções ortonormais e séries de Volterra, enquanto ferramentas modernas referem-se àquelas relacionadas às técnicas de inteligência computacional discutidas acima.

Particular ênfase é dada às aplicações na área de identificação e controle de processos dinâmicos. O trabalho também enfoca de maneira especial os sistemas *fuzzy*, por duas razões principais. A primeira é que esses sistemas conjugam a capacidade de processar informação de natureza incerta ou qualitativa com a capacidade de aproxi-

mação universal e de aprendizagem características das RNAs *feedforward*. A segunda são as diferentes arquiteturas existentes, adequadas a diferentes tipos de aplicação, indo dos modelos lingüísticos e relacionais, com particular habilidade para a incorporação e geração de conhecimento lingüístico na modelagem de um determinado sistema, aos modelos Takagi-Sugeno, com estrutura intrinsecamente adequada para aplicações em controle. As principais contribuições da tese concentram-se em problemas ainda em aberto na literatura relacionados a esses modelos. Uma vez que a maioria desses problemas são compartilhados com as arquiteturas *feedforward* de RNAs, essas também inserem-se no escopo do trabalho. Nesse contexto, especial ênfase é dada às redes RBF (*Radial Basis Functions*), principalmente devido à equivalência existente entre essa e algumas arquiteturas de modelos *fuzzy* que serão consideradas.

Um dos temas centrais abordados no trabalho é o problema do crescimento exponencial na quantidade de regras lingüísticas (ou neurônios) em modelos *fuzzy* (ou neurais), necessárias para representar um sistema com determinada precisão, em função do aumento na dimensão do seu espaço de entrada. Esse problema, que é um dos principais obstáculos à aplicação desses modelos a sistemas de grande porte, é conhecido como “Maldição da Dimensionalidade” (*Curse of Dimensionality*) e será abordado no presente trabalho através de uma estratégia denominada na literatura de *Estrutura Hierárquica*. Nessa estratégia, o espaço de entrada n -dimensional do problema de modelagem é particionado em subespaços de menor dimensão representados por um conjunto de submodelos dispostos em cascata, o que proporciona um crescimento linear (não mais exponencial) na quantidade de regras/neurônios com a dimensão de entrada n . Algumas questões ainda permanecem em aberto na literatura a respeito dessa estratégia. Uma questão fundamental referia-se à interpretação dos modelos, ou seja, como extrair conhecimento lingüístico na forma de um conjunto de regras *fuzzy* a partir de um dado modelo hierárquico, projetado preferencialmente com capacidade de aproximação universal. Essas e outras questões também relevantes, tais como a redução da susceptibilidade da otimização (não convexa) dos modelos a mínimos locais críticos bem como a escolha da ordem (disposição) hierárquica das suas variáveis de entrada, são tratadas nesse trabalho a partir da proposta, desenvolvimento e análise de duas arquiteturas hierárquicas distintas; uma é constituída de submodelos relacionais *fuzzy* com *estrutura simplificada* (equivalentes às redes RBF sob certas condições), enquanto a outra é formada por submodelos relacionais convencionais. As vantagens e desvantagens de cada uma dessas arquiteturas são avaliadas ao longo do trabalho.

A questão da interpretação de modelos hierárquicos é abordada no presente contexto de tal forma que recai no problema de interpretação lingüística de modelos relacionais não hierárquicos. Esse problema tem sido tradicionalmente considerado na literatura de forma heurística, e uma fundamentação matemática consistente era desejada. Essa fundamentação constitui outro tema de destaque do trabalho, que trata a questão da interpretação dos modelos relacionais *fuzzy* (fundamentais no restante dos desenvolvimentos) através de um paradigma original baseado nos novos conceitos de *contribuição* e *representabilidade* de elementos relacionais. Também desenvolve-se uma estratégia

de projeto para esses modelos que permite a estimação ótima dos seus parâmetros relacionais através de algoritmos de mínimos quadrados e programação quadrática. O principal resultado é a estimação ótima global desses parâmetros sem qualquer alteração na estrutura clássica dos modelos, o que permite a extração de regras *fuzzy* a partir dos modelos estimados.

Outro tema importante abordado no trabalho refere-se à questão da identificação de sistemas dinâmicos através de modelos *fuzzy* e neurais, objetivando a aplicação desses modelos ao projeto de controladores automáticos. Alguns problemas estão associados às abordagens usuais, que utilizam regressores dos sinais de entrada e saída do sistema dinâmico como entradas dos modelos. O principal deles é a realimentação dos erros de saída, que usualmente degrada a qualidade da previsão de horizonte longo e dá margem à obtenção de modelos instáveis mesmo na modelagem de sistemas estáveis. Outro problema é a dimensão de entrada dos modelos, que é particularmente elevada para sistemas com ordens elevadas ou mesmo para sistemas com respostas lentas, dependendo da configuração dinâmica adotada. Um terceiro problema é a necessidade da estimação de regressores adequados, isto é, os instantes passados relevantes dos sinais do sistema. Esses problemas são abordados conjuntamente nesse trabalho através de uma estratégia de modelagem baseada em séries de funções ortonormais, como as funções de Laguerre. Nessa estratégia os modelos possuem uma estrutura do tipo Wiener, isto é, possuem uma dinâmica linear, constituída por um conjunto de filtros de funções ortonormais, seguida de um mapeamento estático não linear. Esses modelos são denominados aqui de *Modelos FBO* (Modelos com Funções de Base Ortonormal). O uso de redes neurais com essa estrutura foi proposto anteriormente na literatura no contexto de redes MLP (*Multi-Layer Perceptron*). Nesse trabalho esse conceito é estendido para as redes RBF, a partir de um desenvolvimento matemático original, e generalizado para os modelos *fuzzy* dos tipos relacional e Takagi-Sugeno. Os modelos resultantes são então escritos através de uma formulação em espaço-de-estados que generaliza a representação desses e de outros modelos FBO conhecidos. Essa formulação permite a interpretação matemática clara dos diversos modelos considerados e a análise das diferenças existentes entre eles. Além dos modelos discutidos acima, os modelos *fuzzy* hierárquicos também são desenvolvidos segundo a estrutura FBO, com enfoque especial dado ao caso de sistemas dinâmicos multivariáveis. Nesse caso, diferentes alternativas hierárquicas são propostas e analisadas. Desenvolve-se ainda uma estratégia de controle preditivo baseada na implementação da lei de controle do bem conhecido Controlador Preditivo Generalizado (GPC) associado a modelos de previsão *fuzzy* FBO linearizados. Essa estratégia é aplicada ao controle de uma planta biotecnológica para produção de etanol em escala industrial que, junto a outros casos de estudo, é também utilizada ao longo do trabalho para a avaliação do desempenho de vários dos modelos e algoritmos propostos.

A fundamentação matemática dos modelos FBO propostos sustenta-se nos bem conhecidos modelos de Volterra com estrutura FBO, denominados *Wiener-Volterra*, que também inserem-se no escopo do trabalho. Nesse contexto, desenvolvem-se soluções analíticas globais ótimas para a parametrização das bases de Laguerre utilizadas na

expansão de kernels de Volterra em séries de funções ortonormais. Essas soluções permitem a redução de ordem da representação em espaço-de-estados dos modelos Wiener-Volterra resultantes, simplificando assim os problemas de identificação e controle associados a esses modelos.

1.1 Organização e Contribuições

A organização dos capítulos subseqüentes junto às suas principais contribuições é descrita abaixo:

- **Capítulo 2:** Apresenta uma visão geral a respeito das diferentes classes de modelos não lineares nas quais se sustenta o presente trabalho, que são os modelos *Fuzzy*, Neurais, *Neuro-Fuzzy* e de Volterra/Wiener-Volterra. São discutidos e comparados os aspectos relevantes de cada um desses modelos, com especial ênfase dada aos primeiros e aos últimos, que concentram os principais resultados originais da tese. A principal contribuição do capítulo é uma revisão comparativa original a respeito dos modelos em questão.
- **Capítulo 3:** Aborda os problemas de modelagem quantitativa e qualitativa de sistemas complexos através de modelos relacionais *fuzzy*. Além de um retrospecto histórico a respeito do desenvolvimento desses modelos, apresenta-se também uma abordagem integrada para sua otimização e subseqüente interpretação lingüística, incluindo uma metodologia para simplificação dos modelos resultantes e um conjunto de exemplos acadêmicos e práticos. Principais contribuições:
 1. Compilação bibliográfica detalhada a respeito de modelos relacionais dentro do contexto geral de sistemas e modelagem *fuzzy*.
 2. Introdução dos conceitos de contribuição, representabilidade e representabilidade condicionada de elementos relacionais como um novo paradigma para a interpretação lingüística dos referidos modelos.
 3. Desenvolvimento de uma metodologia sistemática para o projeto estrutural de modelos relacionais que permite uma formulação convexa para o problema de otimização desses modelos e a conseqüente estimação ótima global dos seus parâmetros via algoritmos de mínimos quadrados, mínimos quadrados ponderado, mínimos quadrados recursivo e programação quadrática.
 4. Desenvolvimento de uma metodologia unificada para a representação e extração de conhecimento qualitativo a partir de modelos relacionais bem como para a simplificação dos modelos lingüísticos resultantes.
- **Capítulo 4:** Aborda o problema da dimensionalidade em sistemas *fuzzy* e neurais através de uma estrutura de modelos denominada estrutura hierárquica. Duas

novas classes de modelos hierárquicos são propostas e as características fundamentais de cada uma delas são avaliadas e comparadas. Questões como a otimização, capacidade de aproximação e interpretação lingüística dos modelos propostos, entre outras, são abordadas a partir de fundamentações matemáticas e ilustradas através de resultados experimentais na forma de exemplos práticos e acadêmicos. Principais contribuições:

1. Desenvolvimento de duas novas classes de modelos hierárquicos baseadas na utilização de tipos particulares de modelos relacionais *fuzzy*.
 2. Desenvolvimento de uma metodologia para a extração de conhecimento lingüístico a partir de uma dessas classes de modelos.
 3. Estudo da capacidade de aproximação dos modelos desenvolvidos.
 4. Obtenção de um conjunto de equações genéricas, para modelos com qualquer quantidade de entradas, para a otimização dos parâmetros dos modelos hierárquicos desenvolvidos.
 5. Proposta de diferentes procedimentos algorítmicos para a estimação dos parâmetros dos modelos desenvolvidos.
 6. Proposta para a seleção da ordem hierárquica dos modelos com base em estudos sobre a influência da disposição das suas variáveis de entrada.
 7. Proposta de arquiteturas hierárquicas alternativas.
- **Capítulo 5:** Aborda o problema de identificação de sistemas dinâmicos não lineares através de modelos baseados em Funções de Base Ortonormal (FBO). Tem como tema central o desenvolvimento de modelos *fuzzy* com estrutura FBO e o estudo dos mesmos no que tange a diversos aspectos de interesse, tais como sua formalização e interpretação matemática, capacidade de representação, entre outros. Análises comparativas entre esses modelos e outras classes de modelos FBO são apresentadas tanto no âmbito teórico como em aplicações práticas. Particular ênfase é dada aos modelos FBO do tipo Wiener-Volterra, para os quais desenvolve-se uma metodologia de projeto da base de funções ortonormais segundo a solução de um problema de otimização que visa a redução da ordem dos modelos resultantes. Principais contribuições:
 1. Proposta e desenvolvimento de diferentes tipos de modelos *fuzzy* com estrutura FBO.
 2. Estudo da capacidade de representação dos modelos *fuzzy* FBO propostos.
 3. Interpretação matemática dos modelos *fuzzy* FBO propostos como realizações específicas de uma representação generalizada dada pela interpolação de diferentes modelos FBO locais.

4. Estudos comparativos entre os modelos *fuzzy* FBO propostos e outras classes de modelos FBO a partir de uma formulação unificada que permite a interpretação das diferenças entre essas diversas classes, diferenças essas avaliadas detalhadamente através de argumentos teóricos e experimentos computacionais.
 5. Proposta e desenvolvimento de diferentes arquiteturas de modelos *fuzzy* FBO hierárquicos.
 6. Estudo de estratégias para o projeto das bases de funções ortonormais utilizadas na implementação computacional de modelos FBO.
 7. Obtenção de uma solução analítica global para o desenvolvimento ótimo de kernels de Volterra em funções ortonormais de Laguerre tanto para modelos com bases de Laguerre independentes para o desenvolvimento de cada kernel como para os modelos usuais, nos quais utiliza-se uma base comum ao desenvolvimento compartilhado de todos os kernels.
 8. Desenvolvimento de um algoritmo iterativo para a estimação simultânea dos kernels de Volterra e da solução ótima para o desenvolvimento desses kernels em funções de Laguerre utilizando dados de E/S de um sistema a ser modelado.
- **Capítulo 6:** Aborda o problema de controle automático de sistemas não lineares através de estratégias de controle baseadas em modelos dos processos a serem controlados. As discussões concentram-se nos modelos *fuzzy* com estrutura FBO, que são particularmente favoráveis a aplicações em modelagem de sistemas dinâmicos. Propõe-se uma estratégia de controle preditivo baseada na linearização de modelos *fuzzy* FBO a cada instante de amostragem e aplica-se essa estratégia ao controle de dois processos químicos: um reator de polimerização e uma planta biotecnológica para produção de etanol. Principais contribuições:
 1. Desenvolvimento de uma estratégia de controle preditivo baseada na utilização do Controlador Preditivo Generalizado (GPC) associado a um procedimento de linearização instantânea de modelos *fuzzy* com estrutura FBO.
 2. Aplicação da estratégia proposta ao controle de dois processos químicos em ambiente de simulação.
 - **Capítulo 7:** Apresenta as conclusões e perspectivas para trabalhos futuros.

Capítulo 2

Modelos *Fuzzy*, Neurais e de Volterra

Esse capítulo apresenta as diferentes classes de modelos não lineares consideradas no presente trabalho, que são os modelos *Fuzzy*, Neurais, Neuro-*Fuzzy* e de Volterra/Wiener-Volterra. São discutidos e comparados os aspectos relevantes de cada um desses modelos, com especial ênfase dada aos primeiros e aos últimos, nos quais se concentram as principais contribuições do trabalho.

2.1 Modelos *Fuzzy*

Modelos matemáticos determinísticos ou estocásticos convencionais podem não proporcionar um desempenho satisfatório na modelagem de sistemas complexos. Quando esses modelos não conseguem representar adequadamente um determinado sistema, a busca por modelos alternativos torna-se necessária. Uma classe de modelos que tem ganhado importância crescente nesse contexto são os *Modelos Fuzzy* (Yager e Filev, 1994; Pedrycz, 1996b; Hellendoorn e Driankov, 1997), devido a duas razões principais. A primeira é a habilidade desses modelos para tratar tanto informações de natureza quantitativa quanto qualitativa, permitindo o seu desenvolvimento através de informações fornecidas por especialistas, tais como operadores de processos, bem como a extração de conhecimento lingüístico a partir de dados desses processos. A segunda razão é a capacidade de aproximação universal desses modelos, isto é, a capacidade de aproximar com precisão arbitrária qualquer mapeamento não linear contínuo definido sobre uma região compacta (fechada e limitada) do domínio (Wang e Mendel, 1992; Zeng e Singh, 1994; Zeng e Singh, 1995; Oliveira e Lemos, 1997; Kosko, 1997).

A essência do que se refere nos dias de hoje como modelagem *fuzzy* advém das idéias originais de Zadeh (1973) a respeito da representação de sistemas complexos através de um conjunto de declarações condicionais, denominado algoritmo comportamental *fuzzy*. Esse tipo de algoritmo foi desenvolvido para descrever relações entre variáveis lingüísticas com o propósito específico da representação aproximada do comportamento

de processos de natureza humana.

Seguindo a estrutura básica dos algoritmos comportamentais de Zadeh, Tong (1978) propôs a utilização de um modelo *fuzzy* para a identificação de sistemas a partir de conjuntos de dados de suas entradas e saídas. O tipo de modelo utilizado por Tong é atualmente denominado *Modelo Lingüístico* e é constituído por um conjunto de implicações (regras) *fuzzy* de formato “Se x é A Então y é B ”, onde x e y são variáveis de entrada e saída, respectivamente, e A e B são termos lingüísticos associados a conjuntos *fuzzy* que descrevem lingüisticamente essas variáveis. Para um dado valor de entrada, a saída correspondente é calculada a partir do conjunto de regras através de um método de inferência. Os métodos de inferência (*fuzzy reasoning*) mais conhecidos e adotados são o Min-Max-Gravidade (Mamdani, 1974; Mizumoto, 1991) e o Produto-Soma-Gravidade (Mizumoto, 1991; Mizumoto, 1996). Esses métodos são apresentados no apêndice A a partir de uma formulação unificada e com uma descrição detalhada da estrutura das bases de regras lingüísticas.

Os modelos lingüísticos têm como principal característica serem qualitativamente interpretáveis desde que os respectivos conjuntos de regras sejam semanticamente claros. No entanto, a obtenção de um conjunto adequado de regras a partir de dados de um sistema não é uma tarefa simples (Yager e Filev, 1994), especialmente porque as regras envolvem termos de natureza lingüística. Em seu trabalho pioneiro, Tong (1978) propôs um método *off-line* de “inspeção lógica” associado com heurísticas, estatística e procedimentos de tentativa-e-erro para tratar esse problema. Posteriormente, Graham e Newell (1989) generalizaram a metodologia de Tong através de um algoritmo com capacidade para a geração automática de regras e um gerenciamento heurístico de regras conflitantes. Embora limitadas, essas contribuições expuseram a complexidade do problema e encorajaram a busca por estratégias mais eficientes para abordá-lo (ver (Nozaki et al., 1997; Sudkamp e Hammell II, 1994) e referências inclusas). Uma das abordagens mais conhecidas e eficientes para esse problema é o uso de técnicas de *clustering*, como em (Sugeno e Yasukawa, 1993; Setnes, Babuška e Verbruggen, 1998; Pedrycz e Vasilakos, 1999). Essas técnicas, baseadas em otimização, têm se mostrado capazes de gerar automaticamente conjuntos de regras *fuzzy* que podem descrever precisamente o comportamento de entrada-saída de sistemas complexos. Contudo, os conjuntos *fuzzy* das regras resultantes são usualmente obtidos *a posteriori* pelo algoritmo, e não definidos pelo usuário. Conseqüentemente, os conjuntos e as respectivas regras podem não possuir um significado lingüístico claro (Guillaume, 2001). Mesmo quando restrições de integridade são utilizadas para atenuar esse problema, como em (Espinosa e Vandewalle, 2000), pode-se considerar que uma seqüência mais adequada para a extração de conhecimento lingüístico com base nas idéias originais a respeito de conjuntos e lógica *fuzzy* (Yager et al., 1987) é a definição *a priori* dos termos lingüísticos das variáveis de entrada e saída associados a esses conjuntos, seguida da estimação das relações ou regras existentes entre esses termos, como será discutido no capítulo 3.

Uma abordagem alternativa para a modelagem *fuzzy* foi introduzida por Takagi e Sugeno (1985). O modelo Takagi-Sugeno (TS) possui também uma estrutura baseada

em regras. Contudo, os conseqüentes das regras não são conjuntos *fuzzy* como nos modelos lingüísticos. Esses conseqüentes são formados por funções *crisp* (não *fuzzy*) que mapeam as entradas do modelo em sua saída. Essas funções, também denominadas modelos locais, possuem usualmente uma forma afim em seus argumentos¹. Nesse caso o modelo TS é linear nos parâmetros das referidas funções, que por sua vez podem ser estimados utilizando algoritmos clássicos como os algoritmos de Mínimos Quadrados (*Least Squares*) e filtro de Kalman (Söderström e Stoica, 1989; Ljung, 1999). Uma descrição da estrutura de base de regras do tipo TS, bem como do seu mecanismo de inferência, é apresentada no apêndice A.

Os modelos TS podem ser considerados como uma versão *fuzzy* do método de aproximação linear por partes, que proporciona uma relação linear ou afim de entrada-saída para cada subespaço pré-determinado do espaço de entrada, possuindo a vantagem de serem capazes de combinar as diferentes relações correspondentes a cada uma de suas regras. Essa habilidade de interpolação permite a geração de um mapeamento final mais suave, reduzindo o número de relações individuais e parâmetros de projeto necessários para representar um sistema com precisão arbitrária.

Na última década tem havido um número significativo e crescente de contribuições na literatura a respeito de modelos TS, que incluem metodologias mais eficazes para inferência (Babuška et al., 1996), seleção de estrutura (Jin, 2000), estimação paramétrica (Yen et al., 1998; Johansen et al., 2000) e partição do espaço de entrada (Setnes, 2000; Wong e Chen, 2000; Roubos e Setnes, 2001). Em geral, essas contribuições enfocam aspectos relacionados com a precisão e complexidade desses modelos, especialmente com o propósito de aplicações em controle. De fato, a estrutura dos modelos TS é intrinsecamente adequada para o controle de sistemas dinâmicos (Filev, 1991; Johansen et al., 2000), o que será explorado no capítulo 6. O principal problema com essa abordagem é que os modelos não são interpretáveis no sentido qualitativo, ou seja, não podem proporcionar conhecimento lingüístico devido ao formato funcional (*crisp*) dos conseqüentes das suas regras.

Uma terceira abordagem para a modelagem *fuzzy* é baseada na teoria de equações relacionais (Dubois e Prade, 1980; Pedrycz, 1993). Os modelos correspondentes, denominados *Modelos Relacionais Fuzzy*, consistem de uma representação alternativa dos modelos lingüísticos em que os conjuntos de regras desses últimos são escritos como uma relação *fuzzy* em uma equação relacional (Yager e Filev, 1994) (ver apêndice A). A vantagem dessa representação é que nos modelos lingüísticos tem-se um conjunto de regras que devem ser determinadas, enquanto nos modelos relacionais tem-se apenas uma matriz relacional, que descreve a relação *fuzzy*, a ser estimada. Os modelos relacionais constituem uma alternativa intermediária entre as duas classes de modelos baseados em regras discutidas anteriormente, apresentando características importantes tanto sob aspectos numéricos quanto lingüísticos. Dada a importância desses modelos nos desenvolvimentos do presente trabalho, eles são analisados separadamente, em

¹Embora funções não lineares nos argumentos (lineares ou não nos parâmetros) possam também ser utilizadas (Delgado et al., 2000; Delgado, 2002).

detalhes, no capítulo 3.

As discussões acima envolvem as arquiteturas fundamentais de modelos *fuzzy* que têm sido tipicamente adotadas na literatura desde os desenvolvimentos iniciais dessa classe de modelos. É importante ressaltar que outras abordagens com propósitos específicos têm sido propostas com base em modificações e/ou combinações das arquiteturas fundamentais referidas acima. Exemplos importantes são os *Celibate Fuzzy Models* (Filev e Yager, 1997), desenvolvidos para a modelagem de problemas que requerem soluções exclusivas (não combináveis), tais como diagnósticos e algumas aplicações de tomada de decisão, e os *Fuzzy Multimodels* (Pedrycz, 1996a), desenvolvidos para a modelagem de problemas que admitem múltiplas soluções devido à sua natureza mais relacional do que funcional. Outros exemplos, de particular interesse para o presente trabalho, são os *Modelos Hierárquicos* (Wang, 1998; Campello e Amaral, 1999c; Wang, 1999; Campello e Amaral, 2000b; Chen e Wang, 2000; Meleiro et al., 2001; Campello e Amaral, 2002b; Meleiro et al., nd), desenvolvidos para tratar o bem conhecido problema da dimensionalidade em sistemas *fuzzy*², e os *Modelos com Funções de Base Ortonormal* (FBO) (Oliveira et al., 1999; Campello, Meleiro, Amaral e Maciel Filho, 2001; Campello e Amaral, 2002c; Campello et al., nd), para a modelagem *fuzzy* de sistemas dinâmicos sem realimentação dos erros de previsão e sem a necessidade de estimação dos termos passados relevantes dos seus sinais de entrada e saída. Esses modelos constituem respectivamente os temas dos capítulos 4 e 5.

2.2 Modelos Neurais

O conceito fundamental contido nos sistemas classificados atualmente como Redes Neurais Artificiais (RNAs) foi introduzido através do trabalho de McCulloch e Pitts (1943) que, partindo do princípio de que o cérebro é constituído de unidades básicas, os neurônios, propuseram um modelo matemático para representar essas unidades: o neurônio artificial. A classe denominada RNAs reúne atualmente diversos modelos matemáticos distintos e com domínios de aplicação variados. É o caso por exemplo dos mapas auto-organizáveis (SOM), como as redes de Kohonen (Kohonen, 1982; Kohonen, 1997), das redes recorrentes, como as redes de Hopfield (Hopfield, 1982; Hopfield e Tank, 1985), e das redes *feedforward*, como as redes MLP e RBF (Kosko, 1992; Haykin, 1999). Essas últimas (*feedforward*), assim como os modelos *fuzzy*, destinam-se à aproximação de mapeamentos não lineares genéricos, podendo ser utilizadas em diversas aplicações como modelagem e controle de sistemas dinâmicos, aproximação de funções, classificação de padrões, entre outros, sendo portanto de particular interesse no contexto do presente trabalho.

Utilizando as idéias sobre os neurônios artificiais de McCulloch e Pitts, Rosenblatt (1959) desenvolveu o Perceptron, uma RNA inicialmente com apenas uma camada de

²O conceito de hierarquia em modelos *fuzzy* foi também utilizado por Yager (1998) em um trabalho importante a respeito da representação de prioridades e exceções em bases de regras lingüísticas.

neurônios e posteriormente estendida para múltiplas camadas, denominada então de Perceptron Multi-Camadas ou *Multi-Layer Perceptron* (MLP). O trabalho de Rosenblatt introduziu o conceito de aprendizado supervisionado, em que um dado modelo é treinado para reproduzir (e sempre que possível generalizar) relações entre variáveis a partir de um conjunto de dados, denominados exemplos ou padrões. A popularização das redes MLP se deu a partir do trabalho de Rumelhart et al. (1986), que formalizaram o algoritmo de Retro-propagação (*Back-propagation*, cuja idéia original é atribuída a Werbos (1974)) como um procedimento sistemático para o treinamento dessas redes.

As redes de Funções de Base Radial, ou *Radial Basis Function* (RBF) *networks*, foram propostas por Broomhead e Lowe (1988), diferindo fundamentalmente das MLPs no que diz respeito ao tipo de unidade elementar (neurônio) utilizada. Ao contrário das MLPs, cujos neurônios por motivações biológicas possuem funções de ativação do tipo sigmoidal, as RBFs possuem funções de ativação radiais, como por exemplo as funções Gaussianas e funções tipo sino (*bell-shaped*). Em sua configuração mais comum essas redes possuem apenas uma camada de neurônios de entrada cujas saídas são ponderadas através dos pesos sinápticos de um único neurônio linear para cada saída final da rede. Nesse caso, se as funções de ativação radiais forem pré-definidas³, o modelo torna-se linear em relação aos pesos sinápticos e portanto esses pesos podem ser estimados através de algoritmos clássicos como Mínimos Quadrados (Ljung, 1999), inclusive de forma adaptativa. Dependendo do problema em questão, essa propriedade pode representar uma vantagem significativa em relação às redes MLP treinadas com *back-propagation*, que possuem as dificuldades típicas dos algoritmos de otimização baseados em gradiente, tais como velocidade de convergência e susceptibilidade a soluções locais. Contudo, sabe-se que em problemas de maior porte o método dos mínimos quadrados pode apresentar dificuldades numéricas que demandam atenção e tratamento especial (Wellstead e Zarrop, 1991). Mais além, devido à característica radial das suas funções de ativação, as redes RBF usualmente requerem uma quantidade maior de neurônios para a aproximação de um dado mapeamento não linear com a mesma precisão de uma rede MLP (Haykin, 1999).

Em resumo, pode-se dizer genericamente que, assim como os modelos *fuzzy*, os modelos neurais têm se estabelecido como uma das ferramentas fundamentais para o tratamento de problemas de modelagem complexos para os quais modelos convencionais podem não proporcionar um desempenho satisfatório. Isso deve-se basicamente a três fatores: *i*) A capacidade de aprendizado das RNAs, que tem sido ao longo do tempo uma das matérias de maior interesse nessa área do conhecimento, desde os trabalhos de Hebb (1949) com motivações biológicas (“Postulado de Aprendizagem”), passando pelos trabalhos clássicos de Widrow e Hoff (1960) (algoritmo LMS), Hopfield (1982) (memórias associativas), Rumelhart et al. (1986) (*back-propagation*) e Kohonen (1982) (treinamento não supervisionado), até as modernas abordagens auto-constitutivas (Von Zuben, 1996); *ii*) A capacidade de aproximação universal desses modelos, isto é, assim

³Possivelmente com base na estrutura dos dados a serem aproximados, por exemplo através de técnicas de *clustering* (Bezdek, 1981; Babuška e Verbruggen, 1996).

como os modelos *fuzzy*, os modelos neurais são capazes de aproximar com precisão arbitrária qualquer mapeamento não linear contínuo definido sobre uma região compacta do domínio⁴; *iii*) O tratamento unificado para arquiteturas monovariáveis (SISO), com múltiplas entradas (MISO) ou múltiplas entradas e saídas (MIMO).

2.3 Modelos Neuro-Fuzzy

Das discussões anteriores pode-se notar que as principais características de cada uma das classes de modelos abordadas são a interpretabilidade/transparência (modelos *fuzzy*) e capacidade de aprendizagem (modelos neurais). Através da incorporação mútua dessas características surge uma classe híbrida cujos elementos são denominados modelos *neuro-fuzzy*. É importante ressaltar que essa mesma nomenclatura pode também ser utilizada para se referir a modelos neurais que são equivalentes a modelos *fuzzy* e vice-versa, quando analisados sob ambos os aspectos. É o caso das redes RBF que sob certas condições são equivalentes a modelos TS com conseqüentes constantes e também a modelos lingüísticos com conseqüentes singleton (Hunt et al., 1996; Cho e Wang, 1996). Essa equivalência também pode ser verificada para um tipo de modelo relacional, com estrutura simplificada, conforme mostrado no apêndice A.

Uma das principais abordagens *neuro-fuzzy* tem sido a incorporação de técnicas de aprendizagem neurais para o treinamento de modelos *fuzzy*. Nesse tipo de metodologia híbrida, um conjunto de regras *fuzzy* é em geral parametrizado e os parâmetros são treinados similarmente aos pesos sinápticos de uma RNA (Horikawa et al., 1992; Jang, 1993; Wang, 1994; Lin e Cunningham III, 1995), podendo inclusive incorporar técnicas de inserção e/ou extração automática de regras (neurônios) (Cho e Wang, 1996). Para reduzir o problema de geração de regras inconsistentes ou de difícil interpretação, restrições semânticas também podem ser consideradas (Oliveira, 1999b; Oliveira, 1999a).

Outra abordagem similar é a incorporação de estruturas intrinsecamente *fuzzy* em RNAs, como as arquiteturas neurais desenvolvidas utilizando neurônios que realizam processamentos elementares originários dos métodos de inferência *fuzzy* (Figueiredo et al., 1995; Pedrycz e Gomide, 1998). Nesse tipo de metodologia, ao contrário das abordagens caixa preta convencionais, as redes obtidas permitem algum tipo de interpretação lingüística com base na interpretação das estruturas *fuzzy* utilizadas (e.g. (Pedrycz et al., 1995)).

Nas abordagens descritas acima os valores das variáveis de entrada do modelo são fuzzificados (convertidos para grandezas *fuzzy*) antes de serem processados, isto é, a primeira fase do processamento é a conversão numérica-lingüística dos sinais de entrada (Pedrycz e Gomide, 1998). Outra abordagem possível considera que os pesos sinápticos da rede são grandezas *fuzzy*, e não os sinais processados por ela, como em (Ishibuchi et al., 1993). Essas e outras abordagens referentes ao tema podem ser encontradas de

⁴Discussões e um conjunto de referências sobre esse tema são encontradas em (Haykin, 1999).

forma compilada em (Buckley e Hayashi, 1994; Gupta e Rao, 1994).

De uma maneira geral pode-se afirmar que os modelos híbridos *neuro-fuzzy* apresentam grande potencial para o tratamento de problemas complexos nos domínios de aplicação pertinentes, dada a incorporação mútua daquelas habilidades individuais de cada classe de modelos envolvidos, discutidas anteriormente. É importante salientar que a estrutura hierárquica que será discutida no capítulo 4 é um exemplo típico dessa abordagem híbrida, onde os modelos são desenvolvidos utilizando equacionamentos baseados em lógica *fuzzy* e treinados a partir de técnicas neurais.

2.4 Modelos de Volterra

As pesquisas em torno dos fundamentos matemáticos que constituem a base dos atuais modelos de Volterra para sistemas dinâmicos não lineares iniciaram-se com V. Volterra em 1887, através do estudo de funcionais do tipo (Billings, 1980):

$$y(t) = \sum_{m=1}^{\infty} \int_{\Gamma} \cdots \int_{\Gamma} h_m(\tau_1, \tau_2, \dots, \tau_m) \prod_{i=1}^m u(t - \tau_i) d\tau_i \quad (2.1)$$

Na equação acima, conhecida como Série de Volterra, $u(t)$ e $y(t)$ são sinais na variável de tempo contínuo t e $h_m(\tau_1, \tau_2, \dots, \tau_m)$, para $m = 1, \dots, \infty$, são funções denominadas *kernels* (núcleos) de Volterra. O domínio de integração Γ é genericamente dado por $(-\infty, \infty)$, mas para uma representação causal tem-se $\Gamma = [0, \infty)$ e considerando ainda $u(t) = 0$ para $t < 0$ tem-se $\Gamma = [0, t]$.

A equação (2.1) é uma generalização do modelo linear clássico de convolução, i.e.,

$$y(t) = \int_{\Gamma} h(\tau)u(t - \tau)d\tau \quad (2.2)$$

onde $h(\tau)$ representa a resposta ao impulso de um sistema linear. Portanto, os kernels $h_m(\tau_1, \tau_2, \dots, \tau_m)$ são generalizações de m -ésima ordem e dimensão, para o caso não linear, da função de resposta ao impulso para um sistema linear.

A representação em (2.1) é do tipo:

$$y(t) = \mathcal{O}(\{u(\tau)\}_{\tau \in \Gamma}) \quad (2.3)$$

e é válida para sistemas dinâmicos não lineares nos quais a não linearidade em si (no caso o operador \mathcal{O}) não possui dinâmica, o que exclui sistemas com histerese (Eykhoff, 1974). Se o operador não linear do sistema a ser identificado for analítico, pode-se aproximá-lo por um desenvolvimento em série de Taylor (Kreysig, 1993). Eykhoff (1974) utilizou essa aproximação para obter a representação de Volterra em (2.1), generalizando o método

utilizado para representar sistemas lineares pela integral de convolução (Chen, 1998). Considerando a região de convergência, a importância (contribuição) dos termos da série de Taylor decai assintoticamente a partir de uma certa ordem. No caso de Volterra, a série é convergente no raio $|u(\cdot)| < 1$ (Dumont e Fu, 1993), o que é intuitivo a partir de (2.1) assumindo que os kernels possuem limitantes superior e inferior finitos ao longo do intervalo de tempo considerado. Na verdade, o raio de convergência trata-se apenas de um conceito matemático, uma vez que na prática pode-se sempre realizar uma transformação de variáveis de tal forma que a nova variável de entrada esteja normalizada.

A convergência da série implica na prática que é possível truncá-la a partir de um ponto além do qual os termos de ordem superior podem ser desprezados. Esse truncamento permite a seguinte representação aproximada de ordem finita η :

$$y(t) = \sum_{m=1}^{\eta} \int_{\Gamma} \cdots \int_{\Gamma} h_m(\tau_1, \tau_2, \dots, \tau_m) \prod_{i=1}^m u(t - \tau_i) d\tau_i \quad (2.4)$$

Boyd e Chua (1985) mostraram que o modelo (2.4) é capaz de representar com precisão arbitrária qualquer operador não linear em (2.3) que seja causal, contínuo e invariante no tempo, se uma das seguintes condições for satisfeita:

- i) $u(t) \in K$, onde K é um subconjunto compacto das funções contínuas quadraticamente integráveis em $[0, T]$, isto é, um subconjunto do espaço $L^2[0, T]$. Nesse caso a representação é restrita ao intervalo $[0, T]$.
- ii) $u(t) \in K$, onde K é um subconjunto das funções contínuas em $t \in \mathfrak{R}$ (i.e. $C(\mathfrak{R})$) dado por $K \in \{C(\mathfrak{R}) : |u(t)| \leq M_1 ; |u(s) - u(t)| \leq M_2 |s - t|, \forall s, t \in \mathfrak{R}\}$, ou seja, aqueles sinais limitados por M_1 e tendo uma constante de Lipschitz M_2 (Kreysig, 1993). Nesse caso o operador em (2.3) deve adicionalmente possuir *fading memory*. Para o resultado análogo no caso de tempo discreto a condição de Lipschitz é suprimida, devendo a entrada portanto ser apenas limitada superior e inferiormente.

No presente contexto enfoca-se o problema de modelagem de sistemas através de modelos discretos no tempo, que podem ser implementados computacionalmente de forma algorítmica. Portanto, considera-se nas discussões subseqüentes a versão discreta do modelo em (2.4), por simplicidade e sem perda de generalidade também para o caso SISO, dada por⁵:

$$y(k) = h_0(k) + \sum_{m=1}^{\eta} \sum_{n_1=0}^k \cdots \sum_{n_m=0}^k h_m(n_1, n_2, \dots, n_m) \prod_{i=1}^m u(k - n_i) \quad (2.5)$$

⁵Maiores detalhes sobre o caso contínuo podem ser encontrados em (Eykhoff, 1974).

já supondo que o sistema é causal ($h_m(n_1, \dots, n_m) = 0 \quad \forall n_i < 0$) e com condições iniciais não necessariamente nulas. A resposta livre no instante de tempo discreto k em decorrência dessas condições iniciais é representada pelo kernel de ordem zero $h_0(k)$.

Os kernels do modelo possuem as seguintes propriedades:

1. São ou podem ser feitos simétricos. No caso do kernel de segunda ordem, por exemplo, tem-se $h_2(n_1, n_2) = h_2(n_2, n_1)$.
2. $\lim_{n_i \rightarrow \infty} h_m(n_1, \dots, n_m) = 0$ para $\forall i \in \{1, \dots, m\}$, na representação de sistemas com *fading memory* (Boyd e Chua, 1985) ou memória finita (Eykhoff, 1974).

Quanto à primeira propriedade, tem-se que os termos $h_m(\cdot)$ referentes a qualquer permutação dos índices n_1, \dots, n_m na equação (2.5) multiplicam o mesmo valor $u(k - n_1)u(k - n_2) \dots u(k - n_m)$ naquela equação. Logo, pode-se arbitrar a simetria dos kernels sem nenhuma consequência para a capacidade de aproximação do modelo. Nesse caso, o problema de estimação dos kernels simplifica-se à estimação referente apenas a uma das metades do domínio.

A segunda propriedade é uma consequência da representação de sistemas nos quais a resposta a uma excitação na entrada desaparece com o tempo até não contribuir mais com a saída de forma significativa (*fading memory*) ou definitivamente (memória finita), o que remete a sistemas estáveis (Eykhoff, 1974; Boyd e Chua, 1985). Essa propriedade é importante entre outros fatores porque permite um truncamento em cada termo da série de Volterra. O truncamento do termo de ordem m , por exemplo, é realizado considerando que $h_m(n_1, n_2, \dots, n_m) = 0$ para $n_i > \epsilon_m$ ($\forall i \in \{1, \dots, m\}$), onde o escalar ϵ_m representa o período de tempo além do qual a contribuição do respectivo kernel é desprezível. Nesse caso, pode-se reescrever (2.5) como:

$$y(k) = h_0(k) + \sum_{m=1}^{\eta} \sum_{n_1=0}^{\epsilon_m} \dots \sum_{n_m=0}^{\epsilon_m} h_m(n_1, n_2, \dots, n_m) \prod_{i=1}^m u(k - n_i) \quad (2.6)$$

Dado um sistema dinâmico modelado por Volterra, isto é, descrito pelas equações (2.5) ou (2.6), o problema de identificação consiste em estimar o conjunto de kernels h_m ($m = 1, \dots, \eta$) do modelo. Esse problema é em princípio extremamente complexo, uma vez que os kernels são funções m -dimensionais não parametrizadas. Em (Eykhoff, 1974) discutem-se alguns métodos referentes à determinação dos kernels para o caso contínuo e em (Billings, 1980) apresenta-se um conjunto de referências relacionadas ao tema para os casos contínuo e discreto.

Uma abordagem intuitiva para o caso discreto seria generalizar a idéia dos modelos de resposta ao impulso finita (*Finite Impulse Response* – FIR) (Haykin, 1989; Ljung, 1999), substituindo o valor de cada kernel referente a uma determinada combinação dos instantes n_1, n_2, \dots, n_m por um parâmetro escalar independente a ser estimado. Essa abordagem seria possível para a modelagem de sistemas com *fading memory*, nos

quais, com base na segunda propriedade dos kernels de Volterra discutida anteriormente, o conjunto de parâmetros poderia ser finito através do truncamento dos termos da série, que resulta na equação (2.6). Porém, mesmo no caso dos modelos FIR que envolvem apenas o kernel unidimensional de primeira ordem, a quantidade de parâmetros necessários para uma aproximação satisfatória pode ser elevada, principalmente se o sistema a ser identificado possuir dinâmicas (e conseqüentemente resposta ao impulso) lentas. A inserção de kernels de ordens e dimensões superiores acarretaria um aumento, exponencial com a ordem, na quantidade de parâmetros, o que tornaria essa abordagem infactível para grande parte das aplicações práticas (Billings, 1980).

A discussão acima ilustra o grau de dificuldade relacionado ao problema de estimação dos kernels em modelos de Volterra. Contudo, na última década um conjunto de resultados matemáticos abriu novas perspectivas em relação a esse problema. Um desses resultados foi a obtenção de uma relação unilateral entre RNAs do tipo MLP e as séries de Volterra dadas por (2.5). Mais especificamente, mostrou-se em (Wray e Green, 1994; Bissessur e Naguib, 1995) que pode-se aproximar a equação analítica que descreve uma rede MLP com uma camada escondida (intermediária) de neurônios através de um desenvolvimento em série de Taylor que resulta em uma equação da forma dada em (2.5). Nesse caso, os termos $h_m(n_1, n_2, \dots, n_m)$ ($m = 1, \dots, \eta$) tornam-se funções dos parâmetros da rede, ou seja, os pesos sinápticos e as inclinações/polarizações das funções de ativação. Posteriormente, mostrou-se em (Marmarelis e Zhao, 1997; Chon et al., 1998) que a referida representação de RNAs como séries de Volterra é direta (sem qualquer aproximação de Taylor) quando funções de ativação polinomiais são utilizadas em substituição às sigmóides tradicionais. Esses resultados permitem a estimação dos kernels dos modelos de Volterra através do treinamento dos parâmetros de RNAs do tipo MLP, por exemplo através do algoritmo *back-propagation*.

Outro resultado importante (Dumont e Fu, 1993), que surgiu de forma independente daqueles discutidos acima, foi o desenvolvimento dos kernels de Volterra através de funções ortonormais, como as funções de Laguerre e Kautz (Broome, 1965; Wahlberg, 1994; Wahlberg e Mäkilä, 1996; Oliveira, 1997). Esse desenvolvimento resulta em um tipo de modelo denominado Wiener-Volterra e é detalhado a seguir devido à sua relevância em capítulos subseqüentes desse trabalho.

2.4.1 Modelos Wiener-Volterra

As bases de funções ortonormais, tais como as bases de Laguerre e Kautz, são completas no espaço de Lebesgue⁶ $L^2[0, \infty)$, de modo que qualquer função nesse espaço pode ser aproximada com precisão arbitrária por uma combinação linear de uma dessas bases (Lee, 1960; Dumont e Fu, 1993). No caso discreto tem-se que, para qualquer função $f(\kappa)$ quadraticamente somável em $[0, \infty)$, isto é,

⁶Espaço das funções contínuas quadraticamente integráveis em $[0, \infty)$ (Kreysig, 1978; Mäkilä, 1990).

$$\sum_{\kappa=0}^{\infty} f(\kappa)^2 < \infty \quad (2.7)$$

existe um inteiro $N > 0$ tal que para qualquer $\epsilon > 0$

$$\sum_{\kappa=0}^{\infty} \left(f(\kappa) - \sum_{i=1}^N \alpha_i \phi_i(\kappa) \right)^2 < \epsilon \quad (2.8)$$

onde $\{\phi_1(\kappa), \dots, \phi_N(\kappa)\}$ é o conjunto das N primeiras funções ortonormais e $\alpha_1, \dots, \alpha_N$ são escalares. Portanto, a representação

$$\hat{f}(\kappa) = \sum_{i=1}^N \alpha_i \phi_i(\kappa) \quad (2.9)$$

converge para $f(\kappa)$, sendo exata para N infinito, ou seja

$$f(\kappa) = \sum_{i=1}^{\infty} \alpha_i \phi_i(\kappa) \quad (2.10)$$

Para qualquer inteiro $j > 0$ tem-se de (2.10) que

$$\sum_{\kappa=0}^{\infty} f(\kappa) \phi_j(\kappa) = \sum_{\kappa=0}^{\infty} \phi_j(\kappa) \sum_{i=1}^{\infty} \alpha_i \phi_i(\kappa) = \sum_{i=1}^{\infty} \alpha_i \sum_{\kappa=0}^{\infty} \phi_j(\kappa) \phi_i(\kappa) \quad (2.11)$$

e utilizando a propriedade de ortonormalidade dada por

$$\sum_{\kappa=0}^{\infty} \phi_i(\kappa) \phi_j(\kappa) = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases} \quad (2.12)$$

tem-se de (2.11) que

$$\alpha_j = \sum_{\kappa=0}^{\infty} f(\kappa) \phi_j(\kappa) \quad (2.13)$$

As equações (2.10) e (2.13) representam o desenvolvimento (expansão) da função $f(\kappa)$ em uma base de funções ortonormais. A idéia dessa representação em modelos de

Volterra é desenvolver os kernels dos modelos através desse tipo de base. Esse desenvolvimento é possível para sistemas dinâmicos com *fading memory*, que admitem representações de Volterra com kernels truncados (e portanto quadraticamente somáveis). É importante lembrar que sistemas com integradores possuem resposta ao impulso com energia infinita, e portanto não se enquadram no requerimento acima. Como em geral a presença de integradores em um sistema real é conhecida, pode-se modelar a variação da saída do sistema ao invés do seu valor absoluto, o que é equivalente a retirar o(s) integrador(es) da malha de identificação.

O desenvolvimento aproximado do kernel de primeira ordem em (2.5) é realizado através da representação em (2.9), de forma tal que rescreve-se o respectivo termo da série ($m = 1$) como:

$$\begin{aligned} \sum_{n_1=0}^k h_1(n_1)u(k-n_1) &= \sum_{n_1=0}^k \sum_{i=1}^{N_1} \alpha_i \phi_i(n_1)u(k-n_1) \\ &= \sum_{i=1}^{N_1} \alpha_i \sum_{n_1=0}^k \phi_i(n_1)u(k-n_1) \\ &= \sum_{i=1}^{N_1} \alpha_i l_i(k) \end{aligned} \quad (2.14)$$

onde o termo l_i é a convolução da entrada u com a i -ésima função ortonormal ϕ_i . Como cada função ortonormal no tempo possui, a partir da transformada Z, uma função de transferência no domínio da frequência que define um filtro linear, o termo l_i é simplesmente o resultado de uma filtragem do sinal de entrada u , isto é:

$$l_i(k) = \Phi_i(q^{-1}) u(k) \quad (2.15)$$

onde $\Phi_i(q^{-1})$ é a função de transferência⁷ da i -ésima função ortonormal representada no operador deslocamento q^{-1} .

A aproximação dos kernels de ordem superior é realizada como uma extensão direta do resultado em (2.8) e da respectiva representação em (2.9) para o caso de funções multivariáveis, como ilustrado abaixo para o kernel de segunda ordem:

$$h_2(n_1, n_2) = \sum_{i=1}^{N_2} \sum_{j=1}^{N_2} \alpha_{i,j} \phi_i(n_1) \phi_j(n_2) \quad (2.16)$$

⁷Quando representada no operador q^{-1} a terminologia mais correta é função/operador de transferência pulsada (*pulse-transfer operator*) (Åström e Wittenmark, 1997).

Desenvolvendo todos os kernels até a η -ésima ordem de forma similar ao desenvolvimento em (2.16) e manipulando os respectivos termos da equação (2.5) de forma análoga à manipulação em (2.14), obtém-se:

$$y(k) = \alpha_0 + \sum_{m=1}^{\eta} \sum_{i_1=1}^{N_m} \cdots \sum_{i_m=1}^{N_m} \alpha_{i_1, \dots, i_m} l_{i_1}(k) l_{i_2}(k) \cdots l_{i_m}(k) \quad (2.17)$$

Nota-se portanto que o desenvolvimento dos kernels tornou o modelo *linear nos parâmetros* $\alpha_{(\cdot)}$. A importância desse resultado é que o problema de estimação dos kernels de Volterra resume-se agora à estimação linear desses parâmetros, o que pode ser realizado, por exemplo, através de algoritmos do tipo mínimos quadrados. O número total de parâmetros depende da quantidade de funções ortonormais utilizadas nos desenvolvimentos dos kernels. Como a dinâmica linear dada pelo respectivo conjunto de filtros ortonormais representa uma inserção de conhecimento *a priori* no modelo a respeito do comportamento básico do sistema, conforme será discutido no capítulo 5, é geralmente possível obter uma aproximação satisfatória utilizando um número relativamente pequeno de funções, isto é, $N_m \ll \epsilon_m$ ($m = 1, \dots, \eta$) onde ϵ_m é dado em (2.6).

O escalar α_0 em (2.17) possibilita a representação de sistemas com valor estacionário não nulo (nível constante) na variável de saída. Esse escalar não resulta do desenvolvimento do kernel de ordem zero $h_0(k)$ em (2.5), que não depende das variáveis n_1, \dots, n_m da expansão. Esse kernel é desnecessário no modelo (2.17) porque as condições iniciais que ele representa podem estar armazenadas nos estados dos filtros ortonormais que geram os sinais $l_1(k), \dots, l_n(k)$, onde $n = \max\{N_1, \dots, N_\eta\}$. O fato desses estados serem desconhecidos no início de algum procedimento de simulação não compromete a qualidade do modelo, uma vez que os filtros ortonormais são sempre estáveis, fazendo com que qualquer erro nos estados iniciais tenda assintoticamente a zero com o tempo.

Os modelos de Volterra apresentam características desejáveis no contexto de identificação de sistemas dinâmicos. Uma delas é a estrutura média móvel não linear (*Non-Linear Moving Average* – NLMA), sem recursão na saída, que evita a realimentação de eventuais erros de previsão. A representação em (2.17), especificamente, é do tipo

$$y(k) = \mathcal{H}(l_1(k), \dots, l_n(k)) \quad (2.18)$$

e portanto classifica-se como um modelo do tipo Wiener (Billings, 1980), ou seja, é uma dinâmica linear que gera os sinais $l_1(k), \dots, l_n(k)$ seguida por um mapeamento não linear estático \mathcal{H} para a saída $y(k)$. Por esta razão, esses modelos são denominados Wiener-Volterra. Devido à sua estrutura baseada em funções ortonormais, esse tipo de modelo apresenta uma série de vantagens com relação às estruturas não lineares clássicas⁸, tais como:

⁸Como aquelas classificadas em (Sjöberg et al., 1995).

- Tolerância a dinâmicas não modeladas e sensibilidade reduzida aos parâmetros estimados (Dumont e Fu, 1993), especialmente pela ausência de realimentação de erros de previsão.
- Não é necessário conhecer as ordens e atrasos de transporte do sistema, isto é, os vetores de regressão (ou regressores) das variáveis de entrada e saída.
- Dado que as bases de funções ortonormais são completas, é sempre possível aumentar a *capacidade de aproximação* do modelo simplesmente aumentando o número de funções, embora isso implique um aumento na quantidade de parâmetros e por conseqüência na incerteza (variância da estimação) sobre esses parâmetros (Kashyap e Rao, 1976).
- Ao contrário dos modelos FIR não lineares (NFIR ou NLFIR), que em geral necessitam um vetor de regressão de entrada com uma quantidade elevada de elementos, é geralmente possível obter uma precisão satisfatória com um número relativamente pequeno de funções (Oliveira, 1997).
- Habilidade para tratar atrasos de transporte (Mäkilä, 1990), mesmo longos ou variantes no tempo (Oliveira et al., 1996), sem impossibilitar sua consideração explícita na variável de entrada (Fu e Dumont, 1993).
- Desacoplamento natural das saídas no caso multivariável.

O principal problema com os modelos Wiener-Volterra é que, mesmo com uma quantidade pequena de funções ortonormais, a quantidade de parâmetros a serem estimados pode ser elevada para determinadas aplicações, dependendo do número η de termos da série em (2.17). Mais precisamente, a quantidade de parâmetros é dada por

$$\mu = \frac{1}{2} \left(1 + N_1 + \sum_{i=1}^{\eta} (N_i^i + N_i^{i-1}) \right) \quad (2.19)$$

já considerando a hipótese de simetria dos kernels e do respectivo desenvolvimento em funções ortonormais. Para limitar o número de parâmetros e simplificar a equação do modelo, a série é usualmente truncada na prática a partir do termo de segunda ordem ($\eta = 2$), desprezando-se os termos de ordem superior. Assim, os modelos Wiener-Volterra possuem duas aproximações, uma devido ao truncamento da série propriamente dita (η) e a outra devido ao truncamento da expansão ortonormal (N_1, \dots, N_η). Como conseqüência dessas aproximações, esses modelos tornam-se apropriados para a modelagem de sistemas com não linearidades “brandas” (*mild non-linearities*) (Billings, 1980; Dumont e Fu, 1993), como é o caso, por exemplo, de sistemas bilineares que são descritos por equações muito similares à própria série de Volterra.

De maneira geral os modelos de Volterra e Wiener-Volterra apresentam características importantes no contexto de identificação de sistemas não lineares. Como

discutido anteriormente, são teoricamente capazes de aproximar arbitrariamente bem uma ampla classe de sistemas através de métodos de estimação paramétrica bem estabelecidos na literatura. No entanto, ao contrário dos modelos *fuzzy*/neurais, são restritos a sistemas dinâmicos e não permitem a extração de qualquer conhecimento qualitativo a respeito dos sistemas identificados. Além disso, o caso multivariável é caracterizado pela soma dos modelos individuais correspondentes a cada entrada, não contemplando interações não lineares entre as diferentes entradas, o que pode ser restritivo na modelagem de sistemas que possuem essa característica. Já a restrição a sistemas estáveis apresenta um lado positivo: Garante-se que as representações de Volterra e Wiener-Volterra de um sistema estável são sempre estáveis, o que em princípio não se pode garantir para as representações dadas pelos modelos *fuzzy*/neurais dinâmicos convencionais. Esse problema é abordado no capítulo 5 através do desenvolvimento de modelos *fuzzy*/neurais com funções ortonormais.

2.5 Resumo

A crescente evolução dos sistemas computacionais modernos tem viabilizado o desenvolvimento e a implementação algorítmica de modelos matemáticos cada vez mais flexíveis para a representação de sistemas complexos nos mais variados campos de aplicação, tais como aproximação de funções, classificação de padrões, previsão de séries temporais, processamento de sinais, identificação e controle de processos dinâmicos, entre outros. Entre os vários modelos desenvolvidos com enfoque em uma ou mais das aplicações acima, pode-se destacar os modelos *fuzzy*, neurais, *neuro-fuzzy* e de Volterra/Wiener-Volterra.

Os modelos *fuzzy* têm ganho importância crescente nos últimos anos por serem aproximadores universais capazes de tratar informações tanto de natureza quantitativa como qualitativa. Existem três abordagens fundamentais: Os modelos lingüísticos (também denominados modelos de Mamdani), os modelos Takagi-Sugeno e os modelos relacionais. Os primeiros são lingüisticamente interpretáveis e proporcionam transparência durante a fase de desenvolvimento. Contudo, a obtenção automática de um conjunto de regras lingüísticas que proporcione uma representação quantitativa e qualitativamente adequada de um dado sistema de interesse a partir de dados desse sistema não é uma tarefa trivial. Nos modelos Takagi-Sugeno a solução desse problema é facilitada pela ausência de termos lingüísticos associados às saídas das regras, o que por outro lado compromete a interpretabilidade das mesmas. Devido à sua característica estrutural constituída de uma composição de diferentes modelos locais, o que usualmente permite aproximações precisas através de um número reduzido de parâmetros, os modelos Takagi-Sugeno têm se mostrado particularmente interessantes para aplicações em controle. Os modelos relacionais constituem uma alternativa intermediária entre as duas classes anteriores, apresentando características importantes tanto sob aspectos numéricos quanto lingüísticos.

Os modelos neurais reúnem atualmente diversas abordagens matemáticas distintas

e com domínios de aplicação variados, tais como os mapas auto-organizáveis, as redes recorrentes e as redes *feedforward*. Dentre essas últimas destacam-se as redes RBF e MLP que, assim como os modelos *fuzzy*, destinam-se à aproximação de mapeamentos não lineares genéricos, podendo ser utilizadas nas diversas aplicações mencionadas no início dessa seção. Assim como os modelos *fuzzy*, os modelos neurais têm se estabelecido como uma das ferramentas fundamentais para o tratamento de problemas de modelagem complexos basicamente pela capacidade de aprendizado e de aproximação universal desses modelos através de estratégias comuns às arquiteturas mono e multivariáveis.

Através da integração das características de interpretabilidade/transparência e tratamento de incertezas dos modelos *fuzzy* e das técnicas de treinamento/aprendizado neurais, surge uma classe híbrida cujos elementos são denominados modelos *neuro-fuzzy*. A incorporação mútua das potencialidades de cada uma das classes individuais torna esses modelos de particular importância no contexto de modelagem de sistemas complexos.

Os modelos de Volterra representam uma generalização dos modelos lineares de convolução para a representação de sistemas dinâmicos não lineares. A principal dificuldade associada a esses modelos é a complexidade envolvida no problema de determinação dos kernels de Volterra. Quando esses kernels são desenvolvidos utilizando uma base de funções ortonormais, no entanto, obtém-se uma representação do tipo Wiener (modelo Wiener-Volterra) que é linear nos parâmetros de projeto livre. Essa representação apresenta uma série de vantagens quando comparada às estruturas dinâmicas tradicionais, mas é restrita a sistemas estáveis. Por outro lado, garante-se que a referida representação de um sistema estável é sempre estável.

Os modelos de Volterra e Wiener-Volterra são capazes de aproximar arbitrariamente bem uma ampla classe de sistemas através de métodos de estimação paramétrica bem estabelecidos na literatura. No entanto, ao contrário dos modelos *fuzzy*/neurais, são restritos a sistemas dinâmicos e o caso multivariável não contempla interações não lineares entre as diferentes entradas. Esses modelos também não permitem a inserção ou extração de informação qualitativa a respeito dos sistemas identificados.

2.6 Contribuições

A principal contribuição do presente capítulo é uma revisão comparativa original a respeito das diferentes classes de modelos não lineares nas quais se sustenta o presente trabalho, que são os modelos *Fuzzy*, Neurais, *Neuro-Fuzzy* e de Volterra/Wiener-Volterra.

Capítulo 3

Estimação e Interpretação de Modelos Relacionais *Fuzzy*

Esse capítulo aborda os problemas de modelagem quantitativa e qualitativa de sistemas complexos utilizando modelos relacionais *fuzzy*. Apresenta-se um retrospecto sobre o desenvolvimento desses modelos e propõe-se uma abordagem integrada para sua otimização e subsequente interpretação lingüística, incluindo uma metodologia para simplificação dos modelos resultantes. Apresenta-se também um conjunto de exemplos acadêmicos e práticos para ilustrar a aplicação dos métodos propostos.

3.1 Introdução

O trabalho pioneiro em estimação de modelos relacionais foi desenvolvido por Czogała e Pedrycz (1981). Nesse trabalho, e em outros subsequentes (Pedrycz, 1981; Czogała e Pedrycz, 1982; Pedrycz et al., 1984; Pedrycz, 1984b; Higashi e Klir, 1984a), os autores estudaram a utilização de métodos analíticos para a solução de equações relacionais visando a representação de sistemas na presença de dados *fuzzy*. Um conjunto de métodos analíticos para a solução de equações relacionais pode ser encontrado na literatura (Sanchez, 1976; Higashi e Klir, 1984b; Pedrycz, 1985b; Pedrycz, 1985a; Stamou e Tzafestas, 1999), incluindo tutoriais abrangentes a esse respeito (Di Nola et al., 1991; Pedrycz, 1991; Pedrycz, 1993). Em todos esses métodos assume-se que o conjunto de soluções exatas para o problema não é vazio, e a aplicação dos mesmos sem que essa hipótese seja satisfeita geralmente resulta em soluções inconsistentes. Essa hipótese é restritiva para aplicações práticas, especialmente porque quanto maior o conjunto de dados mais difícil é a existência de soluções exatas para uma equação relacional, principalmente se os dados estiverem contaminados com ruído. Por essa razão, muitos trabalhos foram dedicados a buscar soluções aproximadas para essas equações, envolvendo a análise, seleção e modificação dos conjuntos de dados disponíveis (Pedrycz, 1984b; Gottwald e Pedrycz, 1986; Pedrycz, 1988; Pedrycz, 1990) bem como modificações estruturais das próprias equações (Pedrycz, 1985c).

Além das dificuldades encontradas na prática mesmo para obter soluções aproximadas satisfatórias para as equações relacionais, outro problema associado à modelagem relacional é que uma relação *fuzzy* em sua forma original não permite uma interpretação lingüística direta. Esse problema foi abordado por Tong (1976) e outros autores (Di Nola et al., 1985) em seus trabalhos sobre as propriedades de decomposição de matrizes relacionais, previamente estimadas, em conjuntos de regras lingüísticas. Contudo, as condições necessárias para a verificação dessas propriedades são restritivas para aplicações práticas (Pedrycz et al., 2001), especialmente quando a matriz (relação) a ser decomposta é resultante de um procedimento de estimação sujeito a dados contaminados com ruído. Mesmo quando essas condições são satisfeitas, não existe um procedimento sistemático para a obtenção de um conjunto de regras consistente e lingüisticamente claro.

Com base nas discussões acima, os modelos relacionais podem parecer não constituírem uma ferramenta de modelagem apropriada devido às dificuldades que têm sido apontadas em termos da sua estimação e interpretação. Pedrycz (1983; 1984a) simplificou ambos esses problemas propondo a utilização da técnica de *discretização fuzzy* (Willaeys e Malvache, 1981) para a representação de dados em equações relacionais. Essa técnica foi desde então amplamente difundida no contexto desses modelos, especialmente porque proporciona uma redução significativa em suas dimensões (ver apêndice A). A partir do conceito de discretização *fuzzy*, cada elemento de uma matriz relacional passou a ser visto como uma medida de possibilidade (Zadeh, 1978) de uma regra lingüística relacionando conjuntos referenciais *fuzzy* das variáveis de entrada e saída específicos (Pedrycz, 1984a; Graham e Newell, 1989; Branco e Dente, 2000). Em (Campello e Amaral, 1999b) essa idéia foi aprimorada através da introdução do conceito de *representabilidade* de um elemento relacional. Esse conceito permitiu o desenvolvimento de uma metodologia sistemática e consistente para a extração de conhecimento lingüístico (i.e., regras *fuzzy*) a partir de modelos relacionais, independente do tipo de operador de composição selecionado para implementá-los (Campello e Amaral, 2000a; Campello e Amaral, 2001). Essa metodologia foi posteriormente estendida para permitir também a simplificação (redução) dos conjuntos de regras obtidos (Campello e Amaral, nd), e constitui o tema da seção 3.4.

O conceito de discretização *fuzzy* também permitiu o surgimento de novas abordagens para o problema de estimação numérica dos modelos relacionais. Um método de estimação pioneiro baseado nesse conceito é conhecido como aproximação lingüística de Pedrycz (Pedrycz, 1984a; Graham e Newell, 1988; Bourke e Fisher, 2000a). Nesse método, uma matriz relacional aproximada é obtida agregando-se os produtos Cartesianos *fuzzy* de um conjunto de dados. Embora o esforço computacional exigido seja mínimo, o algoritmo pode agregar elementos conflitantes na matriz, especialmente na presença de dados contaminados com ruído. Apesar disso, esse método aproximado mostrou-se útil para gerar uma solução inicial para algoritmos mais elaborados (Xu e Lu, 1987; Xu, 1989; Graham e Newell, 1989; Lee et al., 1994).

Atualmente, o uso de discretização *fuzzy* combinado com métodos numéricos de

otimização tem mostrado ser a melhor abordagem para a modelagem relacional à medida que metodologias cada vez mais eficazes têm sido propostas na literatura (Bourke e Fisher, 2000b). Exemplos incluem o uso de técnicas de neuro-computação (Oliveira, 1993a), métodos baseados em gradiente (Yi e Chung, 1993; Oliveira e Lemos, 1995), algoritmos de mínimos quadrados (Oliveira e Lemos, 1998), programação quadrática (Campello e Amaral, 1998b; Campello e Amaral, 1999d), computação evolutiva (Chung e Lee, 1998) e identificação preditiva (Wong et al., 2000).

Embora a maioria dessas metodologias geralmente proporcionem resultados satisfatórios, elas apresentam ao menos um dos seguintes problemas: *i*) Não existe garantia de convergência global ou local; *ii*) A interpretabilidade dos modelos resultantes não é levada em consideração. Por essas razões, uma abordagem numérica ótima para a estimação de modelos relacionais visando a subsequente interpretação lingüística desses modelos foi proposta em (Campello e Amaral, 2000a; Campello e Amaral, 2001; Campello e Amaral, nd). Essa abordagem, baseada na utilização de diferentes algoritmos de mínimos quadrados e programação quadrática, constitui o tema da seção 3.3. Antes porém, apresenta-se uma descrição dos modelos relacionais.

3.2 Modelos Relacionais *Fuzzy*

Os modelos relacionais possuem a estrutura padrão de sistemas *fuzzy* composta de interfaces de entrada (fuzzificação) e saída (defuzzificação) e um bloco de processamento (inferência). O bloco de processamento é constituído por uma equação relacional *fuzzy* (Pedrycz, 1993), como segue:

$$\hat{Y} = X_1 \bullet X_2 \bullet \cdots \bullet X_n \bullet R \quad (3.1)$$

onde $\hat{Y} = [\hat{Y}_1 \cdots \hat{Y}_{c_0}]^T$ e $X_i = [X_{i_1} \cdots X_{i_{c_i}}]^T$ ($i = 1, \dots, n$) são respectivamente as representações *fuzzy* da saída \hat{y} e das entradas x_i *crisp*, R ($c_1 \times \cdots \times c_n \times c_0$) é a matriz relacional e “ \bullet ” representa o operador de composição. Os diferentes tipos de entradas e de saída se relacionam através das respectivas interfaces, isto é, X_i é obtida pela fuzzificação de x_i e \hat{y} pela defuzzificação de \hat{Y} .

A equação (3.1) pode ainda ser simplificada como

$$\hat{Y} = \Psi \bullet R \quad (3.2)$$

onde Ψ ($c_1 \times \cdots \times c_n$) é o Produto Cartesiano dado por

$$\Psi = X_1 \times X_2 \times \cdots \times X_n \quad (3.3)$$

que é genericamente definido como o produto de uma norma triangular (norma-t) sobre o espaço dos produtos cruzados das entradas *fuzzy* (Lee, 1990).

A equação (3.2) assume uma forma específica para um determinado operador de composição. Em (Campello, 1997; Campello et al., 1997; Campello et al., 1998; Campello e Amaral, 1998b; Campello e Amaral, 1998a; Campello e Amaral, 1999d) considerou-se o operador max-t, que tem sido historicamente o operador de composição mais utilizado no contexto de sistemas *fuzzy*. Nesse caso, o modelo relacional em (3.2) pode ser reescrito como

$$\hat{Y}_j = \bigvee_{m=1}^l \Psi_m \mathfrak{t} R_{m,j}, \quad j = 1, \dots, c_0 \quad (3.4)$$

onde “ \vee ” representa o operador max, “ \mathfrak{t} ” é uma norma triangular e $l = \prod_{i=1}^n c_i$. É importante observar que, embora a matriz Ψ seja originalmente n -dimensional em (3.2), essa matriz é representada por simplicidade de notação na equação acima como um vetor¹ ($l \times 1$). Analogamente, R é representada como uma matriz $l \times c_0$.

Outro operador de composição também considerado no presente trabalho é o operador média-produto (Oliveira e Lemos, 1997; Oliveira e Lemos, 1998), que resulta em modelos com uma série de propriedades discutidas nas seções seguintes. Utilizando esse operador pode-se reescrever a equação (3.2) como

$$\hat{Y}_j = \frac{1}{l} \sum_{m=1}^l \Psi_m R_{m,j}, \quad j = 1, \dots, c_0 \quad (3.5)$$

sendo que nesse caso Ψ em (3.3) torna-se o Produto de Kronecker (Graham, 1981) das entradas *fuzzy*, isto é,

$$\Psi = [\Psi_1 \ \dots \ \Psi_l]^T = X_1 \otimes X_2 \otimes \dots \otimes X_n \quad (3.6)$$

3.3 Estimação Ótima de Modelos Relacionais

Dado um conjunto de pares de dados de entrada/saída $\{(x_1, \dots, x_n, y)\}$ medidos a partir de um sistema a ser modelado, o problema de estimação no presente contexto consiste em obter uma matriz relacional R que satisfaça algum critério de otimalidade para o modelo adotado. Esse problema pode ser convenientemente simplificado através de uma escolha adequada da estrutura do modelo, que é discutida a seguir.

¹Aplicação do operador *vec*.

3.3.1 Seleção de Estrutura

O conceito de discretização *fuzzy* é adotado em todos os desenvolvimentos subseqüentes em função das razões já discutidas na seção 3.1. De acordo com esse conceito, os dados de entrada e saída podem ser representados (fuzzificados) utilizando conjuntos referenciais *fuzzy*, como segue (ver apêndice A):

$$X_i = [\mathcal{X}_{i_1}(x_i) \mathcal{X}_{i_2}(x_i) \cdots \mathcal{X}_{i_{c_i}}(x_i)]^T \quad (3.7)$$

$$Y = [\mathcal{Y}_1(y) \mathcal{Y}_2(y) \cdots \mathcal{Y}_{c_0}(y)]^T \quad (3.8)$$

onde \mathcal{X}_{i_h} é o h -ésimo conjunto referencial *fuzzy* da i -ésima variável de entrada e \mathcal{Y}_j é o j -ésimo conjunto referencial *fuzzy* da variável de saída. Os conjuntos referenciais *fuzzy* \mathcal{X}_{i_h} e \mathcal{Y}_j são definidos sobre os universos de discurso (UoDs) \mathbf{X}_i e \mathbf{Y} de x_i e y , respectivamente.

Considerando a saída estimada do modelo relacional, \hat{y} , a saída medida do sistema real, y , e as suas respectivas representações *fuzzy* dadas respectivamente por \hat{Y} e Y , tem-se que as relações entre essas variáveis podem ser escritas como

$$Y = \mathcal{L}(y) \quad (3.9)$$

$$\hat{y} = \mathcal{N}(\hat{Y}) \quad (3.10)$$

onde \mathcal{L} e \mathcal{N} representam os mapeamentos dados respectivamente pela equação (3.8) e por um determinado operador de defuzzificação. O operador de defuzzificação da média ponderada (*weighted average*) é adotado nos desenvolvimentos seguintes para a implementação do mapeamento \mathcal{N} . Esse operador, também conhecido como média dos centros, consiste da simplificação usual do operador clássico do centro de gravidade, dada por (Pedrycz, 1993):

$$\hat{y} = \frac{\sum_{j=1}^{c_0} \hat{Y}_j \theta_j}{\sum_{j=1}^{c_0} \hat{Y}_j} \quad (3.11)$$

onde θ_j é o valor modal do j -ésimo conjunto referencial *fuzzy*, \mathcal{Y}_j , que coincide com seu centro sempre que esse conjunto for simétrico.

Um aspecto importante na seleção da estrutura dos modelos relacionais diz respeito aos conjuntos referenciais *fuzzy* das variáveis de entrada e saída nas equações (3.7) e

(3.8). Em princípio, esses conjuntos devem apenas possuir um significado lingüístico claro, caso a interpretação dos modelos resultantes seja desejada. Essa condição recai na denominada Estrutura de Cognição (*frame of cognition*) (Pedrycz, 1995; Pedrycz e Gomide, 1998), revista no apêndice A. Por outro lado, se apenas o problema de modelagem quantitativa for de interesse, os conjuntos referenciais podem ser obtidos a partir de algoritmos ótimos como *fuzzy clustering* (Sugeno e Yasukawa, 1993; Babuška e Verbruggen, 1996). Em qualquer dos casos acima uma especificação adicional é necessária para tornar convexo o problema de otimização referente à estimação da matriz relacional dos modelos. Trata-se da condição de interface de saída ótima (Oliveira, 1993b; Pedrycz e Oliveira, 1996c), dada por

$$\forall y \in \mathbf{Y} : \mathcal{N}(\mathcal{L}(y)) = y \quad (3.12)$$

Essa condição é também conhecida como critério de equivalência de informações (Pedrycz e Oliveira, 1996b; Pedrycz e Oliveira, 1996a) e implica que os mapeamentos \mathcal{L} e \mathcal{N} constituem mecanismos de codificação-decodificação sem perda (*lossless encoding-decoding mechanisms*) (Pedrycz e Gomide, 1998), ou seja, são tais que qualquer valor em \mathbf{Y} pode ser completamente recuperado após uma seqüência de fuzzificação-defuzzificação. Um exemplo típico é dado pelos conjuntos referenciais *fuzzy* com funções de pertinência triangulares sobrepostas em 0.5, como aqueles ilustrados na figura 3.1, que precisamente satisfazem a condição (3.12) (Pedrycz, 1994). Outros tipos de conjuntos *fuzzy* podem ser otimizados segundo o critério (3.12) sujeito a restrições de interpretabilidade lingüística, como discutido em (Oliveira, 1995).

Finalmente, o projeto da estrutura dos modelos relacionais é concluído com a seleção do operador de composição média-produto, que implica a representação dada pela equação (3.5). Esse projeto agrega uma série de propriedades aos modelos que são discutidas nas seções seguintes.

3.3.2 Problema de Otimização

O problema de otimização para a estimação dos modelos relacionais é desenvolvido utilizando um resultado apresentado em (Campello, 1997; Campello et al., 1997; Campello et al., 1998), revisto abaixo na forma da seguinte proposição:

Proposição 3.1 *Se os mapeamentos \mathcal{L} e \mathcal{N} em (3.9) e (3.10) satisfazem a condição (3.12), então a igualdade $\hat{Y} = Y$ entre as saídas fuzzy do modelo e do sistema a ser modelado resulta na igualdade $\hat{y} = y$ entre as suas respectivas saídas crisp.*

Prova: Se $\hat{Y} = Y$, então a equação (3.10) pode ser reescrita como:

$$\hat{y} = \mathcal{N}(Y) \quad (3.13)$$

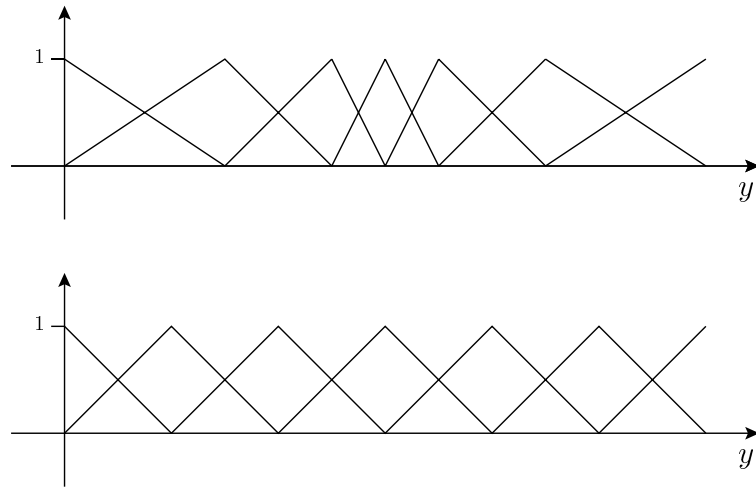


Figura 3.1: Exemplos de partioes *fuzzy* (7 conjuntos referenciais) que satisfazem o critrio de equivalncia de informaoes.

Substituindo (3.9) em (3.13) produz-se:

$$\hat{y} = \mathcal{N}(\mathcal{L}(y)) \quad (3.14)$$

Logo, a igualdade $\hat{y} = y$  obtida a partir das equaoes (3.14) e (3.12). ■

Esse resultado significa que o procedimento de estimaco de modelos relacionais pode ser realizado a partir da minimizao (sobre a matriz relacional R) de um certo critrio de distncia entre as sadas *fuzzy* do sistema a ser modelado e do respectivo modelo, como segue:

$$\min_R J(Y, \hat{Y}) \quad (3.15)$$

onde J denota um critrio de distncia ou custo genrico (funao objetivo). A principal vantagem do critrio acima  que as funoes de pertinncia dos conjuntos referenciais *fuzzy* da varivel de sada so explicitamente consideradas no problema de otimizao atravs do mapeamento \mathcal{L} , ou seja, da equaao (3.8). Dessa forma, ao contrrio dos critrios usuais nos quais apenas as sadas *crisp* so consideradas², o critrio em (3.15) relaciona diretamente a otimizao do modelo aos termos lingsticos referentes  sua varivel de sada. Como o mapeamento de fuzzificao \mathcal{L} produz um nico vetor Y cujos elementos representam a pertinncia de uma dada sada *crisp* y a cada termo lingstico

²Um exemplo interessante  o trabalho de Rovatti e Guerrieri (1996), onde a matriz relacional  tratada como um conjunto *fuzzy* de regras *fuzzy*.

$\mathcal{Y}_{(\cdot)}$, o procedimento de otimização força o modelo a produzir saídas \hat{Y} linguisticamente consistentes, o que se torna possível a partir de uma matriz relacional também consistente. Por outro lado, a minimização de um critério de distância envolvendo apenas as saídas *crisp* é problemática no contexto de modelagem qualitativa, podendo acarretar inconsistências, uma vez que um número infinito de diferentes saídas *fuzzy* \hat{Y} do modelo podem produzir o mesmo valor *crisp* \hat{y} como resultado do procedimento de defuzzificação.

O critério em (3.15) também apresenta propriedades matemáticas e numéricas importantes. A principal delas é a sua independência em relação ao mapeamento de defuzzificação \mathcal{N} em (3.11), que torna não convexos os critérios convencionais *crisp* e os respectivos problemas de otimização. Outra propriedade é que o problema (3.15) pode ser resolvido separadamente para cada componente dos vetores Y e \hat{Y} porque esses componentes são totalmente independentes entre si. Em outras palavras, o j -ésimo componente de \hat{Y} depende apenas da j -ésima coluna da matriz R , o que permite que o procedimento de otimização possa ser efetuado, com complexidade computacional inferior, de forma individual sobre cada coluna da matriz. Sendo assim, dado um conjunto de N pares de dados de entrada e saída do sistema a ser modelado, i.e. $\{ (x_1(k), \dots, x_n(k), y(k)) \}_{k=1}^N$, um modelo relacional desse sistema pode ser estimado através da solução do seguinte problema de otimização:

$$\min_{\phi_j} J_j = \frac{1}{2} \sum_{k=1}^N w(k) (Y_j(k) - \hat{Y}_j(k))^2, \quad j = 1, \dots, c_0 \quad (3.16)$$

onde $w(\cdot) > 0$ são termos de ponderação, ϕ_j representa a j -ésima coluna da matriz R , i.e., $\phi_j \triangleq [R_{1,j} \dots R_{l,j}]^T$, e a saída $\hat{Y}_j(k)$ é rescrita a partir de (3.5) como

$$\hat{Y}_j(k) = \frac{1}{l} \Psi^T(k) \phi_j \quad (3.17)$$

O modelo acima é *linear nos parâmetros*, o que implica que o problema (3.16) é convexo. Logo, esse problema admite a solução global ótima dos mínimos quadrados ponderados (Ljung, 1999), dada por

$$\phi_j = [\Psi_N^T W_N \Psi_N]^{-1} \Psi_N^T W_N Y_{N_j}, \quad j = 1, \dots, c_0 \quad (3.18)$$

onde $W_N = \text{diag}(w(1), \dots, w(N))$ é a matriz de ponderação e os termos Ψ_N (matriz $N \times l$) e Y_{N_j} (vetor $N \times 1$) são dados por

$$\Psi_N = \frac{1}{l} [\Psi(1) \dots \Psi(N)]^T \quad (3.19)$$

$$Y_{N_j} = [Y_j(1) \dots Y_j(N)]^T \quad (3.20)$$

3.3.3 Aspectos Numéricos

Os seguintes aspectos com respeito ao problema (3.16) e à sua solução em (3.18) devem ser consideradas:

1. A formulação de (3.16) a (3.20) diz respeito a um problema de modelagem denominado *completo*. Essa denominação está associada a modelos que representam, através de regras ou parâmetros específicos, todas as regiões do domínio de entrada do problema em questão. Uma base completa de regras *fuzzy*, por exemplo, é tal que existe uma regra para cada partição *fuzzy* (Pedrycz, 1993) do seu domínio, ou seja, para cada combinação possível dos termos lingüísticos das suas variáveis de entrada. Esse é o caso do modelo em (3.17), em que cada elemento relacional está associado à uma das possíveis combinações (antecedente) dos seus conjuntos referenciais *fuzzy* de entrada. No entanto, em alguns casos uma representação *incompleta* é desejada. É o caso de problemas em que determinadas regiões (partições) do domínio não são representadas pelo conjunto de dados disponíveis ou pelo próprio sistema em questão. Nesses casos, os elementos relacionais referentes a essas regiões podem ser anulados e retirados dos vetores de parâmetros ϕ_j a serem otimizados em (3.16). A solução em (3.18) continua sendo a mesma, porém as respectivas colunas de Ψ_N devem ser removidas.
2. As matrizes e vetores de dados envolvidos, isto é, Ψ_N e Y_{N_j} , estão normalizados no intervalo $[0,1]$. Essa “normalização automática” deve-se ao procedimento de fuzzificação dos dados de entrada/saída através das equações (3.7) e (3.8). A normalização dos dados é interessante sob o ponto de vista numérico porque reduz problemas de mau condicionamento da matriz $\Psi_N^T W_N \Psi_N$, cuja inversa deve ser calculada.
3. O cálculo da matriz inversa $[\Psi_N^T W_N \Psi_N]^{-1}$ necessita ser realizado apenas uma vez, pois essa matriz não depende do índice j .
4. A matriz $\Psi_N^T W_N \Psi_N$ (Hessiana das funções objetivo J_j) é semi-definida ou definida positiva por construção, dado que W_N é definida positiva. É bem conhecido que se essa matriz for definida positiva, então sua inversa existe e a solução (3.18) é solução ótima global estrita para o problema (3.16). Essa Hessiana será definida positiva se os dados de entrada disponíveis forem adequados. No presente contexto, condições necessárias para isso são: *i)* Os dados devem estar distribuídos em todas as partições *fuzzy* do domínio representadas pelo modelo, evitando a presença de linhas/colunas completamente nulas na matriz; *ii)* Os universos de discurso das variáveis do modelo devem estar totalmente cobertos pelos respectivos conjuntos referenciais (ver “estrutura de cognição” no apêndice A) de forma que não haja dados *crisp* que produzam dados *fuzzy* completamente nulos como resultado da sua fuzzificação; *iii)* Se os dados representarem um sistema dinâmico,

a respectiva taxa de amostragem não deve ser excessivamente rápida. A verificação desses itens reduz a possibilidade de singularidade (dependência linear) e mau condicionamento numérico da matriz Hessiana. É importante ressaltar que essa questão, embora bem estabelecida com relação à identificação de sistemas lineares (Ljung, 1999), ainda merece estudos mais direcionados no contexto de sistemas *fuzzy*.

5. A possibilidade de mau condicionamento da matriz Hessiana, que traz dificuldades ou até mesmo impossibilita sua inversão numérica, estimula a busca de alternativas à solução dada pela equação (3.18). Uma delas é a solução recursiva dessa equação (Eykhoff, 1974; Franklin et al., 1990; Wellstead e Zarrop, 1991; Ljung, 1999) que constrói implicitamente (recursivamente) a inversa da Hessiana sem que qualquer inversão matricial explícita seja necessária. Além da redução significativa do custo computacional, essa solução é interessante por permitir a estimação *on-line* do modelo e sua aplicação adaptativa a sistemas variantes no tempo. A solução recursiva é descrita por

$$\begin{aligned} \phi_j(k+1) &= \phi_j(k) + \\ &+ \frac{P(k)\Psi(k+1)(lY_j(k+1) - \Psi^T(k+1)\phi_j(k))}{\frac{\lambda}{\gamma} + \Psi^T(k+1)P(k)\Psi(k+1)} \end{aligned} \quad (3.21)$$

$$P(k+1) = \frac{1}{\lambda} \left[P(k) - \frac{P(k)\Psi(k+1)\Psi^T(k+1)P(k)}{\frac{\lambda}{\gamma} + \Psi^T(k+1)P(k)\Psi(k+1)} \right] \quad (3.22)$$

sendo $k = 0, \dots, N-1$ ($j = 1, \dots, c_0$). As condições iniciais são usualmente escolhidas como $\phi_j(0) = \mathbf{0}$ e $P(0) = \alpha I_l$, onde I_l é a matriz identidade ($l \times l$) e α é um escalar. Os escalares λ e γ são denominados respectivamente de fator de esquecimento e ganho do fator de esquecimento, sendo esse último em geral selecionado como $1 - \lambda$ de modo a normalizar a matriz de covariância P (Ljung, 1999). Esses escalares estão relacionados às ponderações $w(k)$ em (3.16) da seguinte forma:

$$w(k) = \gamma \lambda^{N-k} \quad (3.23)$$

Desta equação pode-se observar que a ponderação é decrescente com os instantes de tempo passados, isto é, os dados em instantes anteriores a N são exponencialmente “esquecidos” com fator $0 < \lambda < 1$. Essa ponderação permite a adaptação do modelo a sistemas variantes no tempo. Para sistemas invariantes no tempo tem-se o caso particular não ponderado, isto é, $\lambda = \gamma = 1$, com a matriz de ponderação dada por $W_N = I_N$ (identidade $N \times N$).

Uma questão importante refere-se aos elementos da matriz relacional cujas partições *fuzzy* do domínio de entrada não são estimuladas pelos dados disponíveis. Ao contrário da solução analítica em (3.18), que pode atribuir valores inadequados a esses elementos devido aos problemas de mau condicionamento discutidos nos itens anteriores, a solução recursiva dos mínimos quadrados (ou algoritmo RLS) dada pelas equações (3.21) e (3.22) não atualiza esses elementos com relação às suas condições iniciais. Se essas condições forem nulas, como foi sugerido acima, esses elementos não terão qualquer influência sobre a saída do modelo resultante e as regras lingüísticas correspondentes irão conseqüentemente ser desconsideradas no procedimento de interpretação lingüística discutido na seção 3.4.

O algoritmo RLS apresentado acima não é a única forma de calcular recursivamente a solução (3.18). Quando não há precisão computacional suficiente, algoritmos numericamente robustos baseados na fatorização da matriz de covariância P podem ser utilizados, tal como a fatorização U-D (Wellstead e Zarrop, 1991; Ljung, 1999).

6. Na seção 3.4 será mostrado que, quando se deseja extrair conhecimento lingüístico a partir dos modelos relacionais obtidos, torna-se necessária a inclusão de restrições de não negatividade no problema de otimização, que por sua vez pode ser solucionado através de programação quadrática (Bazaraa et al., 1993). Nesse caso, o problema (3.16) é rescrito como

$$\begin{aligned} \min_{\phi_j} \quad & J'_j = \frac{1}{2} \phi_j^T H \phi_j + f_j^T \phi_j, \quad j = 1, \dots, c_0 \\ \text{s. a} \quad & \phi_j \geq \mathbf{0} \end{aligned} \quad (3.24)$$

onde J'_j é a parcela de J_j em (3.16) que depende de ϕ_j , e H ($l \times l$) e f_j ($l \times 1$) são dados por

$$H = \Psi_N^T W_N \Psi_N \quad (3.25)$$

$$f_j = -\Psi_N^T W_N Y_{N_j} \quad (3.26)$$

3.3.4 Procedimento Algorítmico

1. Obtenha um conjunto de N pares de dados de entrada-saída a partir do sistema: $\{x_1(k), \dots, x_n(k), y(k)\}_{k=1}^N$.
2. Defina os conjuntos referenciais *fuzzy* para cada variável de entrada e saída.

3. Calcule o conjunto de dados *fuzzy* $\{\Psi(k), Y(k)\}_{k=1}^N$ através das equações (3.6), (3.7) e (3.8).
4. Obtenha a matriz relacional R do modelo: Se conhecimento lingüístico for requerido, então calcule essa matriz resolvendo o problema (3.24). Caso contrário, calcule-a utilizando a solução analítica (3.18) ou a solução recursiva dada pelas equações (3.21) e (3.22).

Antes de ilustrar o desempenho do algoritmo através de exemplos, aborda-se na seção seguinte o problema de interpretação lingüística dos modelos.

3.4 Interpretação de Modelos Relacionais

3.4.1 Caracterização do Problema

Na seção 3.1 viu-se que a partir da introdução do conceito de discretização *fuzzy* cada elemento de uma matriz relacional passou a ser tradicionalmente interpretado como uma medida de possibilidade de uma regra lingüística relacionando conjuntos referenciais *fuzzy* específicos das suas variáveis de entrada e saída. Nesse contexto, se \mathcal{X}_i e \mathcal{Y}_j são conjuntos referenciais *fuzzy* das variáveis de entrada x e saída y de um dado modelo³, respectivamente, então o elemento relacional $R_{i,j}$ desse modelo é interpretado como uma regra “SE x é \mathcal{X}_i ENTÃO y é \mathcal{Y}_j com Possibilidade $R_{i,j}$ ”. Sob esse ponto de vista, modelos relacionais parecem ser um tipo especial de modelo lingüístico que considera todas as possíveis regras relacionando cada um dos conjuntos referenciais *fuzzy* das suas variáveis de entrada e saída, com cada regra ponderada pelo valor do elemento relacional correspondente (Babuška e Verbruggen, 1995). No entanto, nos desenvolvimentos subseqüentes é mostrado que essa interpretação não é adequada para qualquer operador de composição *fuzzy* que possa ser adotado na implementação do modelo. Por exemplo, é mostrado que em modelos utilizando o operador de composição max-t, certos elementos relacionais podem contribuir mais com a saída do modelo (ou seja, podem ter mais influência na saída) que outros de valor igual ou maior. Pode-se então afirmar que um elemento relacional com um valor relativamente pequeno pode ser mais representativo do que outro com valor relativamente elevado e, conseqüentemente, qualquer análise individualizada (elemento-a-elemento) da matriz pode produzir resultados inadequados.

O uso do conceito de representabilidade de um elemento de matriz relacional, definido formalmente na seção seguinte, implica em um novo paradigma para a interpretação lingüística de modelos relacionais *fuzzy*, constituindo a base da metodologia proposta para a extração de conhecimento a partir desses modelos.

³Por simplicidade e sem perda de generalidade considera-se o caso monovariável.

3.4.2 Extração de Regras Lingüísticas

Seja o modelo relacional genérico em (3.1) bem como o seu vetor $\mathbf{x} = [x_1 \cdots x_n]^T$ de entradas *crisp*, sendo $\mathbf{x} \in \mathbf{X} \subset \mathfrak{R}^n$ onde $\mathbf{X} = \mathbf{X}_1 \times \cdots \times \mathbf{X}_n$ é o domínio compacto n -dimensional das variáveis de entrada.

Definição 3.1 *Dado um vetor de entradas \mathbf{x} específico, a porção da saída \hat{Y} do modelo referente ao elemento $R_{m,j}$ da matriz relacional é definida como a Contribuição \mathcal{C} de $R_{m,j}$, dada por $\mathcal{C}(R_{m,j}, \mathbf{x})$.*

A contribuição de um elemento relacional indica o quanto esse elemento contribui com a saída do modelo para um dado vetor de entradas \mathbf{x} . Dessa forma, esse conceito aplica-se a valores de entrada específicos. Uma medida de contribuição sobre todo o domínio de entrada é definida a seguir.

Definição 3.2 *A Representabilidade \mathcal{R} de um elemento $R_{m,j}$ é definida como a soma das contribuições \mathcal{C} desse elemento para todos os valores possíveis das variáveis de entrada $x_1 \in \mathbf{X}_1, \cdots, x_n \in \mathbf{X}_n$, isto é, para todo $\mathbf{x} \in \mathbf{X}$, como segue:*

$$\begin{aligned} \mathcal{R}(R_{m,j}) &= \int_{\mathbf{x} \in \mathbf{X}} \mathcal{C}(R_{m,j}, \mathbf{x}) \, d\mathbf{x} \\ &= \int_{x_1 \in \mathbf{X}_1} \int_{x_2 \in \mathbf{X}_2} \cdots \int_{x_n \in \mathbf{X}_n} \mathcal{C}(R_{m,j}, \mathbf{x}) \, dx_1 \, dx_2 \cdots dx_n \end{aligned} \quad (3.27)$$

A representabilidade de um elemento de uma matriz relacional indica o quanto esse elemento pode contribuir com a saída do modelo correspondente. No caso de problemas de modelagem incompletos, onde determinadas regiões do domínio de entrada não são representadas pelo conjunto de dados disponíveis ou pelo próprio sistema em questão (ver seção 3.3.3, item 1), a seguinte definição torna-se adequada.

Definição 3.3 *A Representabilidade Condicionada \mathcal{R}_c de um elemento $R_{m,j}$ a um conjunto de dados de entrada $\mathbb{X} = \{x_1(k), \cdots, x_n(k)\}_{k=1}^N$ é definida como a soma das contribuições \mathcal{C} desse elemento para todos os valores de entrada em \mathbb{X} , ou seja, para todo $\mathbf{x}(k)$ com $k = 1, \cdots, N$, como segue:*

$$\mathcal{R}_c(R_{m,j}) = \sum_{k=1}^N \mathcal{C}(R_{m,j}, \mathbf{x}(k)) \quad (3.28)$$

Definição 3.4 A matriz Θ contendo as medidas de representabilidade de uma matriz relacional R é definida como a Matriz de Representabilidade de R .

Proposição 3.2 De acordo com a definição 3.1 e a equação (3.4), a contribuição \mathcal{C} de cada elemento $R_{m,j}$ de um modelo com composição \max - t é dada por

$$\mathcal{C}(R_{m,j}, \mathbf{x}) = \begin{cases} \Psi_m \mathop{t} R_{m,j}, & \text{se } m = \arg \max_{i=1, \dots, l} \Psi_i \mathop{t} R_{i,j} \\ 0, & \text{caso contrário} \end{cases} \quad (3.29)$$

Exemplo A: Seja um modelo relacional monovariável (SISO) com composição \max -produto (i.e., a norma- t é o produto algébrico) e variáveis de entrada $x \in \mathbf{X}$ e saída $y \in \mathbf{Y}$ normalizadas de tal forma que $\mathbf{X} = \mathbf{Y} = [0, 1]$. As variáveis de entrada e saída possuem dois conjuntos referenciais *fuzzy* cada, rotulados lingüisticamente como “Pequeno” e “Grande”. Esses conjuntos são definidos sobre os universos de discurso \mathbf{X} e \mathbf{Y} dessas variáveis, como mostrado na figura 3.2.

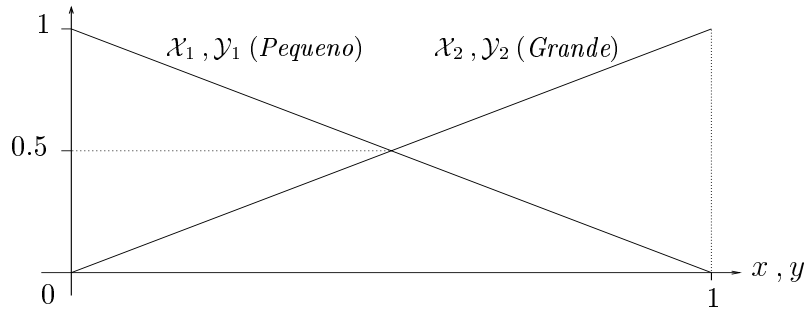


Figura 3.2: Conjuntos referenciais *fuzzy* das variáveis de entrada e saída.

Considere que a matriz relacional do modelo seja dada por

$$R = \begin{bmatrix} 0.5 & 0.5 \\ 0.1 & 0.9 \end{bmatrix} \quad (3.30)$$

Sabendo que cada elemento $R_{m,j}$ relaciona os conjuntos referenciais \mathcal{X}_m e \mathcal{Y}_j , então utilizando a abordagem convencional para a interpretação de modelos relacionais dis-

cutida na seção 3.4.1 obtém-se o seguinte conjunto de regras *fuzzy* a partir do modelo acima:

$$\begin{aligned}
 & \text{SE } x \text{ é Pequeno ENTÃO } y \text{ é Pequeno (possibilidade 0.5)} \\
 & \text{SE } x \text{ é Pequeno ENTÃO } y \text{ é Grande (possibilidade 0.5)} \\
 & \text{SE } x \text{ é Grande ENTÃO } y \text{ é Pequeno (possibilidade 0.1)} \\
 & \text{SE } x \text{ é Grande ENTÃO } y \text{ é Grande (possibilidade 0.9)}
 \end{aligned}$$

Com base nas idéias que geraram o conjunto de regras acima, as duas últimas regras poderiam ser interpretadas simplesmente como “SE x é Grande ENTÃO y é Grande”, já que a medida de possibilidade dessa regra é consideravelmente maior do que aquela da regra “concorrente” (Pedrycz, 1984a). As duas primeiras regras, por sua vez, não podem ser claramente interpretadas uma vez que elas são lingüisticamente conflitantes. A introdução de um termo lingüístico intermediário como *Médio* poderia levar à interpretação “SE x é Pequeno ENTÃO y é Médio”. Contudo, essa nova condição exigiria um novo modelo cuja matriz relacional (2×3) poderia ainda apresentar conflitos.

Esse tipo de problema pode ocorrer porque a interpretação lingüística convencional de modelos relacionais ilustrada acima analisa os elementos da matriz relacional individualmente, e não como um conjunto de elementos correlacionados. Nesse sentido, considere as medidas de contribuição e representabilidade dadas respectivamente por (3.29) e (3.27). Uma vez que o modelo possui uma única entrada com dois conjuntos *fuzzy*, então $\Psi = [\mathcal{X}_1(x) \ \mathcal{X}_2(x)]^T$ ($l = 2$). Logo, a partir das condições definidas no início do exemplo (i.e., $\mathbf{X} = [0, 1]$ e $t =$ produto algébrico), as equações (3.29) e (3.27) são rescritas como

$$\mathcal{C}(R_{m,j}, x) = \begin{cases} \mathcal{X}_m(x) R_{m,j}, & \text{se } m = \arg \max_{i=1,2} \mathcal{X}_i(x) R_{i,j} \\ 0, & \text{caso contrário} \end{cases} \quad (3.31)$$

$$\mathcal{R}(R_{m,j}) = \int_0^1 \mathcal{C}(R_{m,j}, x) dx \quad (3.32)$$

Sabendo que as funções de pertinência dos conjuntos referenciais *fuzzy* na figura 3.2 são dadas por $\mathcal{X}_1(x) = 1 - x$ e $\mathcal{X}_2(x) = x$, é simples verificar utilizando (3.30) que $\mathcal{X}_1(x)R_{1,1} \geq \mathcal{X}_2(x)R_{2,1}$ se e somente se $x \leq \frac{5}{6}$ e $\mathcal{X}_1(x)R_{1,2} \geq \mathcal{X}_2(x)R_{2,2}$ se e somente se $x \leq \frac{1}{2.8}$. Logo, da equação (3.31) tem-se que

$$\begin{aligned}
\mathcal{C}(R_{1,1}, x) &= \begin{cases} \mathcal{X}_1(x) R_{1,1}, & \text{se } x \leq \frac{5}{6} \\ 0, & \text{caso contrário} \end{cases} \\
\mathcal{C}(R_{2,1}, x) &= \begin{cases} \mathcal{X}_2(x) R_{2,1}, & \text{se } x \geq \frac{5}{6} \\ 0, & \text{caso contrário} \end{cases} \\
\mathcal{C}(R_{1,2}, x) &= \begin{cases} \mathcal{X}_1(x) R_{1,2}, & \text{se } x \leq \frac{1}{2.8} \\ 0, & \text{caso contrário} \end{cases} \\
\mathcal{C}(R_{2,2}, x) &= \begin{cases} \mathcal{X}_2(x) R_{2,2}, & \text{se } x \geq \frac{1}{2.8} \\ 0, & \text{caso contrário} \end{cases}
\end{aligned} \tag{3.33}$$

Utilizando as equações acima bem como as equações (3.30) e (3.32), a representabilidade de cada elemento da matriz relacional pode ser calculada como ilustrado abaixo para $R_{1,1}$:

$$\mathcal{R}(R_{1,1}) = \int_0^{\frac{5}{6}} (1-x) 0.5 \, dx = 0.2431 \tag{3.34}$$

As medidas de representabilidade dos outros elementos são calculadas similarmente, produzindo a seguinte matriz de representabilidade:

$$\Theta = \begin{bmatrix} \mathcal{R}(R_{1,1}) & \mathcal{R}(R_{1,2}) \\ \mathcal{R}(R_{2,1}) & \mathcal{R}(R_{2,2}) \end{bmatrix} = \begin{bmatrix} 0.2431 & 0.1467 \\ 0.0153 & 0.3926 \end{bmatrix} \tag{3.35}$$

Essa matriz mostra que, embora os elementos $R_{1,1}$ e $R_{1,2}$ em (3.30) tenham o mesmo valor, $R_{1,1}$ é significativamente mais representativo do que $R_{1,2}$.

Observação 3.1 *Com base nas constatações acima, tem-se que o conhecimento lingüístico pode ser extraído a partir do modelo através da seleção das regras referentes aos elementos mais representativos de cada linha da sua matriz relacional. Assim, a informação é extraída na forma de um conjunto consistente (não conflitante) de regras fuzzy, isto é, um conjunto desprovido de duas regras possuindo o mesmo antecedente e conseqüentes distintos.*

No exemplo anterior, as seguintes regras são extraídas a partir dessa metodologia:

SE x é Pequeno ENTÃO y é Pequeno
SE x é Grande ENTÃO y é Grande

É importante notar que o cálculo analítico das medidas de contribuição e representabilidade ilustrado no exemplo anterior (que envolve integrais múltiplas para modelos multivariáveis) pode ser complexo em problemas práticos. A complexidade depende da dimensão de entrada do modelo, da quantidade de conjuntos referenciais *fuzzy* e da forma de suas funções de pertinência bem como do operador de composição adotado. Nesses casos, uma alternativa é a utilização da medida de representabilidade condicionada definida em (3.28). Essa medida pode aproximar, com exceção de um fator proporcional, a integral em (3.27) através da soma das contribuições calculadas individualmente para um determinado conjunto de dados, sendo melhor a qualidade da aproximação quanto maior e mais homoganeamente distribuído no domínio de entrada for esse conjunto. Outra alternativa é uma seleção adequada do operador de composição *fuzzy* utilizado na equação (3.2) para a implementação do modelo. O operador média-produto, já adotado no desenvolvimento das estratégias ótimas para estimação dos modelos relacionais apresentadas na seção 3.3, é de particular interesse nesse sentido pois apresenta uma propriedade de grande utilidade discutida a seguir.

Proposição 3.3 *Com base na definição 3.1 e na equação (3.5), a contribuição \mathcal{C} de cada elemento $R_{m,j}$ de um modelo com composição média-produto é dada por*

$$\mathcal{C}(R_{m,j}, x) = \frac{1}{l} (\Psi_m R_{m,j}) \quad (3.36)$$

Observação 3.2 *Nesse caso em particular, as contribuições e conseqüentemente as representabilidades dos elementos $R_{m,j}$ de uma mesma linha m da matriz relacional são linearmente (diretamente) proporcionais aos valores desses elementos. Portanto, o conjunto de regras fuzzy pode ser extraído do modelo simplesmente selecionando a regra associada ao elemento com maior valor de cada linha da matriz relacional, sem a necessidade do cálculo da matriz de representabilidade Θ .*

Embora coincida com a abordagem convencional para a extração de regras *fuzzy* a partir de modelos relacionais discutida na seção 3.4.1 e no exemplo A, essa estratégia é aplicada naquele caso de forma equivocada (segundo o paradigma apresentado aqui) a modelos implementados com o operador de composição max-t, que não apresenta a propriedade discutida acima.

Exemplo B: Considere um modelo implementado com composição média-produto, matriz relacional dada por

$$R = \begin{bmatrix} 0.3 & 0.7 \\ 0.8 & 0.2 \end{bmatrix} \quad (3.37)$$

e as demais condições descritas no exemplo A. De acordo com a observação 3.2, as medidas de representabilidade dos elementos de cada linha da matriz relacional nesse caso são proporcionais aos valores desses elementos. Logo, os elementos $R_{1,2}$ e $R_{2,1}$ em (3.37) são os elementos mais representativos das linhas 1 e 2 da matriz, respectivamente. Lembrando que cada elemento $R_{m,j}$ relaciona os conjuntos referenciais \mathcal{X}_m e \mathcal{Y}_j (ver figura 3.2), as seguintes regras *fuzzy* podem ser extraídas do modelo acima sem qualquer cálculo de medidas de representabilidade:

SE x é *Pequeno* ENTÃO y é *Grande*
SE x é *Grande* ENTÃO y é *Pequeno*

Uma questão natural que se pode levantar diz respeito à possibilidade de inconsistências lingüísticas em termos da própria matriz de representabilidade, como por exemplo a presença de linhas nessa matriz constituindo conjuntos *fuzzy* fortemente não convexos (Zadeh, 1965) ou cujos elementos apresentam representabilidades similares. Observa-se experimentalmente (Campello e Amaral, 1999b; Campello e Amaral, 2001) que quando o modelo representa de forma adequada as características essenciais do sistema reduz-se significativamente a probabilidade de incidência de tais inconsistências. A razão é que um sistema não pode ser modelado adequadamente a partir de um conjunto inconsistente de regras. Se regras conflitantes são introduzidas no modelo, este tende a ser impreciso, envolvendo excessiva interpolação e baixa especificidade. Essa idéia é enfatizada quando o procedimento de otimização sobre os parâmetros relacionais do modelo é realizado segundo um critério *fuzzy*, como proposto na seção 3.3.2, que força o mesmo a produzir saídas *fuzzy* lingüisticamente consistentes através de uma matriz relacional também consistente.

Observação 3.3 *Teoricamente, os elementos de uma matriz relacional R devem pertencer ao intervalo de valores de pertinência fuzzy $[0,1]$. Para assegurar esta condição durante o procedimento de otimização do respectivo modelo, as restrições $\mathbf{0} \leq R \leq \mathbf{1}$ devem ser consideradas nesse procedimento. Quando a interpretação lingüística baseada na matriz de representabilidade é utilizada, no entanto, apenas as restrições de não negatividade são essenciais porque o conceito de contribuição apenas não faz sentido para valores negativos. Nesse caso, a estimação da matriz relacional pode ser realizada através da solução do problema convexo em (3.24).*

3.4.3 Simplificação de Bases de Regras

A estratégia de modelagem discutida nas seções anteriores faz com que a matriz relacional do modelo resultante forneça automaticamente informação sobre a importância (representabilidade) de cada uma das regras associadas aos seus elementos. Essa informação pode ser utilizada tanto para a obtenção de um conjunto consistente completo de regras *fuzzy*, conforme mostrado na seção anterior, como também para simplificá-lo

através da eliminação de regras “fracas”, tais como aquelas pouco ou não excitadas pelo conjunto de dados durante a fase de estimação do modelo. Essa questão é de particular interesse em problemas de natureza incompleta, como muitos sistemas de classificação. Esses sistemas possuem regiões do domínio de entrada que não são significativamente representadas por nenhuma classe. Nesse caso, as regras do respectivo modelo correspondentes a classes inexistentes não são representadas pelos dados ou são apenas residualmente (periféricamente) representadas. O mesmo ocorre em sistemas dinâmicos quando o sinal de entrada não excita adequadamente as dinâmicas do sistema, por exemplo devido a restrições operacionais, fazendo com que o conjunto de dados medidos não seja totalmente representativo.

A redução de bases de regras *fuzzy* é um problema que tem sido investigado na literatura (e.g. (Setnes, Babuška, Kaymak e van Nauta Lemke, 1998; Yen e Wang, 1999; Setnes e Babuška, 2001)). Esse problema é abordado no presente contexto utilizando os novos conceitos apresentados na seção 3.4.2. Inicialmente deve-se notar que a propriedade discutida na observação 3.2 não pode ser aplicada para comparar a representabilidade de elementos de diferentes linhas da matriz relacional. Como cada regra *fuzzy* do conjunto extraído é obtida a partir de uma linha diferente da matriz, o cálculo das respectivas representabilidades torna-se inevitável nesse caso para permitir a comparação mútua e posterior redução do conjunto através da eliminação das regras menos representativas. Esse cálculo pode ser realizado utilizando a medida de representabilidade condicionada, introduzida na definição 3.3. Em problemas de classificação nos quais os dados não estão distribuídos de maneira equilibrada entre as diferentes classes, pode-se normalizar cada medida dividindo-a pela quantidade de elementos no conjunto de dados associados à classe correspondente.

Após o cálculo da matriz de representabilidade, essa matriz deve ser normalizada no intervalo $[0,1]$ e um limiar $\xi \in [0,1]$ deve ser especificado de forma que cada regra (do conjunto previamente extraído) correspondente a um elemento relacional com representabilidade inferior a ξ seja excluída⁴. A seleção do limiar ξ constitui uma especificação de projeto. Contudo, a seleção ideal é aquela que resulta no modelo de regras mais simples porém consistente quantitativa e qualitativamente.

3.5 Exemplos

A seguir apresentam-se três exemplos para ilustrar as metodologias desenvolvidas nas seções anteriores. O primeiro é o problema lógico XOR, que ganhou destaque quando Minsky e Papert (1969) mostraram que uma rede Perceptron com uma única camada não é capaz de solucioná-lo, causando assim uma redução drástica nas pesquisas em redes neurais artificiais nos anos 70. Esse exemplo acadêmico é utilizado para ilustrar passo-a-passo os métodos propostos de estimação e extração de conhecimento lingüístico

⁴Essa abordagem sustenta-se em uma aproximação do tipo Booleana (Pedrycz, 1993; Pedrycz et al., 1995).

a partir de modelos relacionais. O segundo exemplo é uma modificação do problema XOR que permite explorar a natureza *fuzzy* dos modelos relacionais. O terceiro e último exemplo refere-se à fornalha de gás de Box e Jenkins (1970), que é um *benchmark* no contexto de identificação de sistemas dinâmicos. Trata-se de uma aplicação real utilizada para ilustrar o procedimento completo de modelagem quantitativa e qualitativa descrito nesse capítulo, incluindo a metodologia de simplificação de bases de regras.

3.5.1 Exemplo 1: Problema XOR

A lógica OU-Exclusivo (XOR) é apresentada na forma de um conjunto de dados de entrada-saída na tabela 3.1.

Tabela 3.1: Padrões lógicos XOR.

k	$x_1(k)$	$x_2(k)$	$y(k)$
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

Esses padrões constituem um problema completo de classificação *não linearmente separável*. Para abordar esse problema, utiliza-se um modelo relacional com duas entradas e uma saída, cujos conjuntos referenciais *fuzzy* são os mesmos ilustrados na figura 3.2 porém rotulados “Zero” e “Um” ao invés de “Pequeno” e “Grande”, respectivamente. Como existem dois conjuntos referenciais *fuzzy* definidos para cada variável, tem-se $c_0 = c_1 = c_2 = 2$ e $l = c_1 c_2 = 4$. Logo, utilizando esses conjuntos e as equações (3.6), (3.7) e (3.8), obtém-se os dados *fuzzy* mostrados na tabela 3.2.

Tabela 3.2: Padrões lógicos XOR (fuzzificados).

k	$\bar{X}_1^T(k)$	$\bar{X}_2^T(k)$	$\bar{\Psi}^T(k)$	$\bar{Y}^T(k)$
1	[1 0]	[1 0]	[1 0 0 0]	[1 0]
2	[1 0]	[0 1]	[0 1 0 0]	[0 1]
3	[0 1]	[1 0]	[0 0 1 0]	[0 1]
4	[0 1]	[0 1]	[0 0 0 1]	[1 0]

Utilizando a quarta e a quinta coluna da tabela 3.2 nas equações (3.19) e (3.20) obtém-se

$$Y_{N_1} = [1 \ 0 \ 0 \ 1]^T \quad (3.38)$$

$$Y_{N_2} = [0 \ 1 \ 1 \ 0]^T \quad (3.39)$$

$$\Psi_N = \frac{1}{4} I_4 \quad (3.40)$$

onde I_4 é a matriz identidade 4×4 . Substituindo Ψ_N na equação (3.18) com $W_N = I_4$ resulta $\phi_j = 4Y_{N_j}$ e:

$$R = [\phi_1 \ \phi_2] = 4 \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (3.41)$$

Tomando então as regras *fuzzy* associadas aos elementos mais representativos da matriz R , como discutido na seção 3.4.2, tem-se o seguinte conjunto:

SE x_1 é *Zero* and x_2 é *Zero* ENTÃO y é *Zero*
 SE x_1 é *Zero* and x_2 é *Um* ENTÃO y é *Um*
 SE x_1 é *Um* and x_2 é *Zero* ENTÃO y é *Um*
 SE x_1 é *Um* and x_2 é *Um* ENTÃO y é *Zero*

que constitui uma representação lingüística adequada da lógica XOR disposta na tabela 3.1. As estimativas quantitativas do modelo podem ser inferidas utilizando os dados *fuzzy* na tabela 3.2, a matriz relacional em (3.41) e as equações (3.5) e (3.11). Essas estimativas são apresentadas na tabela 3.3, onde pode-se observar que correspondem exatamente aos padrões de saída XOR originais.

Tabela 3.3: Estimativas do modelo para os padrões lógicos XOR.

k	$x_1(k)$	$x_2(k)$	$\tilde{Y}^T(k)$	$\hat{y}(k)$
1	0	0	[1 0]	0
2	0	1	[0 1]	1
3	1	0	[0 1]	1
4	1	1	[1 0]	0

3.5.2 Exemplo 2: Problema XOR com Perturbação Estocástica

Dado que os padrões lógicos XOR assumem apenas valores binários, a característica *fuzzy* dos modelos relacionais pode ser melhor explorada através de uma pequena modificação desses padrões. Para tanto, utilizam-se quatro *clusters* de dados sendo que cada *cluster* consiste de 25 padrões gerados a partir da introdução de um ruído branco Gaussiano nos dados originais da tabela 3.1, como mostrado na tabela 3.4.

Tabela 3.4: Padrões lógicos XOR com perturbação estocástica.

k	$x_1(k)$	$x_2(k)$	$y(k)$	Símbolo
1, \dots , 25	$\zeta_1(k)$	$\zeta_2(k)$	$\zeta_3(k)$	“o”
26, \dots , 50	$\zeta_1(k)$	$1+\zeta_2(k)$	$1+\zeta_3(k)$	“x”
51, \dots , 75	$1+\zeta_1(k)$	$\zeta_2(k)$	$1+\zeta_3(k)$	“+”
76, \dots , 100	$1+\zeta_1(k)$	$1+\zeta_2(k)$	$\zeta_3(k)$	“*”

Nessa tabela, $\zeta_i(k)$ ($i = 1, 2, 3$) denotam variáveis aleatórias independentes com distribuição Gaussiana de média zero e variância $\sigma^2 = 1/4$, projetadas para que os *clusters* estejam razoavelmente sobrepostos. Para fins estatísticos, dez conjuntos de dados distintos são gerados utilizando a tabela 3.4. A projeção de um desses conjuntos no domínio de entrada é ilustrada na figura 3.3 junto ao valor de contorno normalizado 0.1 das funções de densidade de probabilidade utilizadas para gerar os *clusters*, isto é, o valor correspondente a 10% do valor máximo dessas funções (associado aos centros dos *clusters*).

Utilizando os dados disponíveis e os mesmos conjuntos referenciais *fuzzy* do exemplo 1 (considerados normais à esquerda e à direita além dos seus valores extremos 0 e 1), dez modelos relacionais distintos são obtidos pela solução do problema (3.24) através de programação quadrática. Todos os modelos produzem as mesmas regras obtidas no exemplo 1, que também constituem uma representação lingüística adequada dos dados na tabela 3.4 se os rótulos dos conjuntos referenciais “Zero” e “Um” forem interpretados no contexto *fuzzy* como “Em Torno de Zero” e “Em Torno de Um”. A matriz relacional do modelo obtido a partir dos dados ilustrados na figura 3.3, por exemplo, é dada (na forma normalizada) por

$$R = \begin{bmatrix} \mathbf{0.88} & 0.24 \\ 0.30 & \mathbf{0.81} \\ 0.12 & \mathbf{1} \\ \mathbf{0.90} & 0.22 \end{bmatrix} \quad (3.42)$$

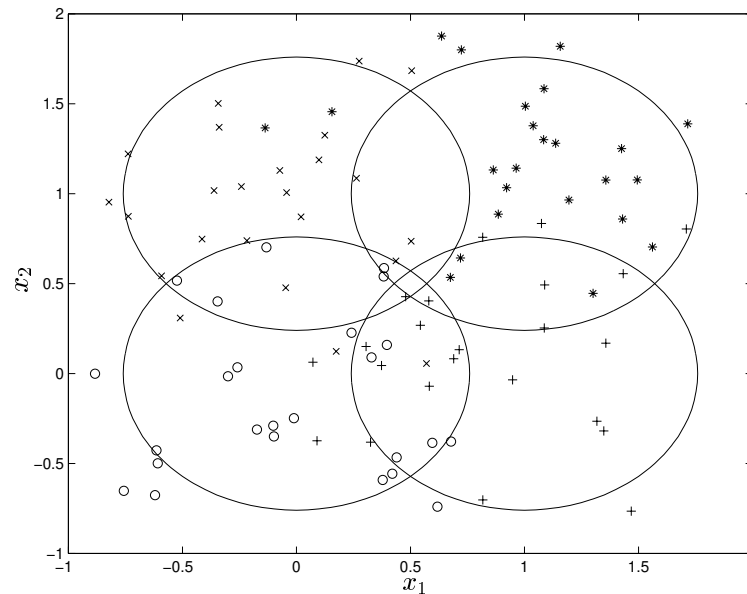


Figura 3.3: Dados estocásticos XOR (símbolos) e valor de contorno 10% das funções de densidade de probabilidade originais. (círculos).

Esse exemplo ilustra que os modelos gerados a partir das metodologias propostas são capazes de detectar as principais características do sistema e extrair as relações que descrevem o seu comportamento fundamental.

3.5.3 Exemplo 3: Fornalha de Gás Box-Jenkins

Nos exemplos anteriores, a quantidade e o posicionamento dos conjuntos referenciais *fuzzy* foram especificados de forma estratégica em relação aos padrões de dados a serem modelados. Em muitos problemas práticos, no entanto, essa escolha estratégica não é possível. Esse é o caso do presente exemplo, que diz respeito à modelagem da fornalha de gás apresentada em (Box e Jenkins, 1970).

O sistema consiste de uma fornalha em que ar e metano são combinados formando uma mistura de gases contendo Dióxido de Carbono (CO_2). A alimentação de ar é mantida constante, enquanto a taxa de alimentação de gás metano u (pés cúbicos por minuto) pode ser variada. O conjunto de dados disponível é composto de 296 pares de dados de entrada-saída, sendo que a saída y é a concentração de Dióxido de Carbono ($\% \text{CO}_2$) na mistura de gases e a entrada w é uma transformação linear da entrada real do sistema u , de modo que

$$u(k) = 0.60 - 0.04w(k) \quad (3.43)$$

A estrutura de modelo adotada para a identificação desse sistema é a mesma já amplamente utilizada na literatura (e.g. (Pedrycz, 1984a; Lee et al., 1994)), ou seja,

$y(k) = \mathcal{F}(y(k-1), w(k-3))$, onde \mathcal{F} representa o mapeamento não linear do modelo. Portanto, o modelo relacional utilizado para a identificação da fornalha de gás possui a seguinte forma: $Y(k) = X_1(k) \bullet X_2(k) \bullet R$, onde $x_1(k) = y(k-1)$ e $x_2(k) = w(k-3)$.

Os conjuntos referenciais *fuzzy* para a presente aplicação são apresentados na figura 3.4. Esses conjuntos foram escolhidos como tendo uma distribuição uniforme, com centros equidistantes e sobreposição em 0.5, sobre os universos de discurso das respectivas variáveis. Esse procedimento condiz com as idéias originais em lógica *fuzzy* e satisfaz as condições da estrutura de cognição discutidas no apêndice A, proporcionando boa interpretabilidade aos modelos sem qualquer esforço computacional. Embora essa distribuição não seja ótima no sentido do critério (3.12), que diz respeito aos conjuntos referenciais da variável de saída, pode-se verificar experimentalmente que trata-se de uma aproximação adequada (Campello, 1997), como será constatado na seqüência do exemplo. É importante observar ainda que os conjuntos referenciais poderiam opcionalmente ser refinados para reduzir o erro final de aproximação dos modelos. Algoritmos para a sintonia-fina de conjuntos *fuzzy* levando em conta restrições de interpretabilidade são descritos em (Oliveira, 1995; Oliveira, 1999a; Oliveira, 1999b).

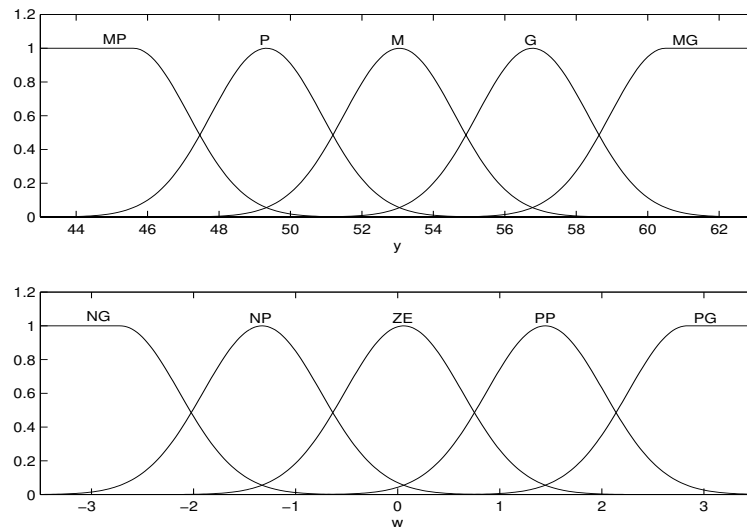


Figura 3.4: Conjuntos *Fuzzy* Gaussianos: *Muito Pequeno*, *Pequeno*, *Médio*, *Grande*, *Muito Grande* (y); *Negativo Grande*, *Negativo Pequeno*, *Zero*, *Positivo Pequeno* e *Positivo Grande* (w).

Visando a posterior extração de conhecimento lingüístico, estima-se inicialmente um modelo relacional resolvendo o problema (3.24), com $W_N = I_N$. Em seguida utiliza-se o modelo obtido para simulações de previsão 1 passo-à-frente e série sintética (i.e. simulação recursiva). As curvas de simulação do modelo são apresentadas na figura 3.5. Os erros quadráticos médios (EQMs) são respectivamente 0.1789 e 1.1387. A simulação em série sintética, na qual ocorre a realimentação dos erros de previsão da saída, contorna em certo grau a indisponibilidade de um segundo conjunto de dados para

a validação do modelo. Esse tipo de simulação também mostra que o modelo obtido pode ser utilizado em aplicações de previsão com horizontes longos, como controle preditivo (Nazzetta et al., 1998).

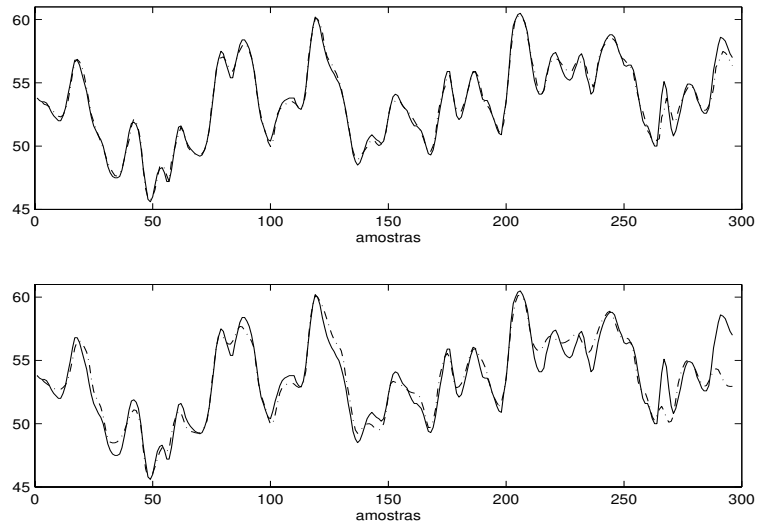


Figura 3.5: Saídas do sistema (linha cheia) e do modelo relacional (linha tracejada): previsão 1 passo-à-frente (acima) e série sintética (abaixo).

Para avaliar o efeito das restrições de interpretabilidade (não negatividade) do problema (3.24) no desempenho quantitativo dos modelos resultantes, um outro modelo relacional é estimado através da solução recursiva irrestrita dada pelas equações (3.21) e (3.22). Os respectivos erros de simulação nesse caso são $EQM = 0.1667$ em previsão 1 passo-à-frente e $EQM = 1.122$ em série sintética. Comparando esses resultados com aqueles obtidos através da solução restrita, nota-se que as restrições não afetam de maneira significativa a qualidade da solução. Por outro lado, as restrições permitem a extração de regras lingüísticas conforme discutido na seção 3.4.2⁵. O conjunto de regras extraído a partir da matriz relacional obtida como solução do problema restrito (3.24) é mostrado na figura 3.6, que também apresenta um conjunto de *trajetórias lingüísticas*. Essas trajetórias foram geradas a partir de seqüências lingüísticas crescentes e decrescentes associadas à variável de entrada w , o que corresponde ao comportamento dessa variável no tocante aos dados utilizados para a obtenção do modelo. Por exemplo, considerando uma seqüência decrescente para a entrada a partir da condição inicial “ $w(k-3)$ é PP e $y(k-1)$ é MP”, a trajetória lingüística baseada no conjunto de regras da figura 3.6 seria a seguinte:

⁵Embora a solução restrita demande, a partir de uma determinada dimensão do problema, um custo computacional maior do que aquela irrestrita obtida recursivamente via algoritmo RLS, esse não foi o caso do presente exemplo.

$$\begin{aligned}
k = i : \quad & w(k-3) \text{ é PP e } y(k-1) \text{ é MP} \rightarrow y(k) \text{ é MP} \\
k = i+1 : \quad & w(k-3) \text{ é ZE e } y(k-1) \text{ é MP} \rightarrow y(k) \text{ é P} \\
k = i+2 : \quad & w(k-3) \text{ é NP e } y(k-1) \text{ é P} \rightarrow y(k) \text{ é M} \\
k = i+3 : \quad & w(k-3) \text{ é NG e } y(k-1) \text{ é M} \rightarrow y(k) \text{ é G}
\end{aligned}$$

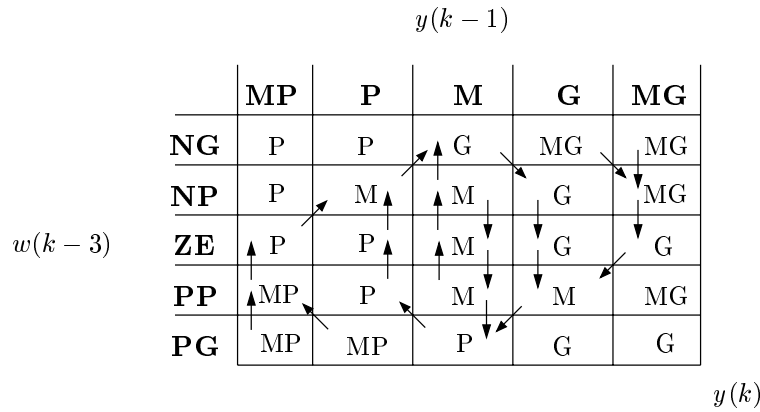


Figura 3.6: Regras *fuzzy* e trajetórias lingüísticas: Modelo completo.

As trajetórias lingüísticas mostradas na figura 3.6 ilustram o comportamento básico do sistema, ou seja, que quando a entrada w cresce a saída y decresce e vice-versa. Uma vez que w varia inversamente com a entrada real do sistema u , como mostra a equação (3.43), a relação predominante entre u e y é positiva (com atraso). Isso significa que quando a alimentação de metano aumenta, a reação na fornalha cresce (com mais metano sendo queimado) e, conseqüentemente, após o período necessário para completar a reação (o atraso de transporte) a concentração de Dióxido de Carbono também aumenta.

Para validar numericamente o conjunto de regras extraído, um modelo lingüístico é implementado a partir dessas regras em associação com o método de inferência *fuzzy* Produto-Soma-Gravidade (ver apêndice A), com discretização 0.1 dos universos de discurso. A simulação do modelo lingüístico em previsão 1 passo-à-frente é mostrada na figura 3.7, ilustrando que o conjunto de regras extraído de fato representa as dinâmicas do sistema. Esse resultado ilustra a eficiência das metodologias propostas para estimação e extração de conhecimento lingüístico a partir de modelos relacionais.

No que segue, a simplificação do conjunto de regras da figura 3.6 é considerada. Primeiramente calcula-se a matriz de representabilidade condicionada, cujos elementos são dados pela equação (3.28). Em seguida, essa matriz é normalizada no intervalo $[0,1]$ e cada regra *fuzzy* da figura 3.6 com medida de representabilidade condicionada inferior a um limiar ξ é eliminada, como discutido na seção 3.4.2. Um limiar de $\xi = 0.05$ (5%), por exemplo, reduz as 25 regras originais para as 15 regras mostradas na figura 3.8.

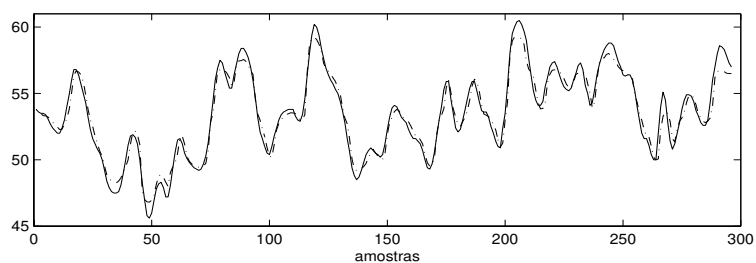


Figura 3.7: Saída do sistema (linha cheia) e do modelo lingüístico extraído (linha tracejada): EQM= 0.4325.

		$y(k - 1)$				
		MP	P	M	G	MG
	NG			G		MG
	NP		M	M	G	MG
$w(k - 3)$	ZE		P	M	G	G
	PP	MP	P	M	M	
	PG	MP				
						$y(k)$

Figura 3.8: Conjunto de regras simplificado ($\xi = 0.05$): Modelo incompleto.

Esse limiar é adequado no presente contexto por várias razões. A primeira é que ele retém a maioria das regras associadas com as trajetórias lingüísticas utilizadas para interpretar o sistema. Por outro lado, ele basicamente elimina regras extremas que de fato não são tão excitadas pelos dados como as intermediárias. Mais além, a simulação do modelo lingüístico simplificado da figura 3.8 utilizando o método Produto-Soma-Gravidade é muito similar àquela do modelo completo mostrada na figura 3.7, com erro EQM=0.5055. Esse resultado significa que o erro de modelagem não cresce de forma significativa (16.88%) frente a uma redução de 40% na dimensão do modelo. Por comparação, um limiar de $\xi = 0.1$ reduz o modelo a 8 regras, cujo erro de previsão é de EQM=0.8722.

Finalmente vale observar que, se um conjunto de regras fosse extraído a partir das medidas de representabilidade condicionada calculadas acima, esse conjunto seria precisamente o mesmo da figura 3.6. Esse resultado indica que a aproximação numérica da matriz de representabilidade através da medida de representabilidade condicionada, que foi discutida na seção 3.4.2, pode de fato ser uma alternativa interessante para a interpretação lingüística de modelos relacionais implementados com outros operadores de composição, onde a extração de regras sem o cálculo dessa matriz não é possível.

O presente exemplo, junto aos anteriores, abrange os aspectos principais das metodologias desenvolvidas e discutidas ao longo desse capítulo. Alguns aspectos específicos são abordados em outros exemplos apresentados em (Campello e Amaral, 2001; Campello e Amaral, nd). Em (Campello e Amaral, 2001), por exemplo, ilustra-se a modelagem de sistemas de regras *fuzzy* com o objetivo de quantificar a eficácia das referidas metodologias. Mostra-se que, além de aproximar com extrema exatidão os mapeamentos *fuzzy* considerados, os modelos relacionais obtidos proporcionam taxas de acerto dos conjuntos de regras extraídos com relação aos originais da ordem de 99%. Já em (Campello e Amaral, nd), o efeito do fator de esquecimento dado pela equação (3.23) na modelagem de um sistema dinâmico variante no tempo é avaliado. Mostra-se que um modelo estimado recursivamente com fator de esquecimento unitário ($\lambda = 1$), isto é, sem esquecimento, apresenta desempenho inferior àqueles estimados com $\lambda < 1$, que adaptam-se às mudanças nas dinâmicas do sistema.

3.6 Resumo e Discussões Complementares

Modelos relacionais constituem uma estratégia importante para modelagem qualitativa uma vez que o seu desenvolvimento segue uma seqüência natural baseada nas idéias originais a respeito de conjuntos e lógica *fuzzy*. Esse desenvolvimento baseia-se na estimação de relações existentes entre termos lingüísticos que são definidos por um usuário ou especialista e que representam regiões de interesse para a descrição do comportamento de um determinado sistema. Os modelos relacionais são uma alternativa aos modelos *fuzzy* baseados em regras, viz., modelos lingüísticos e modelos Takagi-Sugeno. Em relação aos primeiros, a estrutura relacional simplifica o problema de modelagem pois substitui a complexa determinação de uma base de regras lingüísticas adequada

pela estimação dos parâmetros de uma matriz relacional. Por outro lado, essa estrutura permite a inserção e extração de conhecimento lingüístico a respeito do comportamento do sistema modelado, o que não é possível utilizando a abordagem Takagi-Sugeno.

Embora os modelos relacionais *fuzzy* tenham sido amplamente estudados na literatura e reconhecidos como uma ferramenta eficiente para a identificação de sistemas complexos, apresentando capacidade de aproximação universal de mapeamentos não lineares do tipo entrada-saída, pouca atenção foi dada à interpretação lingüística desses modelos. Neste capítulo, abordou-se de forma integrada os problemas de modelagem quantitativa e qualitativa de sistemas complexos através de modelos relacionais. Foi proposta uma metodologia sistemática para o projeto estrutural dos modelos que permite uma formulação convexa para a otimização dos seus parâmetros desconhecidos. Essa formulação admite uma solução ótima global analítica que pode ser obtida numericamente através do método recursivo dos mínimos quadrados ponderados. A interpretabilidade dos modelos é considerada através da utilização de uma função objetivo *fuzzy* e da inserção de restrições convexas no problema de otimização, que nesse caso pode ser resolvido utilizando programação quadrática.

Após a etapa de estimação do modelo, conhecimento lingüístico na forma de um conjunto consistente (não conflitante) de regras *fuzzy* pode ser extraído utilizando os novos conceitos de contribuição e representabilidade que foram introduzidos como um novo paradigma para a interpretação de modelos relacionais. Esses conceitos permitem também a simplificação da base de regras extraída através da eliminação de regras menos representativas.

Ilustrou-se o desempenho das metodologias propostas através de exemplos práticos e acadêmicos. Os resultados obtidos mostram que é possível gerar modelos representativos sem necessariamente modificar os conjuntos *fuzzy* pré-definidos pelo usuário/especialista, não comprometendo assim o significado lingüístico dos mesmos. A razão principal é o efeito de ponderação das regras *fuzzy* proporcionado pelos elementos da matriz relacional, que quantitativamente pode substituir modificações na partição do domínio de entrada do problema (Ishibuchi e Nakashima, 2001). Esse mesmo efeito permite aos modelos relacionais considerarem incertezas (*fuzziness*) de forma explícita não apenas em suas variáveis de entrada, mas também em sua variável de saída, o que reduz a necessidade de incrementos na granularidade (quantidade de conjuntos referenciais) das partições *fuzzy* dessas variáveis. Com isso obtém-se um melhor compromisso entre os aspectos quantitativo e qualitativo, isto é, capacidade de aproximação e interpretabilidade, já caracterizado no próprio *princípio da incompatibilidade* estabelecido por Zadeh (1973). Não obstante, tanto a quantidade de conjuntos *fuzzy* como os conjuntos em si podem ser determinados automaticamente através de métodos como algoritmos genéticos (e.g. ver (Wong e Chen, 2000)) nos casos em que se deseja melhorar a precisão do modelo ao preço de uma possível perda de sua interpretabilidade. Esse, no entanto, não é o escopo do presente trabalho.

3.7 Contribuições e Perspectivas

As principais contribuições desse capítulo são:

1. Compilação bibliográfica detalhada a respeito de modelos relacionais dentro do contexto geral de sistemas e modelagem *fuzzy*.
2. Introdução dos conceitos de contribuição, representabilidade e representabilidade condicionada de elementos relacionais como um novo paradigma para a interpretação lingüística dos referidos modelos.
3. Desenvolvimento de uma metodologia sistemática para o projeto estrutural de modelos relacionais que permite uma formulação convexa para o problema de otimização desses modelos e a conseqüente estimação ótima global dos seus parâmetros via algoritmos de mínimos quadrados, mínimos quadrados ponderado, mínimos quadrados recursivo e programação quadrática.
4. Desenvolvimento de uma metodologia unificada para a representação e extração de conhecimento qualitativo a partir de modelos relacionais bem como para a simplificação dos modelos lingüísticos resultantes.

As principais perspectivas a partir dos resultados obtidos e apresentados no presente capítulo são:

1. Estudo aprofundado sobre as condições para inversibilidade da matriz Hessiana na solução analítica do problema de otimização dos modelos.
2. Desenvolvimento de um método para a aplicação das medidas de representabilidade na simplificação de conjuntos de regras em modelos lingüísticos não extraídos a partir de modelos relacionais.
3. Estudo comparativo entre o método proposto para a simplificação de bases de regras baseado na matriz de representabilidade e os métodos existentes baseados em transformações ortogonais (Yen e Wang, 1999; Setnes e Babuška, 2001) e medidas de similaridade (Setnes, Babuška, Kaymak e van Nauta Lemke, 1998).
4. Estudo para a verificação das condições de aplicação de algoritmos rápidos (Wellstead e Zarrop, 1991) para a solução alternativa da solução recursiva dos mínimos quadrados na presença de um número elevado de parâmetros relacionais a serem estimados.

Capítulo 4

Modelos Hierárquicos

Esse capítulo aborda o problema de dimensionalidade em sistemas *fuzzy* e neurais através de uma estrutura de modelos denominada estrutura hierárquica. Propõem-se dois novos tipos de modelos hierárquicos, cujas características são avaliadas e comparadas. Especial ênfase é dada ao problema de otimização e aos estudos sobre a capacidade de aproximação e interpretação lingüística dos modelos propostos. Exemplos práticos e acadêmicos ilustram a aplicação dos resultados teóricos obtidos.

4.1 Introdução

A capacidade de memória e processamento dos sistemas computacionais recentes tem permitido o desenvolvimento de modelos matemáticos cada vez mais eficientes (e em geral mais complexos) para a representação de sistemas não lineares estáticos e dinâmicos. Conforme discutido no capítulo 2, duas das mais importantes classes de modelos não lineares no contexto atual são os sistemas *fuzzy* e as redes neurais artificiais, especialmente em função da capacidade de aproximação universal desses modelos. No entanto, como estruturas genéricas, i.e., pouco especializadas, esses modelos usualmente requerem uma quantidade elevada de parâmetros a serem estimados. Mais especificamente, o número de parâmetros necessários para proporcionar uma dada precisão cresce exponencialmente com a dimensão do mapeamento a ser aproximado. O mesmo ocorre com a quantidade de dados necessários para a estimação desses parâmetros. Esse problema, conhecido como “Maldição da Dimensionalidade” (*Curse of Dimensionality*) (Kosko, 1997; Haykin, 1999), foi originalmente descrito por Bellman (1961).

Várias abordagens têm sido propostas na literatura com o objetivo de contornar o problema de dimensionalidade em sistemas *fuzzy* e neurais. Uma dessas abordagens é conhecida como ANOVA (Brown et al., 1995), onde a representação de um sistema n -dimensional é realizada pela soma de modelos de menor dimensão (unidimensionais, bidimensionais, etc.), i.e., uma generalização do desenvolvimento polinomial de Gabor-Kolmogorov. Em aplicações em que conjuntos desses submodelos podem ser desprezados, especialmente os de maior ordem, é possível obter uma representação mais

parcimoniosa em comparação a um modelo com estrutura convencional. O principal problema reside na seleção dos submodelos a serem utilizados para a aproximação de um determinado sistema. Em outra abordagem, Combs e Andrews (1998) propuseram uma nova configuração para a implementação de modelos de regras *fuzzy*, denominada URC (*Union-Rule Configuration*). Essa abordagem sustenta-se em um importante resultado obtido pelos autores que consiste da equivalência entre uma implicação lógica com premissa dada pela interseção de múltiplos antecedentes (denominada *Intersection-Rule Configuration* – IRC) e a união entre múltiplas implicações com antecedentes singulares (URC). Esta última evita o problema combinatorial associado à presença de múltiplos antecedentes e que resulta no crescimento exponencial da quantidade de regras (implicações) em modelos *fuzzy* convencionais, i.e., com configuração IRC. Contudo, existem ainda questões em aberto a respeito dessa estratégia, conforme discutido em (Mendel e Liang, 1999). Uma delas é até que ponto o resultado descrito acima, obtido no contexto de lógica *crisp*, pode ser generalizado para o contexto de sistemas *fuzzy*. Güven e Passino (2001) propuseram um modelo com configuração similar que também elimina o problema de explosão combinatorial da quantidade de regras, além de apresentar a propriedade de aproximação universal verificada em modelos *fuzzy* convencionais. Nesse caso, a principal dificuldade é a ausência de transparência/interpretabilidade da estrutura proposta.

Embora as estratégias citadas acima tenham propiciado avanços significativos no sentido de reduzir o problema da dimensionalidade em sistemas *fuzzy* e neurais, a abordagem mais amplamente difundida nesse contexto tem sido a utilização do conceito de hierarquia para o projeto e implementação desses sistemas. Esse conceito, que tem sido utilizado a décadas como base para o tratamento de problemas complexos e de grande porte (Mesarovic et al., 1970), permite que o problema de modelagem possa ser abordado por meio de um conjunto de submodelos de menor dimensão providos de alguma disposição hierárquica caracterizada pela presença de diferentes estágios de processamento (níveis hierárquicos). Em (Uehara e Fujise, 1993) os autores argumentam que a idéia de processamento em múltiplos estágios sustenta-se no conceito de silogismo (Zadeh, 1985). Essa é uma das estratégias do raciocínio humano, particularmente útil para lidar com problemas de grande porte e com a dificuldade para analisar simultaneamente um número elevado de variáveis (Duan e Chung, 2001). Os modelos hierárquicos inserem-se diretamente nesse contexto como arquiteturas de processamento em múltiplos estágios onde resultados preliminares (intermediários) são obtidos a partir da análise de subconjuntos das variáveis do problema e então utilizados junto a outras variáveis de forma a produzir um resultado final.

Diferentes estratégias hierárquicas têm sido desenvolvidas visando contornar o problema de dimensionalidade em sistemas *fuzzy* e neurais. Holve (1997; 1998) propôs uma arquitetura do tipo árvore composta de subsistemas *fuzzy* e a determinação de suas regras lingüísticas através de um algoritmo heurístico. Conforme discutido nos referidos trabalhos, a principal dificuldade é a elevada quantidade de possíveis alternativas estruturais gerada pela arquitetura proposta. Problema similar ocorre nas abor-

dagens desenvolvidas em (Shimojima et al., 1995; Furuhashi et al., 1997; Tachibana e Furuhashi, 1998), onde adotam-se arquiteturas ainda mais flexíveis que requerem a utilização de ferramentas auxiliares como Algoritmos Genéticos (*Genetic Algorithms* – GAs) (Mitchell, 1998; Von Zuben, 2001) para a seleção de configurações estruturais adequadas. Uma forma de simplificar significativamente esse problema é através da adoção de arquiteturas estruturadas, onde apenas a ordem hierárquica das variáveis de entrada deve ser selecionada. Esse é o tema central dos desenvolvimentos subseqüentes do presente capítulo.

Em um trabalho pioneiro nesse contexto, focado na solução do problema de dimensionalidade em sistemas de controle *fuzzy*, Raju et al. (1991) propuseram um controlador com estrutura hierárquica constituída de um conjunto de subcontroladores conectados em cascata. A estrutura proposta apresenta a propriedade de que o número de regras *fuzzy* cresce linearmente (e não exponencialmente) com a dimensão de entrada, viabilizando a utilização dos respectivos controladores em problemas de grande porte (Jamshidi, 1997). Wang (Wang, 1998; Chen e Wang, 2000) analisou a aplicação dessa mesma estrutura em problemas de modelagem. Em sua abordagem, Wang implementou os subsistemas hierárquicos utilizando uma modificação dos modelos Takagi-Sugeno e mostrou que o modelo resultante é um aproximador universal. Sua estratégia, contudo, baseada na construção analítica do modelo, assume disponível o valor de saída do mapeamento a ser aproximado correspondente a qualquer valor no seu domínio. Essa hipótese é muito restritiva para a grande maioria das aplicações, especialmente no contexto de identificação de sistemas dinâmicos. Para contornar esse problema, Wang (1999) propôs a utilização de métodos de otimização baseados em gradiente para a estimação dos parâmetros do modelo a partir de um conjunto finito de dados. No trabalho referido acima, apenas modelos com três entradas foram considerados. Abordagem similar foi proposta em (Chung e Duan, 2000). Nesse caso os autores implementaram os subsistemas hierárquicos utilizando modelos *fuzzy* baseados em regras, sendo esses parametrizados como redes neurais.

Paralelamente aos desenvolvimentos acima, uma outra proposta de modelo baseada na mesma estrutura hierárquica com disposição em cascata foi apresentada por Campello e Amaral (1999c; 2000b). Essa proposta constitui o tema central da seção 4.2 e caracteriza-se pela implementação dos subsistemas hierárquicos utilizando uma simplificação dos modelos relacionais discutidos no capítulo 3. Esses modelos, denominados modelos relacionais simplificados (Oliveira e Lemos, 1997), são sob certas condições completamente equivalentes a redes neurais do tipo RBF (Campello e Amaral, 1999a), conforme discutido no apêndice A. Mesmo nessas condições, sua formulação é particularmente adequada pois trata separadamente cada variável de entrada no seu respectivo domínio unidimensional. Com a utilização dessa formulação para a descrição dos subsistemas hierárquicos, obtém-se um conjunto fechado de equações recursivas para o cálculo do gradiente da saída do modelo em função dos seus parâmetros, isto é, um conjunto de equações genéricas para modelos hierárquicos com qualquer quantidade de entradas. Essas equações permitem o cálculo do gradiente de um dado critério de de-

sempenho a ser minimizado em relação a esses parâmetros e a conseqüente otimização dos modelos a partir de métodos clássicos de otimização irrestrita, tais como gradiente de maior descida (*steepest descent gradient*) e gradiente conjugado de Fletcher-Reeves (Bazaraa et al., 1993). Esse último é de particular interesse no presente contexto pois é adequado a problemas de grande porte uma vez que não requer o cálculo/armazenagem de qualquer matriz Hessiana ou sua inversa, possuindo ainda garantia de convergência com taxa de segunda ordem. A superioridade desse método com relação ao método do gradiente de maior descida, que possui taxa de convergência de primeira ordem, é ilustrada em (Campello e Amaral, 1999c) no contexto de modelos hierárquicos. Com exceção de situações específicas em que métodos de otimização restrita serão necessários, o algoritmo do gradiente conjugado de Fletcher-Reeves será adotado ao longo desse capítulo para a otimização dos modelos considerados.

Como será visto ao longo da seção 4.2, os modelos relacionais hierárquicos discutidos acima apresentam desempenho adequado na modelagem de diferentes tipos de sistemas. O principal problema com esses modelos é que, assim como aqueles descritos anteriormente, não proporcionam conhecimento qualitativo a respeito dos sistemas modelados em decorrência da perda de interpretabilidade intrínseca das arquiteturas hierárquicas. Para contornar esse problema, apresenta-se na seção 4.3 uma classe alternativa de modelos relacionais hierárquicos que permite a extração de regras *fuzzy* através da mesma metodologia discutida no capítulo 3 no contexto de modelos não hierárquicos e mostra-se na seção 4.4 que os modelos correspondentes apresentam a propriedade de aproximação universal.

4.2 Modelos Hierárquicos com Submodelos Relacionais Simplificados

4.2.1 Modelos Relacionais Simplificados e Redes Neurais RBF

Considere o seguinte sistema¹:

$$y = \mathcal{F}(x_1, \dots, x_n) \quad (4.1)$$

onde \mathcal{F} é um operador não linear que realiza o mapeamento das entradas x_i ($i = 1, \dots, n$) na saída y . Esse sistema pode ser representado por um modelo relacional *fuzzy* simplificado (Oliveira e Lemos, 1997) através da seguinte equação:

$$\hat{y} = \Psi^T \Omega \quad (4.2)$$

¹Por simplicidade e sem perda de generalidade MISO, já que um sistema MIMO pode ser sempre representado por um conjunto de equações desse tipo.

onde \hat{y} é a saída do modelo, $\Omega = [\Omega_1 \ \cdots \ \Omega_M]^T$ é o vetor de parâmetros relacionais e $\Psi = [\Psi_1 \ \cdots \ \Psi_M]^T$ é o vetor de entrada *fuzzy*. A diferença básica desses modelos em comparação aos modelos relacionais convencionais discutidos no capítulo 3 é que eles produzem a saída *crisp* \hat{y} diretamente a partir da inferência com um conjunto de entradas, e não a partir da defuzzificação de um conjunto *fuzzy* resultante dessa inferência. Assim como para os modelos relacionais convencionais, o vetor Ψ é dado pelo produto de Kronecker (\otimes) das entradas *fuzzy* individuais, isto é,

$$\Psi = X_1 \otimes X_2 \otimes \cdots \otimes X_n \quad (4.3)$$

que por sua vez são obtidas a partir das entradas *crisp* x_i como

$$X_i = [\mathcal{X}_{i_1}(x_i) \ \mathcal{X}_{i_2}(x_i) \ \cdots \ \mathcal{X}_{i_c}(x_i)]^T \quad (4.4)$$

onde \mathcal{X}_{i_j} ($j = 1, \dots, c$) é o j -ésimo conjunto referencial *fuzzy* da i -ésima variável de entrada e c é o número de conjuntos referenciais definidos para cada variável (em princípio assumido ser único). Demonstra-se no apêndice A que quando esses conjuntos são Gaussianos o modelo relacional dado pelas equações acima é matematicamente equivalente a uma rede neural do tipo RBF. A analogia entre esses modelos é ilustrada nas figuras 4.1 e 4.2 para o caso de duas entradas. A figura 4.1 mostra conjuntos referenciais *fuzzy* Gaussianos definidos nos domínios unidimensionais (universos de discurso) das variáveis de entrada x_1 e x_2 de um modelo relacional simplificado. A figura 4.2 mostra as funções de ativação da rede RBF equivalente, cujas projeções produzem os conjuntos referenciais da figura 4.1.

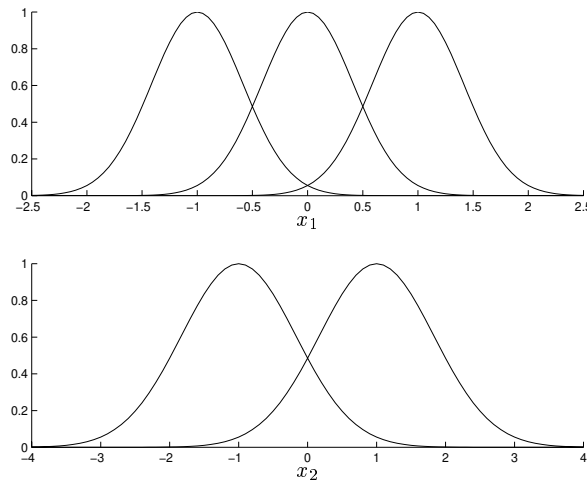


Figura 4.1: Conjuntos referenciais de um modelo *fuzzy* com 2 entradas: $\mathcal{X}_{1_i}(x_1)$, $i = 1, 2, 3$ (acima) ; $\mathcal{X}_{2_i}(x_2)$ $i = 1, 2$ (abaixo).

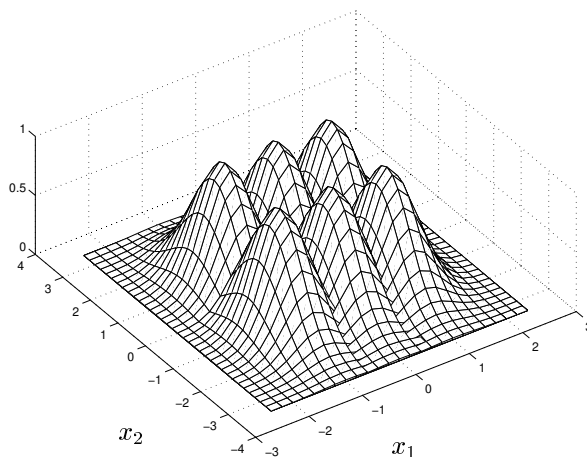


Figura 4.2: Funções de ativação (6 neurônios) da rede neural RBF equivalente.

O modelo dado pelas equações (4.2), (4.3) e (4.4) segue a estrutura convencional de modelos *fuzzy* e redes neurais *feedforward* ilustrada na figura 4.3. O principal problema com essa estrutura é discutido a seguir.

Pode-se verificar a partir das equações (4.3) e (4.4) que o número de elementos dos vetores Ψ e Ω em (4.2) é dado por $M = c^n$, sendo a capacidade de aproximação do modelo proporcional a c . A quantidade M representa o número de regras *fuzzy* associadas ao modelo ou, alternativamente, o número de neurônios da rede RBF equivalente. Esse valor também é o número de parâmetros do modelo a serem estimados (pesos sinápticos na rede RBF) caso os conjuntos referenciais *fuzzy*/funções de ativação sejam mantidos fixos, i.e., pré-especificados. Por outro lado, se os centros e aberturas dos conjuntos referenciais *fuzzy*/funções de ativação forem considerados parâmetros de projeto livre, então a quantidade total de parâmetros do modelo a serem estimados torna-se $\mu = c^n + 2nc$.

A relação exponencial entre o número de entradas n e o número de regras *fuzzy*/neurônios M do modelo é apresentada na figura 4.4 para valores típicos de c . Essa figura ilustra o problema de dimensionalidade em modelos não hierárquicos, ou seja, o crescimento da quantidade de regras/neurônios necessários para cobrir o domínio de entrada com determinada “densidade” como uma função exponencial da dimensão desse domínio.

4.2.2 Estrutura Hierárquica Padrão

Uma alternativa para contornar o problema de dimensionalidade discutido na seção anterior é a estrutura hierárquica mostrada na figura 4.5, onde $n - 1$ submodelos com domínios de entrada bidimensionais são conectados em cascata. Como cada submodelo (também denominado unidade de processamento ou bloco hierárquico) possui apenas duas entradas, o número de regras/neurônios por unidade é c^2 . Conseqüen-

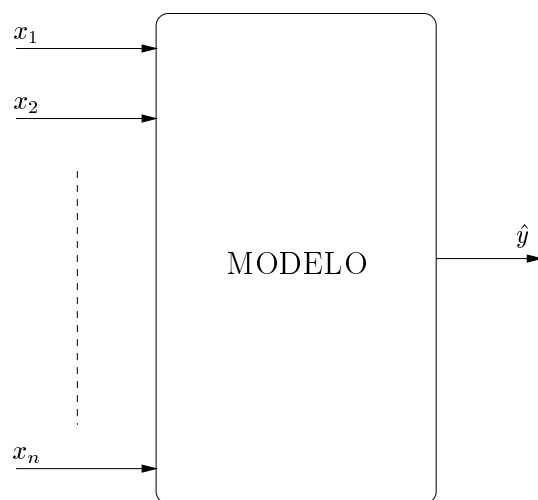
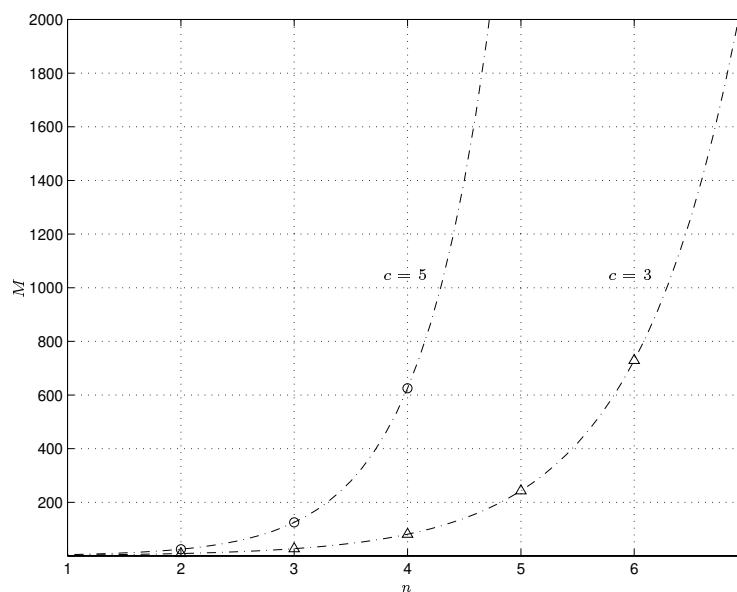


Figura 4.3: Estrutura não hierárquica.

Figura 4.4: Relação entre o número de entradas n e o número de regras *fuzzy*/neurônios M em modelos não hierárquicos (valores típicos de c).

temente, o número total de regras/neurônios do modelo hierárquico é $M = (n - 1)c^2$ ($n \geq 2$). Esse valor também é o número de parâmetros a serem estimados se os conjuntos *fuzzy*/funções de ativação forem mantidos fixos, i.e., pré-especificados. Caso contrário, a quantidade total de parâmetros de projeto livre do modelo torna-se $\mu = (n - 1)c^2 + 2nc + 2(n - 2)c = (n - 1)c^2 + 4(n - 1)c$.

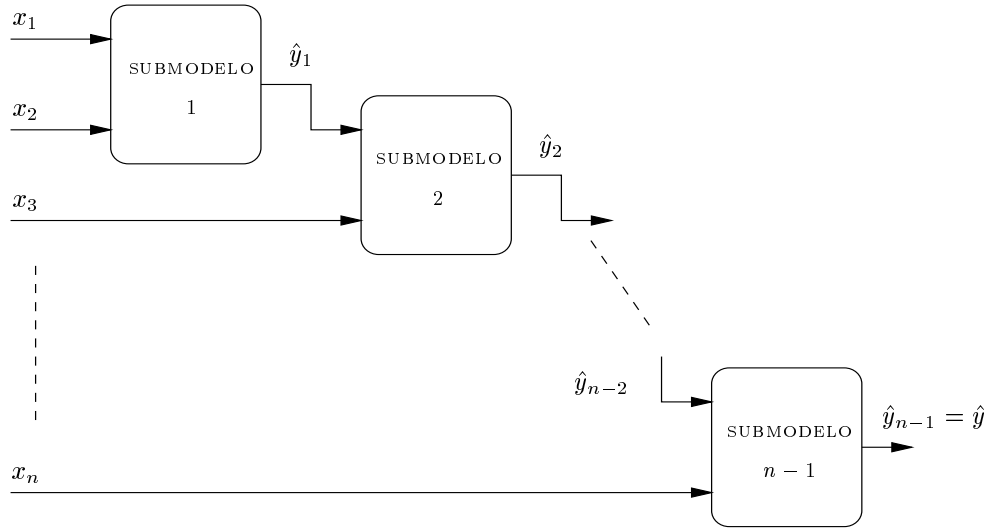


Figura 4.5: Estrutura Hierárquica.

A relação entre o número de entradas n e o número de regras/neurônios M do modelo hierárquico é apresentada na figura 4.6 para valores típicos de c . Essa figura mostra que a taxa de crescimento no número de regras/neurônios em função do número de entradas é constante (para $n \geq 2$). Essa é uma vantagem significativa em comparação ao comportamento da estrutura não hierárquica, ilustrado na figura 4.4.

Com base na formulação apresentada na seção 4.2.1, as equações que descrevem o modelo mostrado na figura 4.5 são dadas por:

$$\hat{y}_i = \Psi_i^T \Omega_i = \sum_{l=1}^{c^2} \Psi_{il} \Omega_{il}, \quad i = 1, \dots, n-1 \quad (4.5)$$

$$\Psi_i = [\Psi_{i1} \ \dots \ \Psi_{ic^2}]^T = \begin{cases} X_{i+1} \otimes \hat{Y}_{i-1}, & i = 2, \dots, n-1 \\ X_{i+1} \otimes X_i, & i = 1 \end{cases} \quad (4.6)$$

$$X_j = [\mathcal{X}_{j1}(x_j) \ \mathcal{X}_{j2}(x_j) \ \dots \ \mathcal{X}_{jc}(x_j)]^T, \quad j = 1, \dots, n \quad (4.7)$$

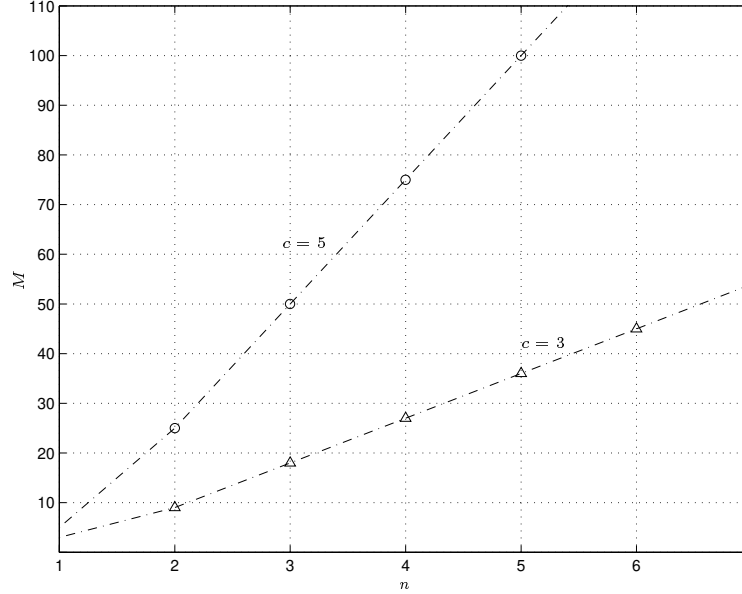


Figura 4.6: Relação entre o número de entradas n e o número de regras *fuzzy*/neurônios M em modelos hierárquicos (valores típicos de c).

$$\hat{Y}_h = [\mathcal{Y}_{h_1}(\hat{y}_h) \ \mathcal{Y}_{h_2}(\hat{y}_h) \ \cdots \ \mathcal{Y}_{h_c}(\hat{y}_h)]^T, \quad h = 1, \dots, n-2 \quad (4.8)$$

onde $\Omega_i = [\Omega_{i_1} \cdots \Omega_{i_{c_2}}]^T$ é o vetor de parâmetros relacionais/pesos sinápticos do i -ésimo bloco hierárquico com vetor *fuzzy* de entrada Ψ_i e saída \hat{y}_i , e $\mathcal{X}_{(\cdot)}$ e $\mathcal{Y}_{(\cdot)}$ são os conjuntos referenciais *fuzzy* associados às entradas originais $x_{(\cdot)}$ e artificiais $\hat{y}_{(\cdot)}$ (saídas intermediárias) do modelo. No caso de conjuntos referenciais Gaussianos tem-se

$$\mathcal{X}_{j_i}(x_j) = \exp\left(-\frac{(x_j - \theta_{j_i})^2}{\sigma_{j_i}^2}\right) \quad (4.9)$$

$$\mathcal{Y}_{h_l}(\hat{y}_h) = \exp\left(-\frac{(\hat{y}_h - \phi_{h_l})^2}{\varphi_{h_l}^2}\right) \quad (4.10)$$

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

4.2.3 Problema de Otimização

Considere um conjunto de N pares de dados de entrada-saída, $\{x_1(k), \dots, x_n(k), y(k)\}_{k=1}^N$, amostrados a partir de um sistema a ser modelado.

Um modelo hierárquico desse sistema pode ser estimado através da solução do seguinte problema de otimização:

$$\min_{\Gamma} J = \frac{1}{2} \sum_{k=1}^N (y(k) - \hat{y}(k))^2 \quad (4.11)$$

onde Γ denota o conjunto de todos os parâmetros de projeto livre do modelo. O problema (4.11) pode ser resolvido através de técnicas de otimização irrestrita, como o algoritmo do gradiente conjugado de Fletcher-Reeves (Bazaraa et al., 1993). Essas técnicas requerem o cálculo do gradiente da função de custo J em relação ao conjunto de parâmetros Γ . O conjunto Γ associado aos modelos hierárquicos em questão é constituído pelos vetores de parâmetros $\Omega_{(\cdot)}$ bem como pelos centros ($\theta_{(\cdot)}$ e $\phi_{(\cdot)}$) e aberturas ($\sigma_{(\cdot)}$ e $\varphi_{(\cdot)}$) dos conjuntos referenciais *fuzzy* de cada submodelo. Tomando as derivadas de J em (4.11) com relação a cada um desses parâmetros através da aplicação da regra da cadeia nas equações (4.5), (4.6), (4.7) e (4.8) tem-se² (Campello e Amaral, 1999c):

$$\frac{\partial J}{\partial \Omega_{h_i}} = - \sum_{k=1}^N e(k) \Psi_{h_i}(k) \lambda_h(k), \quad \begin{array}{l} i = 1, \dots, c^2 \\ h = 1, \dots, n - 1 \end{array} \quad (4.12)$$

$$\frac{\partial J}{\partial \phi_{h_i}} = - \sum_{k=1}^N \left[e(k) \lambda_{h+1}(k) \left(\sum_{l=0}^{c-1} \Omega_{(h+1)(lc+i)} X_{(h+2)(l+1)}(k) \right) \frac{\partial \mathcal{Y}_{h_i}(\hat{y}_h(k))}{\partial \phi_{h_i}} \right], \quad (4.13)$$

$$\begin{array}{l} i = 1, \dots, c \\ h = 1, \dots, n - 2 \end{array}$$

$$\frac{\partial J}{\partial \theta_{h_i}} = - \sum_{k=1}^N \left[e(k) \lambda_{h-1}(k) \left(\sum_{j=1}^c \Omega_{(h-1)(ic-c+j)} \hat{Y}_{(h-2)_j}(k) \right) \frac{\partial \mathcal{X}_{h_i}(x_h(k))}{\partial \theta_{h_i}} \right], \quad (4.14)$$

$$\begin{array}{l} i = 1, \dots, c \\ h = 3, \dots, n \end{array}$$

$$\frac{\partial J}{\partial \theta_{2_i}} = - \sum_{k=1}^N \left[e(k) \lambda_1(k) \left(\sum_{j=1}^c \Omega_{1(ic-c+j)} X_{1_j}(k) \right) \frac{\partial \mathcal{X}_{2_i}(x_2(k))}{\partial \theta_{2_i}} \right], \quad i = 1, \dots, c \quad (4.15)$$

²O procedimento é o mesmo utilizado para calcular as equações dos gradientes locais em redes neurais do tipo Perceptron multi-camadas (MLP) (Haykin, 1999) (*mutatis mutandis*).

$$\frac{\partial J}{\partial \theta_{1_i}} = - \sum_{k=1}^N \left[e(k) \lambda_1(k) \left(\sum_{l=0}^{c-1} \Omega_{1(lc+i)} X_{2(l+1)}(k) \right) \frac{\partial \mathcal{X}_{1_i}(x_1(k))}{\partial \theta_{1_i}} \right], \quad i = 1, \dots, c \quad (4.16)$$

onde $X_{h_i}(k) \triangleq \mathcal{X}_{h_i}(x_h(k))$, $\hat{Y}_{h_i}(k) \triangleq \mathcal{Y}_{h_i}(\hat{y}_h(k))$ e $e(k) \triangleq y(k) - \hat{y}(k)$. O termo $\lambda_{(\cdot)}(k)$ pode ser escrito recursivamente como

$$\lambda_q(k) = \lambda_{q+1}(k) \sum_{j=1}^c \left[\left(\sum_{l=0}^{c-1} \Omega_{(q+1)(lc+j)} X_{(q+2)(l+1)}(k) \right) \frac{\partial \mathcal{Y}_{q_j}(\hat{y}_q(k))}{\partial \hat{y}_q(k)} \right] \quad (4.17)$$

sendo $q = 1, \dots, n-2$ e $\lambda_{n-1}(k) = 1$. Essa recursividade é análoga ao resultado das redes MLP onde o gradiente associado aos pesos sinápticos de uma dada camada de neurônios é obtido em função daquele já calculado para os neurônios da camada subsequente (*Back-propagation*).

As derivadas associadas às aberturas dos conjuntos referenciais *fuzzy* podem ser obtidas a partir das equações (4.13), (4.14), (4.15) e (4.16), simplesmente substituindo $\theta_{(\cdot)}$ e $\phi_{(\cdot)}$ por $\sigma_{(\cdot)}$ e $\varphi_{(\cdot)}$, respectivamente. Os termos derivativos apresentados implicitamente nessas equações podem ser calculados numericamente, independente do tipo dos conjuntos referenciais adotados, ou analiticamente, para um tipo específico. No caso de conjuntos com funções de pertinência Gaussianas, por exemplo, os referidos termos são rescritos explicitamente a partir de (4.9) e (4.10) como segue:

$$\frac{\partial \mathcal{Y}_{h_i}(\hat{y}_h(k))}{\partial \phi_{h_i}} = \frac{2}{\varphi_{h_i}^2} (\hat{y}_h(k) - \phi_{h_i}) \mathcal{Y}_{h_i}(\hat{y}_h(k)) \quad (4.18)$$

$$\frac{\partial \mathcal{Y}_{h_i}(\hat{y}_h(k))}{\partial \varphi_{h_i}} = \frac{2}{\varphi_{h_i}^3} (\hat{y}_h(k) - \phi_{h_i})^2 \mathcal{Y}_{h_i}(\hat{y}_h(k)) \quad (4.19)$$

$$\frac{\partial \mathcal{X}_{h_i}(x_h(k))}{\partial \theta_{h_i}} = \frac{2}{\sigma_{h_i}^2} (x_h(k) - \theta_{h_i}) \mathcal{X}_{h_i}(x_h(k)) \quad (4.20)$$

$$\frac{\partial \mathcal{X}_{h_i}(x_h(k))}{\partial \sigma_{h_i}} = \frac{2}{\sigma_{h_i}^3} (x_h(k) - \theta_{h_i})^2 \mathcal{X}_{h_i}(x_h(k)) \quad (4.21)$$

$$\frac{\partial \mathcal{Y}_{q_j}(\hat{y}_q(k))}{\partial \hat{y}_q(k)} = \frac{-2}{\varphi_{q_j}^2} (\hat{y}_q(k) - \phi_{q_j}) \mathcal{Y}_{q_j}(\hat{y}_q(k)) \quad (4.22)$$

Observação 4.1 *O conjunto de equações de (4.12) a (4.17) foi obtido considerando que o número de conjuntos referenciais fuzzy por variável de entrada é o mesmo para cada variável, e igual a c . Essa condição simplifica significativamente o conjunto de equações e o seu desenvolvimento. Se um número distinto de conjuntos referenciais for especificado para cada variável, de modo a atribuir níveis de granularidade e precisão diferentes a cada uma delas, o mesmo conjunto de equações pode ser utilizado considerando c como sendo o maior valor entre as diferentes quantidades de conjuntos atribuídos às variáveis. Deve-se então retirar dos parâmetros a serem otimizados os centros e aberturas referentes aos conjuntos referenciais excedentes bem como os respectivos elementos relacionais. Esses últimos devem ainda ser igualados a zero para anular as contribuições referentes aos conjuntos referenciais descartados nos cálculos da saída do modelo e das derivadas relacionadas aos demais parâmetros.*

4.2.4 Normalização de Dados e Inicialização de Parâmetros

Sabe-se que, por questões de robustez numérica dos algoritmos de otimização, é recomendável que as variáveis dos modelos sejam normalizadas para que os dados de entrada e saída (E/S) disponíveis para identificação estejam representados em um intervalo comum, como por exemplo $[-1, 1]$. Considerando um sistema com entrada x e saída y , normalizadas através das transformações afins $f_1(x) = a_1 + b_1x$ e $f_2(y) = a_2 + b_2y$, respectivamente, tem-se o esquema de modelagem com normalização ilustrado na figura 4.7.



Figura 4.7: Modelo com normalização de E/S.

Os conjuntos referenciais *fuzzy* do modelo devem ser inicializados sobre os universos de discurso das respectivas variáveis, que nesse caso constituem o próprio intervalo de normalização. Uma maneira simples e que em geral proporciona resultados satisfatórios é inicializar esses conjuntos através de uma distribuição homogênea dentro do intervalo de normalização (Güven e Passino, 2001), com centros igualmente espaçados e aberturas, por exemplo, iguais à distância entre dois centros consecutivos, como ilustrado na figura 4.8 para $c = 5$ conjuntos Gaussianos. Essa distribuição será adotada nos desenvolvimentos subsequentes a não ser que o contrário seja estabelecido.

Conforme indica a figura 4.8, essa distribuição também é válida para os conjuntos referenciais *fuzzy* $\mathcal{Y}_{(\cdot)}$ correspondentes às variáveis intermediárias do modelo, que podem

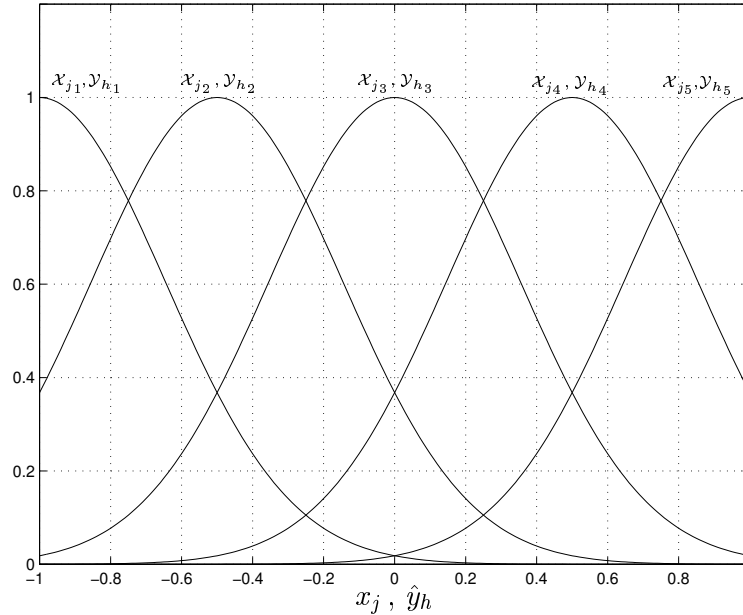


Figura 4.8: Distribuição homogênea de conjuntos referenciais *fuzzy*.

ser arbitrados no mesmo intervalo de normalização adotado. Logo, uma inicialização adequada para os elementos relacionais (vetores $\Omega_{(\cdot)}$) pode ser obtida através de valores aleatórios com média zero e variância pequena o suficiente para que as saídas dos submodelos estejam ao menos aproximadamente contidas naquele intervalo, evitando assim a saturação dos conjuntos referenciais associados a essas variáveis.

É importante notar que na modelagem de sistemas estáticos as variáveis de E/S são independentes e, portanto, normalizadas com fatores individuais. Na modelagem de sistemas dinâmicos as variáveis envolvidas não são necessariamente independentes pois podem estar associadas a uma mesma entrada ou saída do sistema, porém em instantes de amostragem distintos. Nesse caso, normalizam-se os sinais do sistema e, portanto, cada variável do modelo referente a uma mesma entrada ou saída desse sistema é normalizada utilizando os mesmos fatores. Essa condição não altera o fato de que essas variáveis são normalizadas em um intervalo comum e, portanto, em ambos os casos os parâmetros do modelo podem ser inicializados da mesma forma discutida acima.

4.2.5 Seleção da Ordem Hierárquica

A seguir realiza-se uma análise sobre a influência da disposição (ordem) das variáveis de entrada nos modelos com estrutura hierárquica. Essa análise baseia-se nos resultados obtidos em dois casos de estudo com características distintas e de particular interesse para as investigações em questão.

1º Caso de Estudo: Sistema Dinâmico

Seja o seguinte sistema dinâmico não linear (Narendra e Parthasarathy, 1990; Wang, 1994):

$$y(k) = 0.3y(k-1) + 0.6y(k-2) + 0.6\sin(\pi u(k-1)) + 0.3\sin(3\pi u(k-1)) + 0.1\sin(5\pi u(k-1)) \quad (4.23)$$

O sistema acima é um *benchmark* na área de modelagem e controle de sistemas dinâmicos utilizando lógica *fuzzy* e RNAs e é de particular interesse no presente contexto porque as não linearidades concentram-se apenas na variável de entrada independente u , isto é, a complexidade da dinâmica restringe-se ao mapeamento entre $u(k-1)$ e a saída $y(k)$.

Seja uma arquitetura em cascata para representar o sistema (4.23) utilizando as três entradas discretas $u(k-1)$, $y(k-1)$ e $y(k-2)$, possuindo portanto dois blocos hierárquicos. Nesse exemplo avalia-se e compara-se o desempenho de treinamento e previsão dos modelos obtidos quando a entrada $u(k-1)$, onde concentram-se as não linearidades, é disposta em cada um desses blocos. A arquitetura básica considerada é a ilustrada na figura 4.9, sendo uma configuração hierárquica caracterizada por $x_1(k) = u(k-1)$, $x_2(k) = y(k-1)$ e $x_3(k) = y(k-2)$ e a outra caracterizada por $x_1(k) = y(k-1)$, $x_2(k) = y(k-2)$ e $x_3(k) = u(k-1)$.

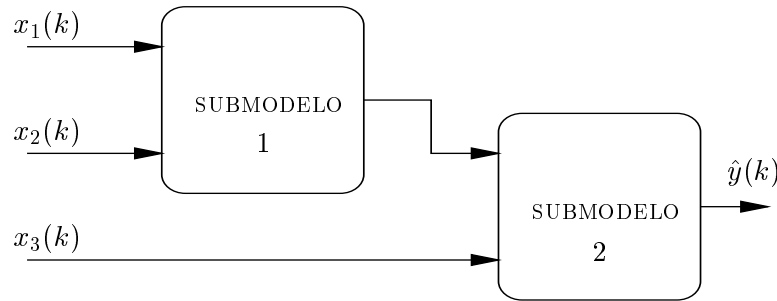


Figura 4.9: Modelo hierárquico com 3 entradas.

Dois conjuntos independentes de dados (treinamento e validação) são gerados a partir da equação que descreve o sistema. Os dados de treinamento são obtidos utilizando o mesmo sinal de excitação adotado em (Narendra e Parthasarathy, 1990; Wang, 1994), isto é, $u(k) = \sin\left(\frac{2\pi k}{250}\right)$. Toma-se um conjunto de 260 pares de dados de entrada-saída $\{u(k), y(k)\}_{k=1}^{260}$, sendo os dez primeiros referentes ao transitório inicial (condições iniciais nulas) e os demais 250 referentes a um período completo do sinal de excitação. Os dados de validação são gerados com $u(k) = \sin\left(\frac{w\pi k}{250}\right)$, onde $k = 1, \dots, 500$ e w é um escalar que varia linearmente no tempo de 1 ($k = 1$) até 4 ($k = 500$), i.e., da metade até o dobro do valor referente aos dados de estimação.

Considera-se inicialmente a primeira configuração hierárquica. Os parâmetros de um modelo com $c = 5$ conjuntos referenciais *fuzzy* Gaussianos por variável de entrada são inicializados conforme discutido na seção 4.2.4. Para fins estatísticos, utilizam-se cinco inicializações distintas para os parâmetros relacionais. O critério de parada utilizado para o procedimento de treinamento dos modelos correspondentes a cada uma dessas inicializações é $EQM < \epsilon$, onde EQM é o erro quadrático médio entre as saídas do modelo e do sistema ($EQM = \frac{2J}{N}$) e $\epsilon = 0.005$ é um limiar de erro pequeno o suficiente para os propósitos da presente análise. A evolução do treinamento é ilustrada na figura 4.10 e os resultados finais são apresentados na tabela 4.1.

Tabela 4.1: Desempenhos de treinamento e simulação de um modelo hierárquico otimizado a partir de cinco inicializações distintas. Hierarquia: $x_1(k) = u(k - 1)$, $x_2(k) = y(k - 1)$ e $x_3(k) = y(k - 2)$.

Inicialização	Épocas	EQM previsão 1 passo	EQM série sintética
1	173	0.0298	0.1765
2	128	0.0738	0.2894
3	234	0.1636	2.8143
4	196	0.0840	0.4077
5	120	0.0438	0.2435

As colunas da esquerda para a direita na tabela 4.1 mostram respectivamente o rótulo de inicialização do modelo, o número de épocas (iterações do algoritmo envolvendo todo o conjunto de dados) necessárias para atingir o critério de parada estabelecido e os EQMs referentes às simulações dos modelos em previsão 1 passo-à-frente e série sintética (previsão recursiva), sendo as duas últimas referentes aos dados de validação. Os melhores e piores desempenhos de simulação (modelos 1 e 3 respectivamente) são ilustrados nas figuras 4.11 e 4.12. É interessante notar através da figura 4.10 que ocorreu sobre-treinamento (*overfitting*) do modelo 3 aproximadamente após a centésima época, a partir da qual pode-se observar que o erro de validação passa a crescer persistentemente à medida que o erro de estimação continua decrescendo. Logo, o resultado de simulação desse modelo poderia ser superior se o seu treinamento fosse interrompido antes.

Pode-se notar a partir da tabela 4.1 bem como das figuras 4.11 e 4.12 que inicializações distintas para o procedimento de otimização podem levar a modelos com diferentes capacidades de generalização. Contudo, o desempenho global dos modelos resultantes pode ser considerado satisfatório, especialmente porque o pior modelo em simulação (modelo 3) segue o sistema real em série sintética por mais de 220 instantes de amostragem (figura 4.12), que é um horizonte de previsão longo mesmo para as aplicações mais exigentes.

Finalmente, todo o procedimento de treinamento e simulação de modelos descrito acima é repetido para a segunda configuração hierárquica discutida anteriormente. A

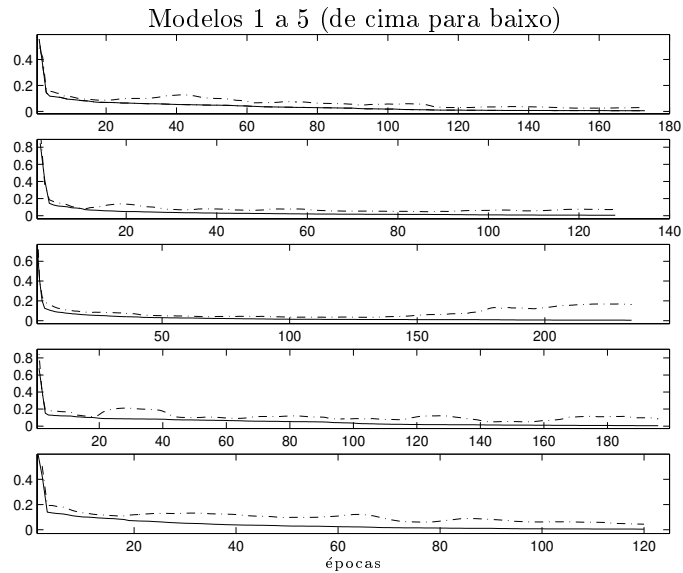


Figura 4.10: Evolução do EQM para dados de estimação (linha cheia) e validação (linha tracejada) durante treinamento de modelos hierárquicos com hierarquia $x_1(k) = u(k - 1)$, $x_2(k) = y(k - 1)$ e $x_3(k) = y(k - 2)$.

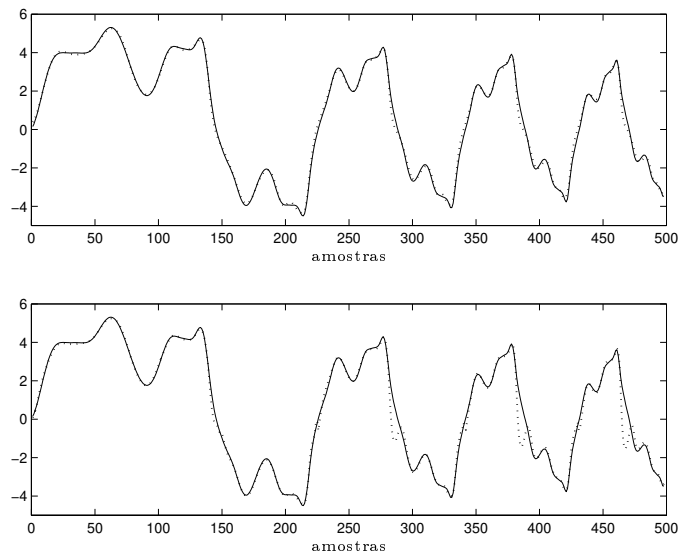


Figura 4.11: Saída do sistema (linha cheia) e previsão 1 passo-à-frente de 2 modelos hierárquicos (linha pontilhada): Melhor modelo (acima) ; pior modelo (abaixo).

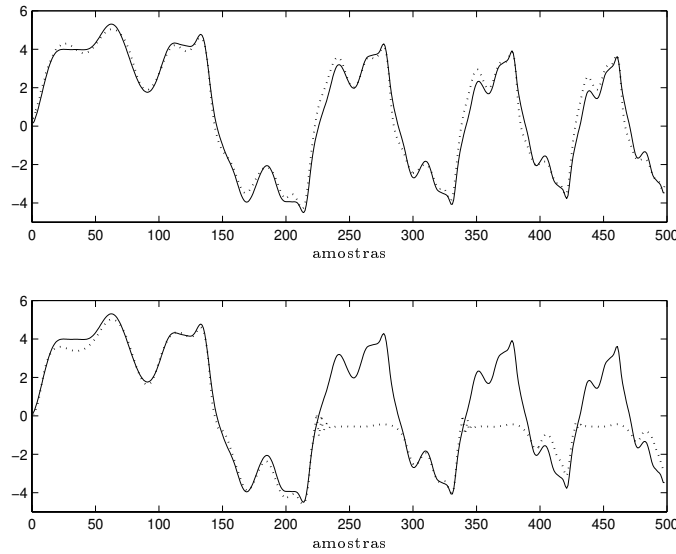


Figura 4.12: Saída do sistema (linha cheia) e série sintética de 2 modelos hierárquicos (linha pontilhada): Melhor modelo (acima) ; pior modelo (abaixo).

evolução do treinamento é ilustrada na figura 4.13 e os resultados finais são mostrados na tabela 4.2. Comparando as tabelas 4.1 e 4.2 pode-se basicamente notar que: *i)* A primeira configuração hierárquica mostrou-se muito mais robusta à inicialização dos parâmetros; *ii)* Em média, os desempenhos de treinamento e simulação dos modelos sob essa configuração foram significativamente superiores. Antes de prosseguir com uma discussão mais detalhada desses resultados, um outro caso de estudo é apresentado.

Tabela 4.2: Desempenhos de treinamento e simulação de um modelo hierárquico otimizado a partir de cinco inicializações distintas. Hierarquia: $x_1(k) = y(k - 1)$, $x_2(k) = y(k - 2)$ e $x_3(k) = u(k - 1)$.

Inicialização	Épocas	EQM previsão 1 passo	EQM série sintética
1	591	0.3124	4.0505
2	348	0.2445	1.4180
3	1805	0.1906	0.7226
4	536	0.2560	3.6453
5	83	0.0539	0.3826

2º Caso de Estudo: Função Sinc no \mathbb{R}^2

Neste caso considera-se a seguinte função:

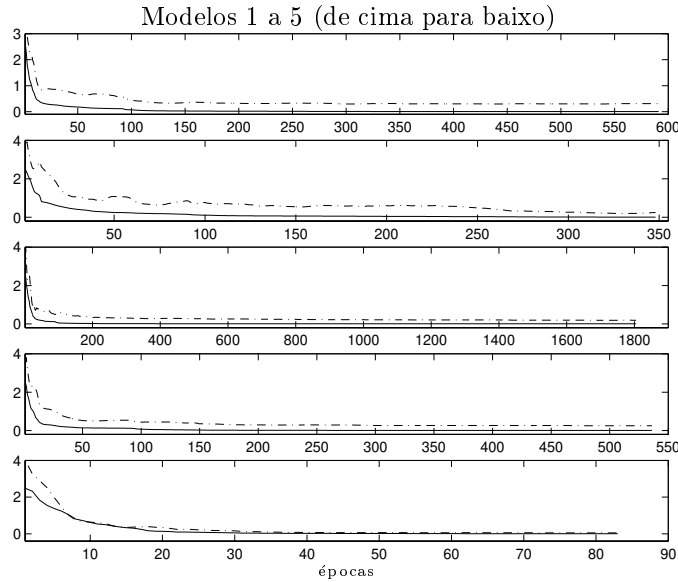


Figura 4.13: Evolução do EQM para dados de estimação (linha cheia) e validação (linha tracejada) durante treinamento de modelos hierárquicos com hierarquia $x_1(k) = y(k - 1)$, $x_2(k) = y(k - 2)$ e $x_3(k) = u(k - 1)$.

$$f(x_1, x_2) = \text{sinc}(x_1) \cdot \text{sinc}(x_2) \quad (4.24)$$

onde

$$\text{sinc}(x_i) = \begin{cases} \text{sen}(x_i)/x_i & \text{se } x_i \neq 0, \\ 1 & \text{caso contrário.} \end{cases} \quad (4.25)$$

O respectivo mapeamento não linear, ilustrado na figura 4.14, é de particular interesse pois é constituído pelo produto de duas não linearidades idênticas em cada uma das suas variáveis de entrada. Contudo, como o domínio desse mapeamento é bidimensional (o que permite a sua avaliação gráfica), a estratégia de modelagem hierárquica ilustrada na figura 4.15 torna-se necessária. Essa estratégia permite a representação do mapeamento, cuja complexidade é a mesma para cada uma das suas variáveis de entrada, através de um modelo onde essas variáveis são representadas em níveis hierárquicos e portanto com graus de liberdade distintos.

São tomados 200 pares de dados de entrada-saída para treinamento do modelo hierárquico da figura 4.15. Esses dados são gerados a partir de (4.24) e (4.25) com x_1 e x_2 amostrados aleatoriamente (com distribuição uniforme) em $[-3, 3]$. Os parâmetros do modelo, com $c = 5$ conjuntos referenciais *fuzzy* Gaussianos por variável de entrada, são inicializados conforme discutido na seção 4.2.4. É importante observar que todos

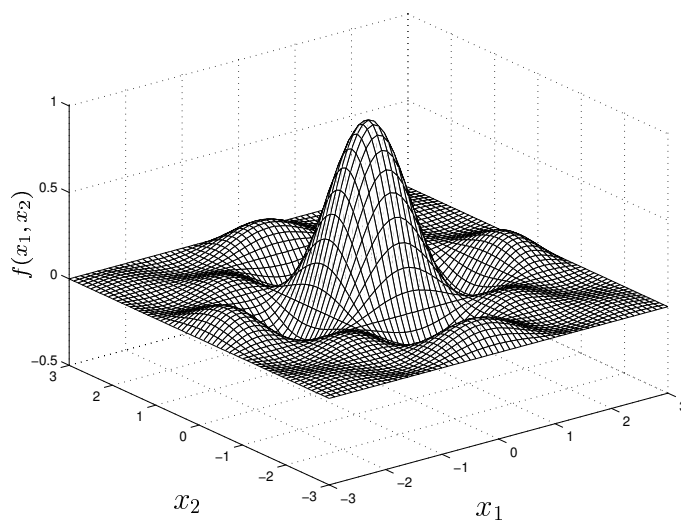
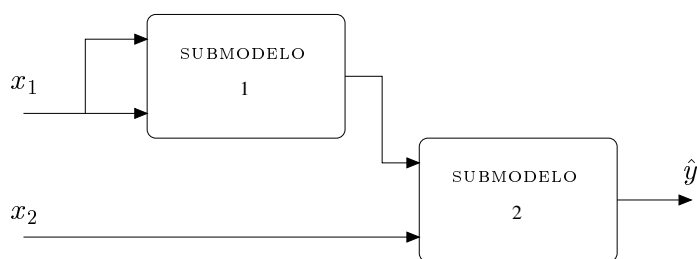
Figura 4.14: Função Sinc em um domínio compacto do \mathbb{R}^2 .

Figura 4.15: Modelo hierárquico com domínio bidimensional.

os resultados discutidos a seguir, referentes a essa inicialização, representam qualitativamente resultados análogos obtidos a partir de inicializações distintas.

O modelo é treinado por 250 épocas a partir das quais nenhuma melhora significativa na qualidade de aproximação é proporcionada, como mostrado na figura 4.16. O mapeamento do modelo final, obtido a partir de uma grade de 60×60 valores de entrada no domínio compacto $[-3, 3]^2$, é ilustrado na figura 4.17, e a superfície de erro de modelagem é apresentada na figura 4.18 (observar a escala).

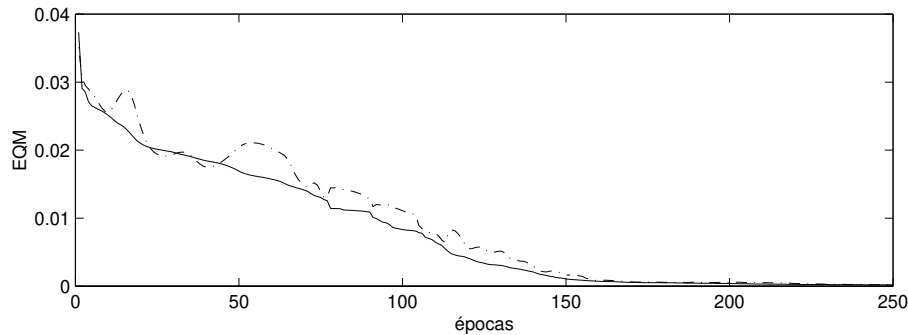


Figura 4.16: Evolução do EQM entre as saídas do mapeamento Sinc e do seu modelo hierárquico para dados de estimação (linha cheia) e validação (linha tracejada) ao longo do procedimento de treinamento desse modelo.

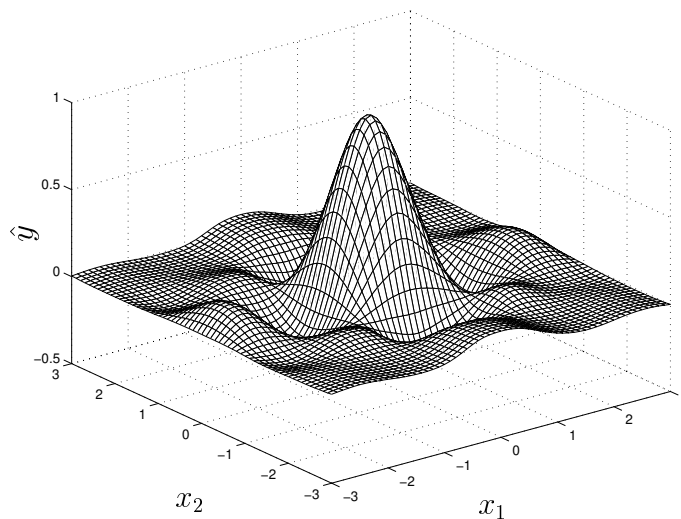


Figura 4.17: Modelo hierárquico da função Sinc em um domínio compacto do \mathcal{R}^2 .

Analisando as três superfícies, pode-se verificar que as não linearidades foram melhor aproximadas com relação à entrada x_1 , que é a variável com nível hierárquico mais alto. Esse resultado torna-se mais evidente considerando que a característica fundamental do mapeamento original, que é o produto de duas não linearidades independentes

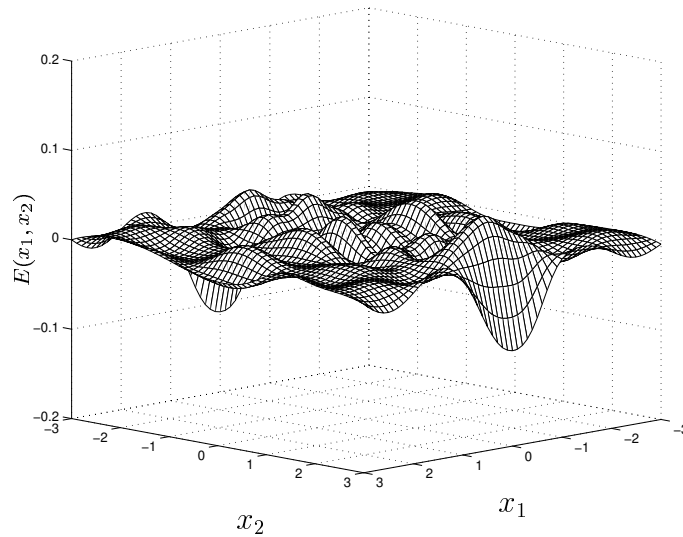


Figura 4.18: Superfície de erro de aproximação da função Sinc: $E(x_1, x_2) = f(x_1, x_2) - \hat{y}(x_1, x_2)$.

em cada uma das variáveis de entrada, tenha sido bem aproximada. Essa hipótese é consistente em função da simetria (aproximada) do mapeamento do modelo. Logo, se a não linearidade referente a x_2 tivesse sido aproximada com precisão similar àquela referente a x_1 , então o modelo deveria gerar valores quase nulos para $x_2 \approx 3$ e $x_2 \approx -3$ ($\forall x_1 \in [-3, 3]$), pois nesses casos tem-se que $\text{sinc}(x_2) \approx 0$, o que não foi verificado. Mais além, para $x_2 = 0$ ($\text{sinc}(x_2) = 1$), por exemplo, pode-se notar que o modelo representa adequadamente o mapeamento não linear em função de x_1 ; porém o oposto não ocorre com a mesma precisão.

Análise dos Resultados

Dado que uma alteração na hierarquia altera o modelo em si e portanto a função objetivo a ser minimizada na estimação dos seus parâmetros, é esperada uma alteração na evolução do treinamento de dois modelos hierárquicos inicializados com exatamente os mesmos parâmetros mas com ordens hierárquicas distintas para duas ou mais variáveis de entrada. Qualquer alteração na hierarquia faz com que as inicializações, como por exemplo a distribuição inicial dos conjuntos referenciais *fuzzy*, estejam mais ou menos distantes de soluções ótimas ou sub-ótimas, dependendo do mapeamento a ser aproximado e da ordem hierárquica escolhida. Dessa forma, dada uma solução inicial dentro do espaço de parâmetros, é a ordem hierárquica (e a respectiva função objetivo resultante) quem determina a solução final (ponto de mínimo) a ser alcançada pelo treinamento e o respectivo número de iterações necessárias para alcançá-la.

Dos resultados obtidos nos dois casos de estudo apresentados é possível inferir que os modelos são sensíveis quanto à hierarquia em um sentido mais amplo, porque alterou-se

de forma correlacionada o desempenho geral de modelagem com alterações específicas na ordem hierárquica. Particularmente, foi possível observar no primeiro caso de estudo que o desempenho de modelagem foi em média significativamente superior quando aquelas variáveis de entrada cujas relações para a saída eram mais complexas foram dispostas em níveis hierárquicos mais elevados. No segundo caso de estudo, onde a complexidade associada às diferentes variáveis eram iguais, foi observado que aproximou-se melhor as relações de entrada-saída referentes àquela disposta no nível hierárquico mais alto. Esses resultados sustentam-se no fato de que quanto mais alto o seu nível hierárquico, mais complexo torna-se o mapeamento de uma variável de entrada para a saída do modelo, envolvendo um número maior de unidades de processamento (submodelos) e portanto de graus de liberdade, isto é, mais parâmetros livres associados.

A sensibilidade dos modelos hierárquicos à disposição das suas variáveis de entrada representa de certa forma um ponto desfavorável dessa estratégia, já que a hierarquia torna-se um parâmetro adicional de projeto. Outra questão desfavorável é a perda da capacidade de processamento paralelo, natural dos sistemas *fuzzy* e RNAs *feedforward* convencionais. Por outro lado, a redução proporcionada na quantidade de parâmetros aumenta automaticamente a capacidade de generalização dos modelos.

Deve-se salientar que havendo conhecimento *a priori* a respeito da estrutura do sistema a ser modelado, esse conhecimento pode ser utilizado na seleção da ordem hierárquica. Esse procedimento será ilustrado no capítulo seguinte no contexto dos modelos hierárquicos desenvolvidos a partir do conceito de funções de base ortonormal (modelos hierárquicos FBO). É também importante observar que a ordem hierárquica e até mesmo as próprias variáveis de entrada, dentre possíveis candidatas, bem como outros parâmetros de projeto (e.g. a quantidade de conjuntos referenciais *fuzzy* por variável), podem ser selecionados automaticamente através de técnicas de computação evolutiva, conforme ilustrado em (Shimozima et al., 1995; Furuhashi et al., 1997; Tachibana e Furuhashi, 1998), onde GAs foram utilizados. Nesse caso, tem-se o requerimento de uma contrapartida computacional cuja viabilidade depende de cada aplicação.

4.2.6 Estruturas Hierárquicas Alternativas

A seleção da ordem hierárquica discutida na seção anterior remete a um problema intrínseco da estrutura hierárquica padrão considerada nos desenvolvimentos até o presente ponto. Trata-se da impossibilidade de atribuir níveis hierárquicos iguais a variáveis de entrada distintas. A eliminação dessa restrição é de particular interesse em problemas onde deseja-se estabelecer ordens hierárquicas dentro de diferentes grupos de variáveis, mas não necessariamente entre esses grupos. Essa condição pode ser obtida através da estrutura genérica ilustrada na figura 4.19.

Várias alternativas hierárquicas são possíveis a partir dessa estrutura básica que é formada pela composição de diferentes modelos hierárquicos individuais. A composição dos diferentes modelos é representada pelo operador genérico Ξ , que pode ser um mapeamento *fuzzy*/neural (inclusive hierárquico) ou mesmo qualquer operador algébrico

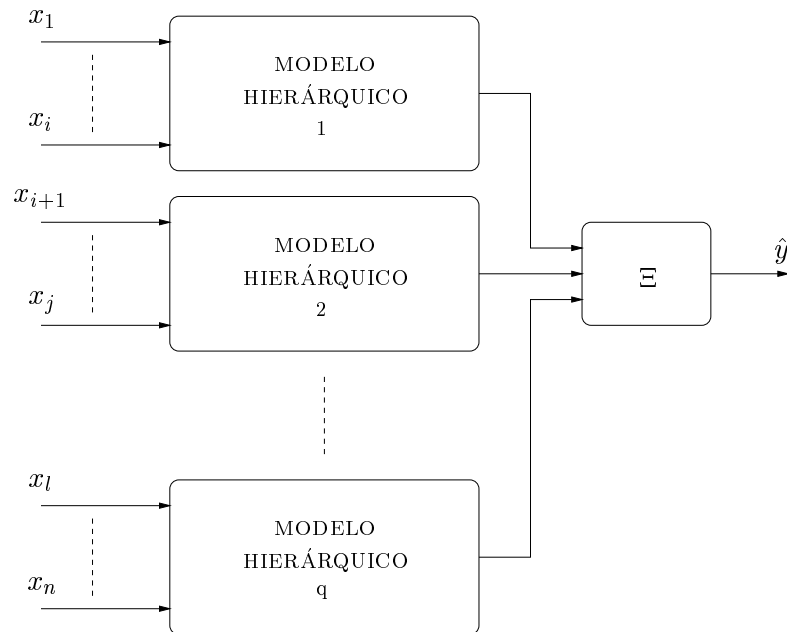


Figura 4.19: Estrutura hierárquica alternativa.

simples.

Observação 4.2 *Como a derivada da soma é a soma das derivadas, se o operador Ξ for implementado pela soma das saídas dos modelos individuais, então as equações do gradiente da função de custo J em (4.11) com relação aos parâmetros de cada modelo serão as mesmas desenvolvidas para um modelo único, discutidas na seção 4.2.3. Caso contrário, requer-se para a aplicação da regra da cadeia o cálculo adicional das derivadas parciais correspondentes ao operador Ξ selecionado.*

Uma outra estratégia que também permite a atribuição de um mesmo nível hierárquico a diferentes variáveis é a utilização de submodelos com mais de duas entradas. Um exemplo é ilustrado na figura 4.20.

Contudo, esse tipo de estrutura apresenta um número maior de regras e parâmetros em comparação à estrutura padrão, onde cada submodelo possui necessariamente apenas duas entradas. De fato, é simples demonstrar que a estrutura hierárquica padrão produz os modelos mais parcimoniosos possíveis (Raju et al., 1991).

4.2.7 Abordagens Algorítmicas Alternativas

Assim como para as arquiteturas convencionais de redes neurais e mesmo de sistemas *fuzzy* (quando os respectivos conjuntos referenciais não são pré-especificados), o problema de otimização dos modelos hierárquicos discutido na seção 4.2.3 não é convexo,

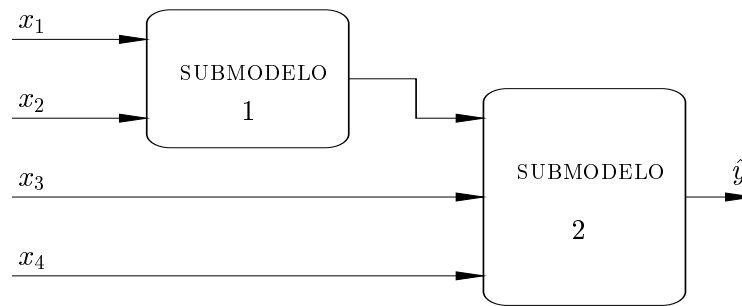


Figura 4.20: Estrutura hierárquica alternativa com 4 entradas.

estando portanto sujeito a mínimos locais. Esse problema é evidenciado com o aumento da quantidade de entradas do modelo, já que a complexidade do seu mapeamento cresce com o nível hierárquico. Nesse contexto, as alternativas hierárquicas discutidas na seção anterior podem reduzir esse problema. Existem, no entanto, possíveis modificações na estratégia de otimização dos modelos que auxiliam no tratamento dessa questão independente da arquitetura utilizada. Além de técnicas bem estabelecidas como a perturbação de soluções locais ao longo do procedimento de treinamento (otimização), algumas alternativas heurísticas direcionadas ao caso particular dos modelos hierárquicos são discutidas a seguir.

O principal problema referente à otimização de modelos hierárquicos de grande porte é a possibilidade de convergência precoce, para valores inadequados, dos parâmetros associados aos submodelos com níveis hierárquicos mais baixos, isto é, mais próximos da saída final do modelo. A maior sensibilidade da saída a variações nos mesmos reflete-se matematicamente em valores absolutos maiores para os respectivos elementos do gradiente do critério de custo a ser minimizado. Uma alternativa para esse problema é a utilização de diferentes fases de treinamento, iniciando somente com o submodelo de nível hierárquico mais elevado e envolvendo de maneira gradativa os submodelos subsequentes. O problema de convergência precoce é acentuado em relação aos conjuntos referenciais *fuzzy* das variáveis intermediárias dado que esses conjuntos podem ficar saturados, conforme discutido na seção 4.2.4. Logo, uma alternativa mais simples, porém eficaz, é a divisão do treinamento em apenas duas fases; a primeira envolvendo apenas os elementos relacionais (vetores $\Omega_{(\cdot)}$) e a segunda envolvendo também os demais parâmetros, isto é, todos os conjuntos referenciais *fuzzy*. Outra opção é a utilização de algoritmos genéticos para reduzir a sensibilidade do treinamento dos modelos aos mínimos locais. Nesse contexto, pode-se utilizar uma estratégia híbrida em que o treinamento também é dividido em duas fases (Mitchell, 1998): Na primeira fase um GA básico é utilizado apenas para localizar uma solução inicial adequada que é refinada em uma segunda fase através da otimização padrão via gradiente, como em (Shimojima et al., 1995). Essa estratégia será ilustrada no capítulo seguinte em uma aplicação prática dos modelos hierárquicos desenvolvidos a partir do conceito de funções de base ortonormal (modelos hierárquicos FBO).

Outra limitação do procedimento de otimização padrão discutido na seção 4.2.3 é que ele se restringe a aplicações *off-line*. Uma alternativa para o treinamento *on-line* dos modelos, particularmente na identificação de sistemas dinâmicos, é a utilização do algoritmo LMS (*Least Mean Squares*) (Haykin, 1989; Haykin, 1999). Esse algoritmo atualiza os parâmetros do modelo em cada instante de amostragem através de um passo, geralmente fixo, na direção do gradiente de um critério de custo instantâneo dado por J na equação (4.11) com $N = 1$. Nesse caso, o mesmo conjunto de equações apresentado na seção 4.2.3 pode ser utilizado para o cálculo do gradiente, porém com $N = 1$. Esse conjunto de equações pode ainda ser utilizado como base para linearizar o modelo, permitindo a aplicação do Filtro de Kalman Estendido (Anderson e Moore, 1979) para a estimação *on-line* dos seus parâmetros. Esse tema constitui uma das perspectivas para desenvolvimentos futuros referentes ao presente trabalho.

4.2.8 Exemplo: Problema de Classificação *Iris*

Seja o conjunto de dados *Iris* (Bezdek et al., 1999), cuja modelagem constitui um dos mais difundidos *benchmarks* nas áreas de *clustering* e classificação de padrões. Trata-se de um conjunto de padrões fenotípicos mensurados a partir de três espécies de flores: *Iris setosa*, *Iris versicolor* e *Iris virginica*. Para respeitar a nomenclatura normalmente adotada nas áreas do conhecimento relacionadas acima, essas espécies serão referidas na seqüência como “classes”, sem qualquer relação com o sentido biológico da palavra. Cada uma das três classes é representada por 50 amostras, de forma que o conjunto de dados possui um total de 150 padrões amostrais. Os padrões são vetores com quatro elementos contendo os comprimentos e larguras das sépalas e pétalas de representantes das respectivas classes. Esses quatro elementos são considerados como entradas para os modelos de classificação. As saídas correspondentes são geradas artificialmente atribuindo valores específicos para cada classe, como -1 , 0 e 1 para *setosa*, *versicolor* e *virginica*, respectivamente. Representa-se assim o sistema classificatório a ser aproximado através de um conjunto de dados de entrada-saída dado por $\{x_1(k), \dots, x_4(k), y(k)\}_{k=1}^{150}$. Para um dado modelo desse sistema, a classificação de um determinado padrão de entrada em uma das três classes é feita associando esse padrão à classe cuja saída artificial é melhor aproximada pela respectiva saída \hat{y} do modelo, isto é, *setosa* se $\hat{y} < -0.5$, *versicolor* se $-0.5 < \hat{y} < 0.5$ e *virginica* se $\hat{y} > 0.5$.

Adota-se então um modelo hierárquico com quatro entradas e três conjuntos referenciais *fuzzy* Gaussianos por variável de entrada ($c = 3$). Essa quantidade foi selecionada com base em estudos estruturais a respeito desse problema que são discutidos na seção 4.3.5. Consideram-se duas ordens hierárquicas distintas. Uma com a disposição das variáveis de entrada dada pelos comprimentos e larguras das sépalas (entradas x_1 e x_2) e pétalas (entradas x_3 e x_4), e a outra dada pelos comprimentos e larguras das pétalas (entradas x_1 e x_2) e sépalas (entradas x_3 e x_4), sendo x_1 e x_4 as entradas com maior e menor nível hierárquico, respectivamente. Essas disposições são denominadas respectivamente de ordens hierárquicas A e B.

Considera-se inicialmente um modelo com ordem hierárquica B. Os parâmetros do modelo são inicializados conforme discutido na seção 4.2.4. Para melhorar o desempenho do seu treinamento, a otimização desses parâmetros é realizada em duas fases, conforme discutido na seção 4.2.7. Na primeira fase (10 épocas de treinamento), apenas os parâmetros relacionais são atualizados a partir dos valores iniciais, e na segunda fase (90 épocas de treinamento) todos os parâmetros são atualizados. A figura 4.21-a apresenta a evolução do erro quadrático médio entre os dados de saída (artificiais) e as respectivas saídas do modelo ao longo do procedimento de otimização. As saídas inferidas a partir da aplicação de todo o conjunto de dados de entrada no modelo final obtido (cujo erro associado é de $EQM=0.0200$) são ilustradas na figura 4.21-b. A partir dessas saídas, o procedimento de classificação descrito anteriormente produz 147 padrões classificados corretamente e apenas 3 de forma incorreta. Todos esses três são classificados incorretamente entre as classes *versicolor* e *virginica*. Através da figura e de uma análise dos dados (ver seção 4.3.5) pode-se notar que essas classes são mais similares entre si do que a classe *setosa*, que difere em maior grau das demais.

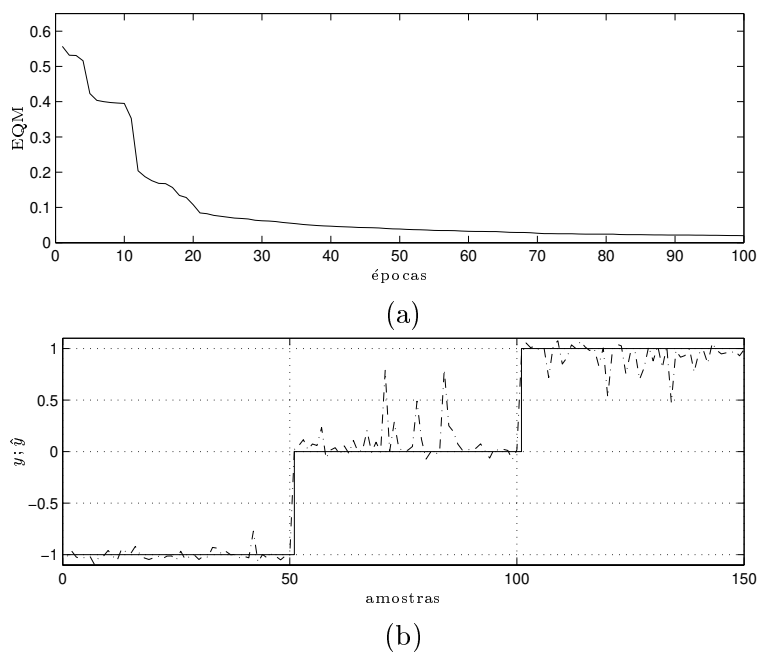


Figura 4.21: (a) Evolução do EQM ao longo do treinamento; (b) Saída do modelo (linha tracejada) e saídas artificiais (linha cheia) associadas às classes *setosa* (-1), *versicolor* (0) e *virginica* ($+1$): Ordem hierárquica B.

Considera-se a seguir um modelo com ordem hierárquica A. Os parâmetros do modelo são inicializados conforme discutido na seção 4.2.4 e a otimização desses parâmetros é realizada exatamente como no caso anterior. A figura 4.22-a apresenta a evolução do erro quadrático médio entre os dados de saída (artificiais) e as respectivas saídas do modelo ao longo do procedimento de otimização. As saídas inferidas a partir do mode-

lo final obtido (cujo erro associado é de $EQM=0.0222$) são ilustradas na figura 4.22-b. A partir dessas saídas, o procedimento de classificação também produz 147 padrões classificados corretamente e apenas 3 de forma incorreta, todos esses classificados incorretamente entre as classes *versicolor* e *virginica*.

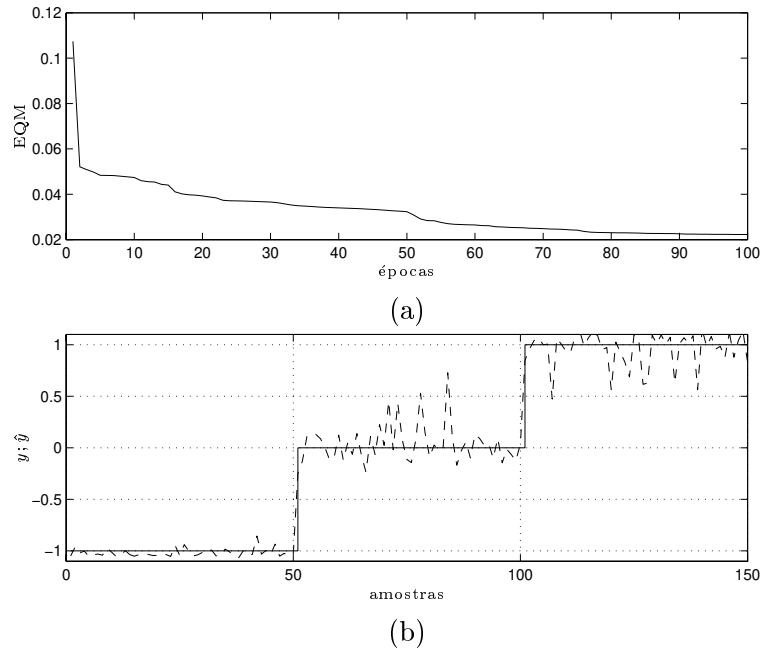


Figura 4.22: (a) Evolução do EQM ao longo do treinamento; (b) Saída do modelo (linha tracejada) e saídas artificiais (linha cheia) associadas às classes *sestosa* (-1), *versicolor* (0) e *virginica* ($+1$): Ordem hierárquica A.

Nos procedimentos de modelagem descritos acima avaliou-se somente o desempenho dos modelos finais com base no mesmo conjunto de dados utilizados no seu treinamento, o que proporciona apenas uma indicação da sua capacidade de aproximação. Para uma avaliação da capacidade de generalização dos modelos, que representa um critério adequado para a interrupção do treinamento dos mesmos, separa-se aleatoriamente 10% dos padrões referentes a cada classe para constituírem um conjunto de dados de validação. Considera-se então o modelo com ordem hierárquica A, em função de resultados que serão discutidos no exemplo da seção 4.3.5, e uma nova inicialização de parâmetros, otimizados da mesma forma descrita anteriormente (em 2 fases), porém por 150 épocas e utilizando o novo conjunto de dados de estimação. A figura 4.23 apresenta a evolução do erro quadrático médio entre os dados de saída e as respectivas saídas do modelo ao longo do procedimento de otimização.

Pode-se observar na figura 4.23 que a partir de aproximadamente 90 épocas passa a ocorrer sobre-treinamento dos parâmetros do modelo, caracterizado pelo crescimento médio do erro referente aos dados de validação à medida que o erro referente aos dados de estimação decresce. Esses erros, no entanto, não são fundamentais do ponto de

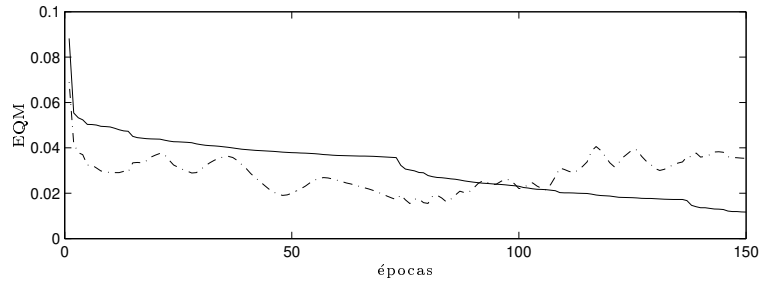


Figura 4.23: Evolução do EQM ao longo do treinamento com relação aos dados de estimação (linha cheia) e validação (linha tracejada): Ordem hierárquica A.

vista do problema classificatório, onde o desempenho de classificação é o fator mais importante. Os desempenhos de classificação dos modelos parciais, obtidos a cada 10 épocas de treinamento, são apresentados na tabela 4.3. Observa-se que o melhor modelo é obtido após 90 épocas, justamente a partir das quais começa haver sobre-treinamento.

Tabela 4.3: Evolução do desempenho de classificação do modelo hierárquico a cada 10 épocas de treinamento.

Épocas	Erros (dados estimação)	Erros (dados validação)
10	5	0
20	5	0
30	5	0
40	4	0
50	6	0
60	6	0
70	7	0
80	4	0
90	2	0
100	4	1
110	3	1
120	1	1
130	1	1
140	2	1
150	2	1

É importante destacar que, embora o problema *Iris* tenha sido amplamente utilizado na literatura para fins de comparação entre diferentes modelos de classificação (e.g. ver (Delgado, 2002)), esta não é a finalidade do presente exemplo. Vale lembrar que um aumento na quantidade “c” de conjuntos referenciais *fuzzy* por variável de entra-

da, por exemplo, poderia melhorar o desempenho quantitativo do modelo hierárquico. Nesse contexto, tem-se que o objetivo do exemplo é apenas ilustrar a utilização da estratégia hierárquica, cujas vantagens estruturais foram discutidas anteriormente, em uma aplicação típica.

4.3 Modelos Hierárquicos com Submodelos Relacionais Convencionais

4.3.1 Estrutura Hierárquica Padrão

Uma alternativa adicional para a construção de modelos com estrutura hierárquica é a implementação das unidades de processamento utilizando modelos relacionais convencionais, discutidos no capítulo 3, em substituição aos modelos relacionais simplificados. A estrutura padrão nesse caso é ilustrada, para 4 variáveis de entrada, na figura 4.24. Embora as discussões subseqüentes tomem essa estrutura como base, vale lembrar que outras opções estruturais também são possíveis, tais como aquelas discutidas na seção 4.2.6.

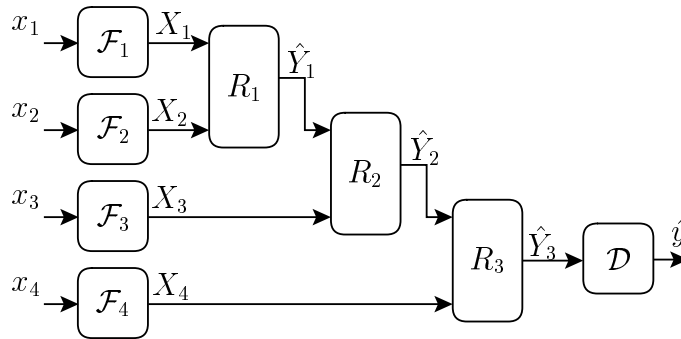


Figura 4.24: Modelo relacional hierárquico com 4 entradas.

Na figura 4.24, $\mathcal{F}_{(\cdot)}$ e \mathcal{D} representam mapeamentos de fuzzificação e defuzzificação dados pelas seguintes equações, respectivamente:

$$\begin{aligned} X_i &= [X_{i_1} \ X_{i_2} \ \cdots \ X_{i_c}]^T \\ &= [\mathcal{X}_{i_1}(x_i) \ \mathcal{X}_{i_2}(x_i) \ \cdots \ \mathcal{X}_{i_c}(x_i)]^T, \quad i = 1, \dots, n \end{aligned} \quad (4.26)$$

$$\hat{y} = \frac{\sum_{h=1}^c \hat{Y}_{(n-1)_h} \phi_h}{\sum_{h=1}^c \hat{Y}_{(n-1)_h}} \quad (4.27)$$

onde \mathcal{X}_{i_j} é o j -ésimo conjunto referencial *fuzzy* associado à i -ésima variável de entrada e ϕ_h é o valor modal do h -ésimo conjunto referencial *fuzzy* associado à variável de saída (que coincide com seu centro se esse conjunto for simétrico). Por simplicidade e sem perda de generalidade, considera-se inicialmente uma mesma quantidade c de conjuntos referenciais definidos para cada variável de entrada e para a variável de saída do modelo.

A saída *fuzzy* do i -ésimo submodelo relacional pode ser escrita como

$$\hat{Y}_i^T = [\hat{Y}_{i_1} \ \hat{Y}_{i_2} \ \cdots \ \hat{Y}_{i_c}] = \Psi_i^T R_i, \quad i = 1, \dots, n-1 \quad (4.28)$$

ou individualmente para cada elemento como

$$\hat{Y}_{i_j} = \sum_{l=1}^{c^2} \Psi_{i_l} R_{i_l,j} \quad (4.29)$$

sendo $R_{i_l,j}$ e Ψ_{i_l} , respectivamente, elementos específicos da matriz relacional R_i ($c^2 \times c$) e do vetor *fuzzy* de entrada Ψ_i ($c^2 \times 1$) do i -ésimo submodelo. A equação (4.29) representa a implementação do i -ésimo submodelo relacional através do operador de composição *fuzzy* soma-produto (Sing e Postlethwaite, 1997), que é equivalente ao operador média-produto utilizado no capítulo 3 exceto por um fator proporcional ($1/l$), não utilizado aqui por simplicidade de notação. A matriz R_i representa um subconjunto dos parâmetros de projeto livre do modelo, enquanto o vetor Ψ_i é dado por

$$\Psi_i = [\Psi_{i_1} \ \cdots \ \Psi_{i_{c^2}}]^T = \begin{cases} X_{i+1} \otimes \hat{Y}_{i-1}, & i = 2, \dots, n-1 \\ X_{i+1} \otimes X_i, & i = 1 \end{cases} \quad (4.30)$$

onde \otimes denota o produto de Kronecker. Nesses modelos hierárquicos cada uma das unidades de processamento (submodelo relacional convencional) tem como saída um vetor *fuzzy* \hat{Y}_i , e não um valor *crisp* \hat{y}_i como nos modelos das seções anteriores. Como pode ser visto nas equações acima, esse vetor é utilizado diretamente como entrada da unidade subsequente, eliminando desse modo os procedimentos de fuzzificação e os conjuntos referenciais *fuzzy* referentes às variáveis intermediárias. Tem-se portanto que os consequentes *fuzzy* de um submodelo são antecedentes do submodelo seguinte, o que é pressuposto do raciocínio silogístico discutido na seção introdutória.

4.3.2 Interpretação Lingüística

O principal problema associado aos modelos hierárquicos, quando analisados sob o ponto de vista de lógica *fuzzy*, é a dificuldade de interpretação qualitativa das relações entre os parâmetros desses modelos e o comportamento dos sistemas representados por eles.

Essa dificuldade ocorre porque as variáveis intermediárias do modelo usualmente são artificiais, i.e., não estão associadas a variáveis do sistema correspondente. Logo, essas variáveis não possuem qualquer significado físico e por conseguinte não permitem interpretação lingüística (Wan e Wang, 2000). Um resultado importante para solucionar esse problema é a obtenção de uma relação unilateral entre os modelos hierárquicos constituídos de submodelos relacionais *fuzzy* convencionais e os modelos relacionais *fuzzy* convencionais não hierárquicos. Esse resultado permite a identificação de um modelo hierárquico, com a respectiva redução na quantidade de parâmetros a serem estimados que esses modelos proporcionam, seguida da geração do modelo não hierárquico equivalente. Esse procedimento possibilita a extração de regras lingüísticas a partir do modelo equivalente através da utilização da metodologia apresentada no capítulo 3. A transformação referida acima é descrita através do seguinte teorema (Campello e Amaral, 2002b):

Teorema 4.1 *Existe uma transformação de equivalência unilateral entre o modelo hierárquico dado pelas equações (4.26), (4.28) e (4.30) e um modelo relacional convencional não hierárquico, com vetor fuzzy de saída dado por*

$$\hat{\mathbf{Y}}^T = [X_1 \otimes X_2 \otimes \cdots \otimes X_n]^T \Omega \quad (4.31)$$

onde Ω é a matriz relacional, definida como

$$\Omega \triangleq \begin{bmatrix} \Omega_{1,\dots,1,1} & \Omega_{1,\dots,1,2} & \cdots & \Omega_{1,\dots,1,c} \\ \Omega_{1,\dots,2,1} & \Omega_{1,\dots,2,2} & \cdots & \Omega_{1,\dots,2,c} \\ \vdots & \vdots & \vdots & \vdots \\ \Omega_{c,\dots,c,1} & \Omega_{c,\dots,c,2} & \cdots & \Omega_{c,\dots,c,c} \end{bmatrix} \quad (4.32)$$

A transformação caracteriza-se completamente pela seguinte equação:

$$\Omega_{h_1, h_2, \dots, h_n, j} = \sum_{m_1=1}^c \sum_{m_2=1}^c \cdots \sum_{m_\beta=1}^c \tilde{R}_{1_{h_2, h_1, m_1}} \tilde{R}_{2_{h_3, m_1, m_2}} \tilde{R}_{3_{h_4, m_2, m_3}} \cdots \cdots \tilde{R}_{\beta_{h_\gamma, m_\alpha, m_\beta}} \tilde{R}_{\gamma_{h_n, m_\beta, j}} \quad (4.33)$$

com $\tilde{R}_{i_{p,q,j}} \triangleq R_{i_{\rho,j}}$, onde $p, q = 1, \dots, c$ e $\rho = (p-1)c + q$. As variáveis auxiliares α , β e γ são definidas como $\alpha + 2 = \beta + 1 = \gamma \triangleq n - 1$ para reduzir os subíndices da expressão.

Prova:

Para demonstrar a transformação entre um modelo hierárquico de submodelos relacionais convencionais e o seu equivalente não hierárquico, utilizam-se as seguintes

definições auxiliares: $\tilde{\Psi}_{i_{p,q}} \triangleq \Psi_{i_\rho}$ e $\tilde{R}_{i_{p,q,j}} \triangleq R_{i_{\rho,j}}$, onde $p, q = 1, \dots, c$ e $\rho = (p-1)c + q$. Considera-se inicialmente um modelo com $n = 4$, como aquele ilustrado na figura 4.24, e $c = 2$. Nesse caso, para $i \in \{1, 2, 3\}$ e $j \in \{1, 2\}$ tem-se

$$\begin{bmatrix} \tilde{R}_{i_{1,1,j}} & \tilde{R}_{i_{1,2,j}} \\ \tilde{R}_{i_{2,1,j}} & \tilde{R}_{i_{2,2,j}} \end{bmatrix} \triangleq \begin{bmatrix} R_{i_{1,j}} & R_{i_{2,j}} \\ R_{i_{3,j}} & R_{i_{4,j}} \end{bmatrix} \quad (4.34)$$

$$\begin{bmatrix} \tilde{\Psi}_{i_{1,1}} & \tilde{\Psi}_{i_{1,2}} \\ \tilde{\Psi}_{i_{2,1}} & \tilde{\Psi}_{i_{2,2}} \end{bmatrix} \triangleq \begin{bmatrix} \Psi_{i_1} & \Psi_{i_2} \\ \Psi_{i_3} & \Psi_{i_4} \end{bmatrix} \quad (4.35)$$

Das equações (4.26), (4.28), (4.30) e (4.35) pode-se verificar que

$$\begin{bmatrix} \tilde{\Psi}_{1,1} & \tilde{\Psi}_{1,2} \\ \tilde{\Psi}_{1,2,1} & \tilde{\Psi}_{1,2,2} \end{bmatrix} = \begin{bmatrix} X_{2_1} X_{1_1} & X_{2_1} X_{1_2} \\ X_{2_2} X_{1_1} & X_{2_2} X_{1_2} \end{bmatrix} \quad (4.36)$$

$$\begin{bmatrix} \tilde{\Psi}_{2,1,1} & \tilde{\Psi}_{2,1,2} \\ \tilde{\Psi}_{2,2,1} & \tilde{\Psi}_{2,2,2} \end{bmatrix} = \begin{bmatrix} X_{3_1} \hat{Y}_{1_1} & X_{3_1} \hat{Y}_{1_2} \\ X_{3_2} \hat{Y}_{1_1} & X_{3_2} \hat{Y}_{1_2} \end{bmatrix} \quad (4.37)$$

$$\begin{bmatrix} \tilde{\Psi}_{3,1,1} & \tilde{\Psi}_{3,1,2} \\ \tilde{\Psi}_{3,2,1} & \tilde{\Psi}_{3,2,2} \end{bmatrix} = \begin{bmatrix} X_{4_1} \hat{Y}_{2_1} & X_{4_1} \hat{Y}_{2_2} \\ X_{4_2} \hat{Y}_{2_1} & X_{4_2} \hat{Y}_{2_2} \end{bmatrix} \quad (4.38)$$

Utilizando as equações (4.29), (4.34) e (4.35), pode-se rescrever a saída do primeiro submodelo como

$$\hat{Y}_{1_j} = \sum_{p=1}^2 \sum_{q=1}^2 \tilde{\Psi}_{1_{p,q}} \tilde{R}_{1_{p,q,j}}, \quad j = 1, 2 \quad (4.39)$$

e a partir de (4.36) e (4.39) como

$$\begin{aligned} \hat{Y}_{1_j} = & X_{2_1} X_{1_1} \tilde{R}_{1_{1,1,j}} + X_{2_1} X_{1_2} \tilde{R}_{1_{1,2,j}} + \\ & X_{2_2} X_{1_1} \tilde{R}_{1_{2,1,j}} + X_{2_2} X_{1_2} \tilde{R}_{1_{2,2,j}}, \quad j = 1, 2 \end{aligned} \quad (4.40)$$

A saída do segundo submodelo é rescrita analogamente a partir de (4.29), (4.34), (4.35) e (4.37) como

$$\begin{aligned} \hat{Y}_{2j} = & X_{3_1} \hat{Y}_{1_1} \tilde{R}_{2_{1,1,j}} + X_{3_1} \hat{Y}_{1_2} \tilde{R}_{2_{1,2,j}} + \\ & X_{3_2} \hat{Y}_{1_1} \tilde{R}_{2_{2,1,j}} + X_{3_2} \hat{Y}_{1_2} \tilde{R}_{2_{2,2,j}} , \quad j = 1, 2 \end{aligned} \quad (4.41)$$

e de (4.40) e (4.41) como

$$\begin{aligned} \hat{Y}_{2j} = & X_{3_1} X_{2_1} X_{1_1} (\tilde{R}_{1_{1,1,1}} \tilde{R}_{2_{1,1,j}} + \tilde{R}_{1_{1,1,2}} \tilde{R}_{2_{1,2,j}}) + \\ & X_{3_1} X_{2_1} X_{1_2} (\tilde{R}_{1_{1,2,1}} \tilde{R}_{2_{1,1,j}} + \tilde{R}_{1_{1,2,2}} \tilde{R}_{2_{1,2,j}}) + \\ & X_{3_1} X_{2_2} X_{1_1} (\tilde{R}_{1_{2,1,1}} \tilde{R}_{2_{1,1,j}} + \tilde{R}_{1_{2,1,2}} \tilde{R}_{2_{1,2,j}}) + \\ & X_{3_1} X_{2_2} X_{1_2} (\tilde{R}_{1_{2,2,1}} \tilde{R}_{2_{1,1,j}} + \tilde{R}_{1_{2,2,2}} \tilde{R}_{2_{1,2,j}}) + \\ & X_{3_2} X_{2_1} X_{1_1} (\tilde{R}_{1_{1,1,1}} \tilde{R}_{2_{2,1,j}} + \tilde{R}_{1_{1,1,2}} \tilde{R}_{2_{2,2,j}}) + \\ & \quad \vdots \\ & X_{3_2} X_{2_2} X_{1_2} (\tilde{R}_{1_{2,2,1}} \tilde{R}_{2_{2,1,j}} + \tilde{R}_{1_{2,2,2}} \tilde{R}_{2_{2,2,j}}) \end{aligned} \quad (4.42)$$

De (4.42) pode-se verificar que

$$\hat{Y}_{2j} = \sum_{h_1=1}^2 \sum_{h_2=1}^2 \sum_{h_3=1}^2 X_{3_{h_3}} X_{2_{h_2}} X_{1_{h_1}} \left(\sum_{m=1}^2 \tilde{R}_{1_{h_2,h_1,m}} \tilde{R}_{2_{h_3,m,j}} \right) , \quad j = 1, 2 \quad (4.43)$$

Utilizando (4.29), (4.34), (4.35), (4.38) e (4.42), pode-se rescrever a saída do terceiro submodelo hierárquico como

$$\begin{aligned} \hat{Y}_{3j} = & X_{4_1} X_{3_1} X_{2_1} X_{1_1} (\tilde{R}_{1_{1,1,1}} \tilde{R}_{2_{1,1,1}} \tilde{R}_{3_{1,1,j}} + \tilde{R}_{1_{1,1,2}} \tilde{R}_{2_{1,2,1}} \tilde{R}_{3_{1,1,j}} \\ & + \tilde{R}_{1_{1,1,1}} \tilde{R}_{2_{1,1,2}} \tilde{R}_{3_{1,2,j}} + \tilde{R}_{1_{1,1,2}} \tilde{R}_{2_{1,2,2}} \tilde{R}_{3_{1,2,j}}) + \\ & X_{4_1} X_{3_1} X_{2_1} X_{1_2} (\tilde{R}_{1_{1,2,1}} \tilde{R}_{2_{1,1,1}} \tilde{R}_{3_{1,1,j}} + \tilde{R}_{1_{1,2,2}} \tilde{R}_{2_{1,2,1}} \tilde{R}_{3_{1,1,j}} \\ & + \tilde{R}_{1_{1,2,1}} \tilde{R}_{2_{1,1,2}} \tilde{R}_{3_{1,2,j}} + \tilde{R}_{1_{1,2,2}} \tilde{R}_{2_{1,2,2}} \tilde{R}_{3_{1,2,j}}) + \\ & X_{4_1} X_{3_1} X_{2_2} X_{1_1} (\tilde{R}_{1_{2,1,1}} \tilde{R}_{2_{1,1,1}} \tilde{R}_{3_{1,1,j}} + \tilde{R}_{1_{2,1,2}} \tilde{R}_{2_{1,2,1}} \tilde{R}_{3_{1,1,j}} \\ & + \tilde{R}_{1_{2,1,1}} \tilde{R}_{2_{1,1,2}} \tilde{R}_{3_{1,2,j}} + \tilde{R}_{1_{2,1,2}} \tilde{R}_{2_{1,2,2}} \tilde{R}_{3_{1,2,j}}) + \\ & \quad \vdots \\ & X_{4_1} X_{3_2} X_{2_1} X_{1_1} (\tilde{R}_{1_{1,1,1}} \tilde{R}_{2_{2,1,1}} \tilde{R}_{3_{1,1,j}} + \tilde{R}_{1_{1,1,2}} \tilde{R}_{2_{2,2,1}} \tilde{R}_{3_{1,1,j}} \\ & + \tilde{R}_{1_{1,1,1}} \tilde{R}_{2_{2,1,2}} \tilde{R}_{3_{1,2,j}} + \tilde{R}_{1_{1,1,2}} \tilde{R}_{2_{2,2,2}} \tilde{R}_{3_{1,2,j}}) + \\ & \quad \vdots \\ & X_{4_2} X_{3_2} X_{2_2} X_{1_2} (\tilde{R}_{1_{2,2,1}} \tilde{R}_{2_{2,1,1}} \tilde{R}_{3_{2,1,j}} + \tilde{R}_{1_{2,2,2}} \tilde{R}_{2_{2,2,1}} \tilde{R}_{3_{2,1,j}} \\ & + \tilde{R}_{1_{2,2,1}} \tilde{R}_{2_{2,1,2}} \tilde{R}_{3_{2,2,j}} + \tilde{R}_{1_{2,2,2}} \tilde{R}_{2_{2,2,2}} \tilde{R}_{3_{2,2,j}}) \end{aligned} \quad (4.44)$$

De (4.44) pode-se induzir a seguinte expressão:

$$\hat{Y}_{3_j} = \sum_{h_1=1}^2 \sum_{h_2=1}^2 \sum_{h_3=1}^2 \sum_{h_4=1}^2 X_{4_{h_4}} X_{3_{h_3}} X_{2_{h_2}} X_{1_{h_1}} \cdot \left(\sum_{m_1=1}^2 \sum_{m_2=1}^2 \tilde{R}_{1_{h_2, h_1, m_1}} \tilde{R}_{2_{h_3, m_1, m_2}} \tilde{R}_{3_{h_4, m_2, j}} \right), \quad j = 1, 2 \quad (4.45)$$

Desse caso particular ($n = 4$ e $c = 2$) pode-se induzir a equação para o caso geral, que é dada por

$$\hat{Y}_{(n-1)_j} = \sum_{h_1=1}^c \sum_{h_2=1}^c \cdots \sum_{h_n=1}^c X_{1_{h_1}} X_{2_{h_2}} \cdots X_{n_{h_n}} \Omega_{h_1, h_2, \dots, h_n, j} \quad (4.46)$$

onde $j = 1, \dots, c$ e

$$\Omega_{h_1, h_2, \dots, h_n, j} \triangleq \sum_{m_1=1}^c \sum_{m_2=1}^c \cdots \sum_{m_\beta=1}^c \tilde{R}_{1_{h_2, h_1, m_1}} \tilde{R}_{2_{h_3, m_1, m_2}} \tilde{R}_{3_{h_4, m_2, m_3}} \cdots \cdots \tilde{R}_{\beta_{h_\gamma, m_\alpha, m_\beta}} \tilde{R}_{\gamma_{h_n, m_\beta, j}} \quad (4.47)$$

sendo α , β e γ definidos como $\alpha + 2 = \beta + 1 = \gamma \triangleq n - 1$ para reduzir os subíndices da expressão. Definindo

$$\Omega \triangleq \begin{bmatrix} \Omega_{1, \dots, 1, 1} & \Omega_{1, \dots, 1, 2} & \cdots & \Omega_{1, \dots, 1, c} \\ \Omega_{1, \dots, 2, 1} & \Omega_{1, \dots, 2, 2} & \cdots & \Omega_{1, \dots, 2, c} \\ \vdots & \vdots & \vdots & \vdots \\ \Omega_{c, \dots, c, 1} & \Omega_{c, \dots, c, 2} & \cdots & \Omega_{c, \dots, c, c} \end{bmatrix} \quad (4.48)$$

e $\hat{Y} \triangleq \hat{Y}_{n-1}$, obtém-se (4.46) na forma vetorial, como segue:

$$\hat{Y}^T = [X_1 \otimes X_2 \otimes \cdots \otimes X_n]^T \Omega \quad (4.49)$$

■

O teorema 4.1 assume um mesmo número “ c ” de conjuntos *fuzzy* por variável e o uso da estrutura hierárquica padrão, com submodelos providos necessariamente de apenas duas entradas. Essas hipóteses simplificam significativamente a transformação de equivalência entre os modelos relacionais hierárquicos e não hierárquicos bem como

o seu desenvolvimento, mas não são restritivas. De fato, pode-se verificar que existe uma transformação de equivalência para qualquer modelo com submodelos relacionais convencionais, mesmo que esses submodelos sejam providos de quantidades diferentes de entradas e de conjuntos *fuzzy* por variável. Como exemplo, considera-se o modelo ilustrado na figura 4.20, com três conjuntos *fuzzy* definidos para a variável x_1 e dois conjuntos *fuzzy* definidos para todas as demais variáveis. Através de procedimento análogo àquele utilizado na prova do teorema 4.1, verifica-se que a saída *fuzzy* desse modelo hierárquico pode ser escrita através de uma representação não hierárquica equivalente dada por:

$$\hat{Y}_2^T = [X_1 \otimes X_2 \otimes X_3 \otimes X_4]^T \Omega \quad (4.50)$$

sendo a matriz relacional equivalente nesse caso dada por:

$$\Omega = \begin{bmatrix} \Omega_{1,1,1,1,1} & \Omega_{1,1,1,1,2} \\ \Omega_{1,1,1,2,1} & \Omega_{1,1,1,2,2} \\ \vdots & \vdots \\ \Omega_{2,2,2,3,1} & \Omega_{2,2,2,3,2} \end{bmatrix} \quad (4.51)$$

cujos elementos são dados por

$$\Omega_{h_1, h_2, h_3, h_4, j} = R_{2_{\alpha, j}} R_{1_{\beta, 1}} + R_{2_{(\alpha+1), j}} R_{1_{\beta, 2}} \quad (4.52)$$

com $\alpha = 4(h_4 - 1) + 2(h_3 - 1) + 1$, $\beta = 3(h_2 - 1) + h_1$ e $j = 1, 2$.

4.3.3 Otimização

A seguir discute-se o problema de estimação dos modelos hierárquicos com submodelos relacionais convencionais, que consiste em otimizar, segundo algum critério, os seus parâmetros de projeto livre. Assumindo inicialmente que se deseja extrair conhecimento lingüístico a partir dos modelos obtidos, supõe-se que os seus conjuntos referenciais *fuzzy* de entrada e saída, bem como os rótulos lingüísticos associados a eles, sejam definidos *a priori*, como discutido no capítulo 3. Nesse caso, os parâmetros desconhecidos do modelo resumem-se aos elementos das matrizes relacionais R_i ($i = 1, \dots, n - 1$).

Os métodos de extração de conhecimento lingüístico apresentados no capítulo 3 baseiam-se em uma condição de interpretabilidade que é dada pela não negatividade dos elementos da matriz relacional em uma equação do tipo (4.31). Da equação (4.33) tem-se que uma condição suficiente para a não negatividade dos elementos da matriz relacional do modelo equivalente não hierárquico é dada pela não negatividade dos elementos relacionais do modelo hierárquico original. Essa restrição deve portanto ser

considerada no problema de otimização para identificação desses modelos, tratado na seqüência.

No capítulo 3 foram discutidas as vantagens da utilização de um critério de custo *fuzzy* para o problema de otimização de modelos relacionais, como o critério abaixo:

$$J = \frac{1}{2} \sum_{k=1}^N \sum_{h=1}^c (Y_h(k) - S_h(k))^2 \quad (4.53)$$

onde N é o número disponível de amostras de entrada-saída $(\{x_1(k), \dots, x_n(k), y(k)\}_{k=1}^N)$ do sistema a ser modelado, $S_h(k) \triangleq \hat{Y}_{(n-1)h}(k)$ é o h -ésimo elemento do vetor *fuzzy* de saída do modelo para o k -ésimo padrão de dados de entrada e $Y_h(k)$ é a fuzzificação do k -ésimo valor de saída do sistema, dada por

$$\begin{aligned} Y(k) &= [Y_1(k) \ Y_2(k) \ \dots \ Y_c(k)]^T \\ &= [\mathcal{Y}_1(y(k)) \ \mathcal{Y}_2(y(k)) \ \dots \ \mathcal{Y}_c(y(k))]^T \end{aligned} \quad (4.54)$$

sendo \mathcal{Y}_h o h -ésimo conjunto referencial *fuzzy* associado à variável de saída. Mostrou-se no capítulo 3 que esse tipo de função objetivo envolvendo grandezas *fuzzy* é mais adequado no contexto de modelagem qualitativa, i.e., para extração de conhecimento lingüístico, quando comparado ao critério clássico de soma dos erros quadráticos entre as saídas *crisp* do sistema e do respectivo modelo, ou seja

$$J_2 = \frac{1}{2} \sum_{k=1}^N (y(k) - \hat{y}(k))^2 \quad (4.55)$$

basicamente porque esse último não envolve diretamente os conjuntos referenciais *fuzzy* e os respectivos rótulos lingüísticos associados à variável de saída do sistema/modelo. Mostrou-se ainda que a minimização de (4.53) pode ser tão eficiente sob o ponto de vista de precisão quantitativa dos modelos obtidos quanto a de (4.55), desde que os conjuntos referenciais *fuzzy* da saída $\mathcal{Y}_{(\cdot)}$ sejam definidos de forma adequada.

Das considerações acima, o problema de otimização para a estimação dos parâmetros relacionais dos modelos hierárquicos dados pelas equações (4.26) a (4.30) pode ser escrito como

$$\begin{aligned} \min_{R_1, \dots, R_{(n-1)}} \quad & J = \frac{1}{2} \sum_{k=1}^N \sum_{h=1}^c (Y_h(k) - S_h(k))^2 \\ \text{s. a} \quad & R_1 \geq \mathbf{0}, \dots, R_{(n-1)} \geq \mathbf{0} \end{aligned} \quad (4.56)$$

Esse problema pode ser resolvido utilizando técnicas de programação não linear com restrições. Devido à complexidade da função de custo J (que não é quadrática nos seus

argumentos, por exemplo) e à simplicidade das restrições (que são lineares), um método adequado para a solução ótima ou sub-ótima desse problema é o algoritmo do gradiente projetado (Bazaraa et al., 1993). O principal inconveniente computacional relacionado a esse método, especialmente em problemas de grande dimensão, que é geralmente o caso de sistemas *fuzzy* e RNAs, é o cálculo da matriz inversa $(A_1 A_1^T)^{-1}$, onde A_1 é a matriz de restrições ativas. Contudo, no presente contexto esse problema não existe pois A_1 é sempre uma matriz identidade devido à presença somente de restrições de não negatividade em (4.56).

O método do gradiente projetado requer o cálculo do gradiente da função de custo J em relação aos parâmetros desconhecidos, ou seja, os elementos das matrizes R_i . Esse gradiente pode ser calculado aplicando a regra da cadeia nas equações (4.53), (4.28) e (4.30). O procedimento é análogo àquele realizado para o caso de submodelos hierárquicos simplificados, na seção 4.2.3. As equações para o caso geral com qualquer número de entradas podem ser induzidas a partir do caso particular com $n = 5$ entradas, sendo dadas por:

$$\frac{\partial J}{\partial R_{f_l, j}} = - \sum_{k=1}^N \Psi_{f_l}(k) \sum_{h=1}^c e_h(k) \Theta_{f, j, h}(k), \quad f = 1, \dots, n-3 \quad (4.57)$$

$$\frac{\partial J}{\partial R_{(n-2)_l, j}} = - \sum_{k=1}^N \Psi_{(n-2)_l}(k) \sum_{h=1}^c e_h(k) \xi_{(n-2), j, h}(k) \quad (4.58)$$

$$\frac{\partial J}{\partial R_{(n-1)_l, j}} = - \sum_{k=1}^N e_j(k) \Psi_{(n-1)_l}(k) \quad (4.59)$$

onde $l = 1, \dots, c^2$, $j = 1, \dots, c$, $e_j(k) \triangleq Y_j(k) - S_j(k)$ e $\Theta_{f, j, h}(k)$ é definido como

$$\Theta_{f, j, h}(k) \triangleq \sum_{m_1=1}^c \sum_{m_2=1}^c \dots \sum_{m_v=1}^c \lambda_{1, m_1, h}(k) \lambda_{2, m_2, m_1}(k) \dots \lambda_{v, m_v, m_{v-1}}(k) \xi_{f, j, m_v}(k) \quad (4.60)$$

sendo $v \triangleq \max\{n-2-f, 1\}$ e

$$\lambda_{g, m_g, q}(k) \triangleq \sum_{i=0}^{c-1} R_{(n-g)(ic+m_g), q} X_{(n-g+1)(i+1)}(k) \quad (4.61)$$

$$\xi_{g, j, q}(k) \triangleq \sum_{i=0}^{c-1} R_{(g+1)(ic+j), q} X_{(g+2)(i+1)}(k) \quad (4.62)$$

Caso de Estudo: Problema Lógico

Seja o problema de modelagem (classificação) do diagrama lógico apresentado na tabela 4.4 como um conjunto de dados de entrada-saída.

Tabela 4.4: Tabela de Dados (Padrões Lógicos).

k	$x_1(k)$	$x_2(k)$	$x_3(k)$	$y(k)$
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	0	1	1	0
5	1	0	0	1
6	1	0	1	0
7	1	1	0	0
8	1	1	1	1

O conjunto de padrões na tabela 4.4 pode ser gerado por uma composição em cascata de duas portas lógicas do tipo OU-Exclusivo (XOR), como na figura 4.25, ou do tipo OU-Exclusivo Inverso (NXOR), como na figura 4.26 (Taub, 1984).

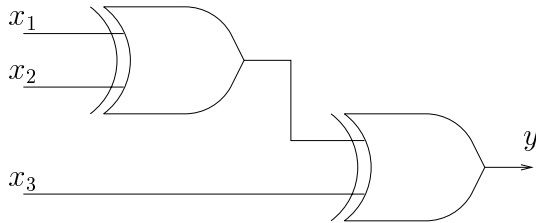


Figura 4.25: Lógica XOR em cascata.

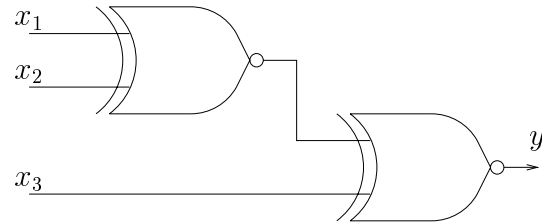


Figura 4.26: Lógica NXOR em cascata.

Utiliza-se um modelo relacional hierárquico para modelagem desse sistema lógico. Como o conjunto de dados é booleano, definem-se dois conjuntos referenciais *fuzzy* ($c = 2$) do tipo *singleton* para cada variável do modelo, como segue³:

$$\mathcal{X}_i(x_i(k)) = \mathcal{Y}_1(y(k)) = \begin{cases} 1, & \text{se } x_i(k), y(k) = 0, \\ 0, & \text{caso contrário.} \end{cases} \quad (4.63)$$

³Para os únicos dois valores que as variáveis assumem, isto é, 0 e 1, é simples verificar que esses conjuntos referenciais possuem a propriedade de interface ótima que permite a utilização do critério de custo (4.53) (ver seção 3.3).

$$\mathcal{X}_{i_2}(x_i(k)) = \mathcal{Y}_2(y(k)) = \begin{cases} 1, & \text{se } x_i(k), y(k) = 1, \\ 0, & \text{caso contrário.} \end{cases} \quad (4.64)$$

aos quais se atribuem os rótulos lingüísticos *Zero* e *Um*, respectivamente.

Os parâmetros relacionais do modelo são treinados por dez iterações do método do gradiente projetado partindo de duas inicializações aleatórias distintas no intervalo $[0, 0.1]$ (ver seção 4.3.4). A evolução do erro quadrático médio entre a saída real e a saída do modelo ao longo do procedimento de otimização é ilustrado na figura 4.27.

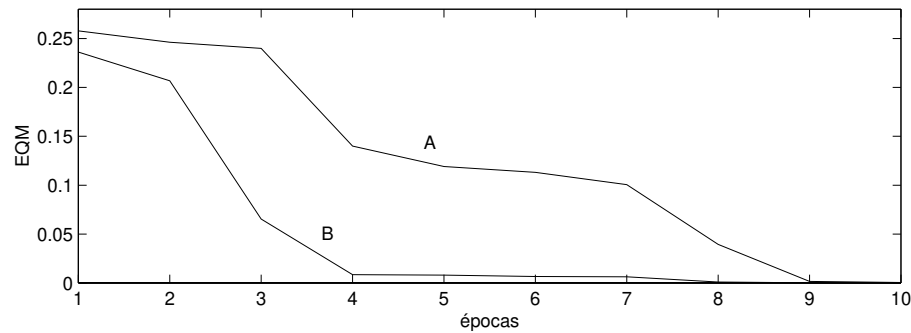


Figura 4.27: Evolução do erro quadrático médio entre a saída real e a saída do modelo para duas inicializações de parâmetros distintos (A e B).

Para as duas inicializações o erro final praticamente se anula, sendo de 4.089×10^{-4} para a inicialização A e 1.357×10^{-5} para a inicialização B. As matrizes relacionais finais correspondentes são dadas nas equações (4.65) e (4.66) abaixo, bem como as respectivas matrizes equivalentes obtidas através da transformação em (4.33).

$$R_1 = \begin{bmatrix} 0 & \mathbf{1.0009} \\ \mathbf{0.9859} & 0.0221 \\ \mathbf{0.9891} & 0.0191 \\ 0 & \mathbf{1.0008} \end{bmatrix} \Rightarrow \Omega = \begin{bmatrix} \mathbf{1.0071} & 0.0219 \\ 0.0197 & \mathbf{1.0071} \\ 0.0192 & \mathbf{1.0078} \\ \mathbf{1.0103} & 0.0192 \\ 0.0222 & \mathbf{1.0046} \\ \mathbf{1.0071} & 0.0222 \\ \mathbf{1.0070} & 0.0219 \\ 0.0197 & \mathbf{1.0070} \end{bmatrix} \quad (4.65)$$

$$R_2 = \begin{bmatrix} 0 & \mathbf{1.0185} \\ \mathbf{1.0062} & 0.0219 \\ \mathbf{1.0211} & 0 \\ 0.0197 & \mathbf{1.0062} \end{bmatrix}$$

$$\begin{array}{l}
R_1 = \begin{bmatrix} \mathbf{1.0114} & 0 \\ 0.0036 & \mathbf{0.9995} \\ 0.0045 & \mathbf{0.9991} \\ \mathbf{1.0116} & 0 \end{bmatrix} \\
R_2 = \begin{bmatrix} \mathbf{0.9911} & 0.0031 \\ 0 & \mathbf{1.0024} \\ 0.0035 & \mathbf{0.9908} \\ \mathbf{1.0021} & 0 \end{bmatrix}
\end{array}
\Rightarrow
\Omega = \begin{bmatrix} \mathbf{1.0023} & 0.0031 \\ 0.0035 & \mathbf{1.0021} \\ 0.0045 & \mathbf{1.0015} \\ \mathbf{1.0012} & 0.0045 \\ 0.0035 & \mathbf{1.0019} \\ \mathbf{1.0016} & 0.0035 \\ \mathbf{1.0025} & 0.0031 \\ 0.0035 & \mathbf{1.0023} \end{bmatrix} \quad (4.66)$$

Pode-se verificar que os elementos mais representativos de cada linha da matriz Ω são os mesmos para ambos os modelos obtidos. Tomando as regras *fuzzy* relacionadas a cada um desses elementos, conforme discutido no capítulo 3 (ver seção 3.4), obtém-se o seguinte conjunto:

SE x_1 é *Zero* e x_2 é *Zero* e x_3 é *Zero* ENTÃO y é *Zero*
SE x_1 é *Zero* e x_2 é *Zero* e x_3 é *Um* ENTÃO y é *Um*
SE x_1 é *Zero* e x_2 é *Um* e x_3 é *Zero* ENTÃO y é *Um*
SE x_1 é *Zero* e x_2 é *Um* e x_3 é *Um* ENTÃO y é *Zero*
SE x_1 é *Um* e x_2 é *Zero* e x_3 é *Zero* ENTÃO y é *Um*
SE x_1 é *Um* e x_2 é *Zero* e x_3 é *Um* ENTÃO y é *Zero*
SE x_1 é *Um* e x_2 é *Um* e x_3 é *Zero* ENTÃO y é *Zero*
SE x_1 é *Um* e x_2 é *Um* e x_3 é *Um* ENTÃO y é *Um*

que é uma representação lingüística exata dos padrões lógicos da tabela 4.4.

Esse exemplo é de particular interesse porque a estrutura em cascata do sistema lógico coincide com a estrutura do modelo hierárquico, permitindo uma análise individual dos submodelos hierárquicos obtidos. Nesse contexto, pode-se verificar através das respectivas matrizes relacionais que a partir das inicializações A e B o modelo converge respectivamente para representações internas das estruturas lógicas nas figuras 4.26 e 4.25, ou seja, para os valores ótimos mais próximos de cada inicialização.

No caso de estudo acima ilustrou-se a extração de conhecimento lingüístico de modelos relacionais hierárquicos possibilitada pela imposição das seguintes restrições estruturais na construção desses modelos:

- Pré-especificação de conjuntos referenciais *fuzzy* lingüisticamente interpretáveis que são mantidos fixos durante o processo de otimização.
- Restrições de não negatividade dos elementos das matrizes relacionais.

Quando não se deseja extrair qualquer conhecimento lingüístico dos modelos obtidos, pode-se otimizar o desempenho quantitativo desses modelos eliminando essas restrições. Nesse caso, pode-se minimizar o critério (4.55) utilizando qualquer técnica de otimização irrestrita (Bazaraa et al., 1993). A minimização pode ser realizada tanto em relação aos parâmetros relacionais como em relação aos parâmetros dos conjuntos referenciais *fuzzy* do modelo. As equações do gradiente da função de custo J_2 em (4.55) com relação aos parâmetros relacionais são dadas por (para $n \geq 3$):

$$\frac{\partial J_2}{\partial R_{f_i,j}} = - \sum_{k=1}^N e(k) \Psi_{f_i}(k) \left(\frac{\sum_{h=1}^c (\phi_h - \hat{y}(k)) \Theta_{f,j,h}(k)}{\sum_{h=1}^c \hat{Y}_{(n-1)_h}(k)} \right), \quad f = 1, \dots, n-3 \quad (4.67)$$

$$\frac{\partial J_2}{\partial R_{(n-2)_i,j}} = - \sum_{k=1}^N e(k) \Psi_{(n-2)_i}(k) \left(\frac{\sum_{h=1}^c (\phi_h - \hat{y}(k)) \xi_{(n-2),j,h}(k)}{\sum_{h=1}^c \hat{Y}_{(n-1)_h}(k)} \right) \quad (4.68)$$

$$\frac{\partial J_2}{\partial R_{(n-1)_i,j}} = - \sum_{k=1}^N e(k) \Psi_{(n-1)_i}(k) \frac{\phi_j - \hat{y}(k)}{\sum_{h=1}^c \hat{Y}_{(n-1)_h}(k)} \quad (4.69)$$

com $l = 1, \dots, c^2$, $j = 1, \dots, c$, $e(k) \triangleq y(k) - \hat{y}(k)$ e os termos $\Theta_{(\cdot)}(k)$ e $\xi_{(\cdot)}(k)$ dados respectivamente pelas equações (4.60) e (4.62).

As equações do gradiente em relação aos conjuntos referenciais *fuzzy* das variáveis de entrada são dadas por:

$$\frac{\partial J_2}{\partial \mathcal{X}_{f_j}(x_f(k))} = - \sum_{k=1}^N e(k) \left(\frac{\sum_{h=1}^c (\phi_h - \hat{y}(k)) \tilde{\Theta}_{f,j,h}(k)}{\sum_{h=1}^c \hat{Y}_{(n-1)_h}(k)} \right), \quad f = 1, \dots, n-1 \quad (4.70)$$

$$\frac{\partial J_2}{\partial \mathcal{X}_{n_j}(x_n(k))} = - \sum_{k=1}^N e(k) \left(\frac{\sum_{h=1}^c (\phi_h - \hat{y}(k)) \tilde{\xi}_{n,j,h}(k)}{\sum_{h=1}^c \hat{Y}_{(n-1)_h}(k)} \right) \quad (4.71)$$

onde $j = 1, \dots, c$ e

$$\tilde{\Theta}_{f,j,h}(k) \triangleq \sum_{m_1=1}^c \sum_{m_2=1}^c \cdots \sum_{m_\nu=1}^c \lambda_{1m_1,h}(k) \lambda_{2m_2,m_1}(k) \cdots \lambda_{\nu m_\nu, m_{\nu-1}}(k) \tilde{\xi}_{f,j,m_\nu}(k) \quad (4.72)$$

com $\nu \triangleq n - \max\{f, 2\}$ e os termos $\lambda_{(\cdot)}(k)$ e $\tilde{\xi}_{(\cdot)}(k)$ dados respectivamente pela equação (4.61) e por:

$$\tilde{\xi}_{g,j,q}(k) \triangleq \begin{cases} \sum_{i=1}^c R_{(g-1)(jc-c+i),q} \hat{Y}_{(g-2)_i}(k) & \text{para } g = 3, \dots, n \\ \sum_{i=1}^c R_{1(jc-c+i),q} X_{1_i}(k) & \text{para } g = 2 \\ \sum_{i=0}^{c-1} R_{1(ic+j),q} X_{2(i+1)}(k) & \text{para } g = 1 \end{cases} \quad (4.73)$$

Supondo que os conjuntos referenciais \mathcal{X}_{ij} ($i = 1, \dots, n$, $j = 1, \dots, c$) possam ser parametrizados em seus centros θ_{ij} e aberturas σ_{ij} , como é o caso daqueles com funções de pertinência Gaussianas, triangulares, tipo sino, entre outros, tem-se:

$$\frac{\partial J_2}{\partial \theta_{ij}} = \frac{\partial J_2}{\partial \mathcal{X}_{ij}(x_i(k))} \frac{\partial \mathcal{X}_{ij}(x_i(k))}{\partial \theta_{ij}} ; \quad \frac{\partial J_2}{\partial \sigma_{ij}} = \frac{\partial J_2}{\partial \mathcal{X}_{ij}(x_i(k))} \frac{\partial \mathcal{X}_{ij}(x_i(k))}{\partial \sigma_{ij}} \quad (4.74)$$

sendo que os termos $\partial \mathcal{X}_{ij}(x_i(k)) / \partial \theta_{ij}$ e $\partial \mathcal{X}_{ij}(x_i(k)) / \partial \sigma_{ij}$ podem ser calculados numericamente, independente do tipo de função de pertinência, ou analiticamente para um tipo específico.

O critério J_2 em (4.55) independe de qualquer fuzzificação da saída do sistema dada pela equação (4.54) e, por conseqüência, é função dos conjuntos referenciais *fuzzy* de saída $\mathcal{Y}_{(\cdot)}$ apenas em relação aos seus valores modais $\phi_{(\cdot)}$ na equação (4.27). As derivadas parciais de J_2 em relação a esses parâmetros são dadas por:

$$\frac{\partial J_2}{\partial \phi_j} = - \sum_{k=1}^N \left(\frac{e(k) \hat{Y}_{(n-1)_j}(k)}{\sum_{h=1}^c \hat{Y}_{(n-1)_h}(k)} \right) \quad (4.75)$$

As equações de (4.67) até (4.75) constituem o conjunto de equações para o cálculo do gradiente da função de custo J_2 em relação aos parâmetros de projeto livre dos modelos hierárquicos, suficiente para o procedimento de otimização desses modelos.

Observação 4.3 Assim como para os modelos hierárquicos com submodelos relacionais simplificados, discutidos na seção 4.2, o conjunto de equações de (4.67) a (4.75) foi obtido assumindo que a quantidade c de conjuntos referenciais fuzzy por variável de entrada é a mesma para cada variável, o que simplifica o conjunto de equações e o seu desenvolvimento. Como naquele caso, se um número distinto de conjuntos referenciais for selecionado para cada variável, o mesmo conjunto de equações pode ser utilizado considerando c como o maior valor entre as diferentes quantidades de conjuntos atribuídos às variáveis. Deve-se então retirar dos parâmetros a serem otimizados os centros e aberturas referentes aos conjuntos referenciais “indesejados” bem como os respectivos elementos relacionais, que devem ser igualados a zero para anular as contribuições referentes aos conjuntos referenciais descartados nos cálculos da saída do modelo e das derivadas relacionadas aos demais parâmetros.

4.3.4 Normalização de Dados e Inicialização de Parâmetros

Os aspectos referentes à normalização de dados e inicialização de parâmetros nos modelos hierárquicos com submodelos relacionais convencionais são basicamente os mesmos discutidos na seção 4.2.4 para submodelos simplificados. A única exceção diz respeito à ausência dos conjuntos referenciais fuzzy das variáveis intermediárias, que não necessitam mais ser especificados. Por conseqüência, não existe a preocupação com a saturação desses conjuntos na inicialização dos parâmetros relacionais (matrizes $R_{(\cdot)}$). Apenas quando deseja-se a interpretação lingüística dos modelos, é necessário atribuir valores a esses parâmetros de tal forma que o vetor de saída final do modelo \hat{Y}_{n-1} não exceda grosseiramente o intervalo de valores fuzzy $[0, 1]$. Nesse caso, deve-se ainda selecionar os conjuntos referenciais fuzzy das variáveis de entrada e saída de forma a satisfazer as condições de interpretabilidade discutidas no capítulo 3.

4.3.5 Exemplo: Problema de Classificação *Iris*

Considera-se novamente o problema de classificação *Iris* apresentado na seção 4.2.8. No presente exemplo, no entanto, avalia-se apenas a modelagem qualitativa do conjunto de dados, realizada através dos modelos hierárquicos com submodelos relacionais convencionais. Para a avaliação dos resultados, toma-se como referência um conjunto de regras lingüísticas obtido a partir desses dados através do método de projeção de fuzzy clusters (Sugeno e Yasukawa, 1993). Conforme discutido a seguir, esse método mostra-se adequado para esse problema em particular.

O método citado acima baseia-se na utilização do bem conhecido algoritmo de clustering denominado *Fuzzy C-Means* (FCM) (Bezdek, 1981). Esse algoritmo determina, através da minimização de uma função objetivo do tipo não supervisionada, o valor de pertinência fuzzy de cada amostra dos dados a um determinado grupo (*cluster*). A quantidade de grupos é pré-especificada e a somatória das pertinências de cada amostra aos diferentes grupos é imposta unitária através de restrições no problema de otimização.

O conjunto dos valores de pertinência, no domínio n -dimensional (nD) dos dados, em relação a um determinado grupo, é denominado *fuzzy cluster*.

No algoritmo proposto em (Sugeno e Yasukawa, 1993), os *fuzzy clusters* são projetados nos componentes unidimensionais do seu domínio e conjuntos *fuzzy* são otimizados para representar essas projeções de uma forma parametrizada adequada. A aplicação desse método no conjunto de dados *Iris* é ilustrada nas figuras 4.28 a 4.31. Nessas figuras, os conjuntos de pontos representam as projeções de cada um dos três *clusters*⁴ em cada uma das dimensões referentes às variáveis do problema. As variáveis x_i representam respectivamente os comprimentos e larguras das sépalas (x_1 e x_2) e pétalas (x_3 e x_4) das *Irises*. Junto às referidas projeções, apresentam-se também nas figuras conjuntos *fuzzy* triangulares otimizados para melhor representá-las. O tipo de conjunto, central ou extremo (i.e. normal à esquerda/direita), é definido pela ordem dos centros dos *clusters* projetados em cada dimensão.

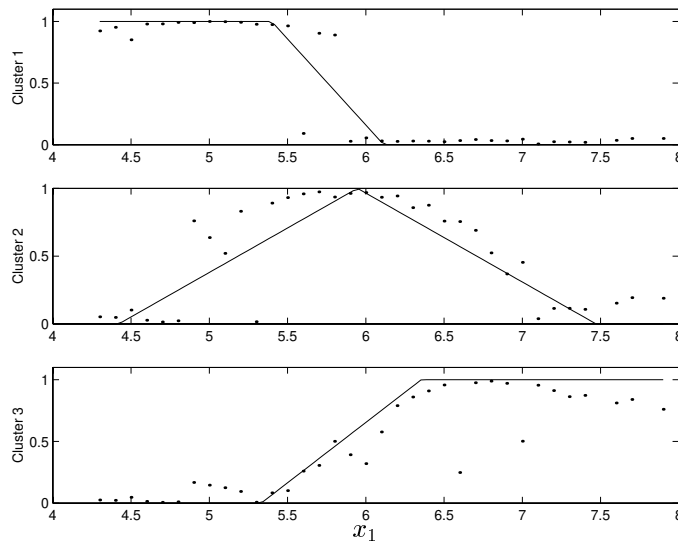


Figura 4.28: Projeções dos três *clusters* no componente x_1 do domínio 4D e respectivas aproximações triangulares: Conjuntos *fuzzy* \mathcal{A}_1 , \mathcal{B}_1 e \mathcal{C}_1 para a entrada x_1 (de cima para baixo).

Em um problema de classificação, a associação de cada *fuzzy cluster* a uma classe específica pode ser realizada tomando a amostra ou amostras com maior pertinência àquele *cluster* e verificando a classe a que ela pertence. Nas figuras 4.28 a 4.31, os *clusters* 1, 2 e 3 estão associados respectivamente às classes *setosa*, *versicolor* e *virginica*. Supondo que as classes são “bem comportadas”, isto é, que representam regiões aproximadamente convexas do domínio a partir, por exemplo, de distribuições Gaussianas das suas características fenotípicas (o que é biologicamente consistente), pode-se

⁴O algoritmo FCM foi executado com 3 *clusters*, pois nesse caso sabe-se *a priori* que o problema em estudo possui três classes.

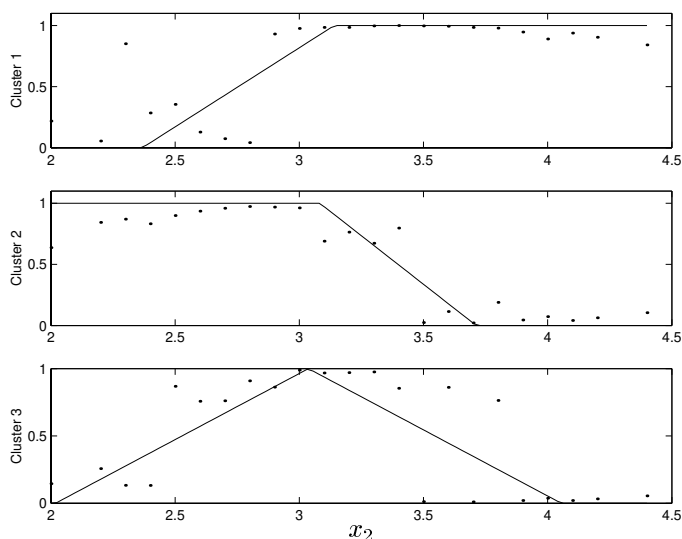


Figura 4.29: Projeções dos três *clusters* no componente x_2 do domínio 4D e respectivas aproximações triangulares: Conjuntos *fuzzy* \mathcal{C}_2 , \mathcal{A}_2 e \mathcal{B}_2 para a entrada x_2 (de cima para baixo).

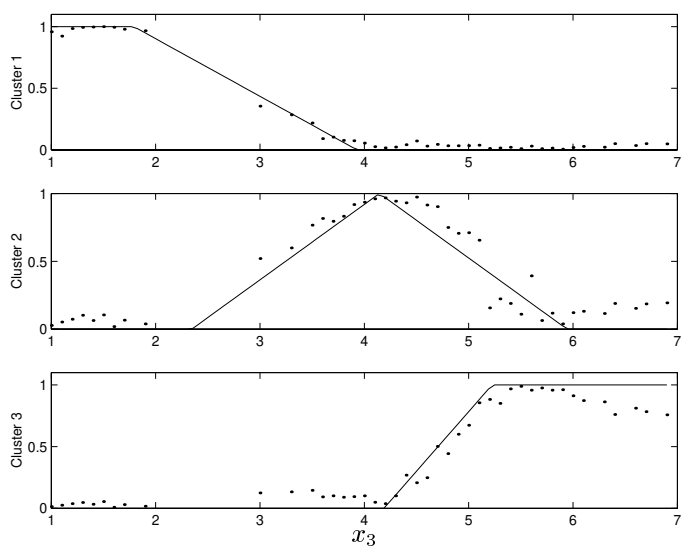


Figura 4.30: Projeções dos três *clusters* no componente x_3 do domínio 4D e respectivas aproximações triangulares: Conjuntos *fuzzy* \mathcal{A}_3 , \mathcal{B}_3 e \mathcal{C}_3 para a entrada x_3 (de cima para baixo).

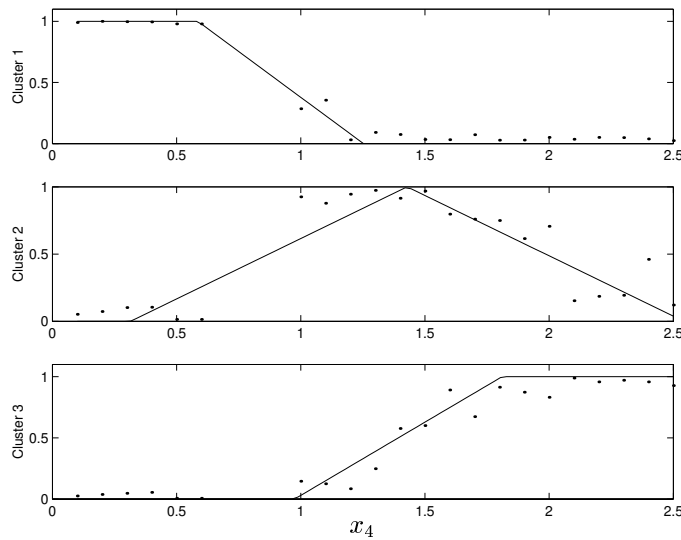


Figura 4.31: Projeções dos três *clusters* no componente x_4 do domínio 4D e respectivas aproximações triangulares: Conjuntos *fuzzy* \mathcal{A}_4 , \mathcal{B}_4 e \mathcal{C}_4 para a entrada x_4 (de cima para baixo).

considerar que a tabela 4.5 representa uma descrição qualitativa básica dos dados *Iris*. Essa hipótese também parece razoável a partir da observação dos dados e das próprias projeções dos *clusters*, que indicam algo similar ao ilustrado simplificada na figura 4.32 para o caso bidimensional ($x_1 \times x_3$).

Tabela 4.5: Regras lingüísticas referenciais para o problema *Iris*.

x_1	x_2	x_3	x_4	<i>cluster</i>
\mathcal{A}_1	\mathcal{C}_2	\mathcal{A}_3	\mathcal{A}_4	1
\mathcal{B}_1	\mathcal{A}_2	\mathcal{B}_3	\mathcal{B}_4	2
\mathcal{C}_1	\mathcal{B}_2	\mathcal{C}_3	\mathcal{C}_4	3

A tabela 4.5 foi obtida a partir dos conjuntos *fuzzy* apresentados nas figuras 4.28 a 4.31. A primeira linha, por exemplo, representa a regra lingüística “Se x_1 é \mathcal{A}_1 e x_2 é \mathcal{C}_2 e x_3 é \mathcal{A}_3 e x_4 é \mathcal{A}_4 ENTÃO a *Iris* é *sestosa*”. Essa e as demais regras são qualitativamente equivalentes àquelas regras obtidas em (Wu e Chen, 1999) para o mesmo problema, e serão tomadas como referencial para os desenvolvimentos subseqüentes utilizando os modelos hierárquicos.

Considera-se a seguir um modelo hierárquico com quatro entradas para modelagem qualitativa dos dados *Iris*. Para fins de comparação com a base de regras referencial da tabela 4.5, adotam-se os mesmos conjuntos *fuzzy* daquela tabela para as variáveis de entrada do modelo, isto é, aqueles conjuntos triangulares ilustrados nas figuras 4.28 a

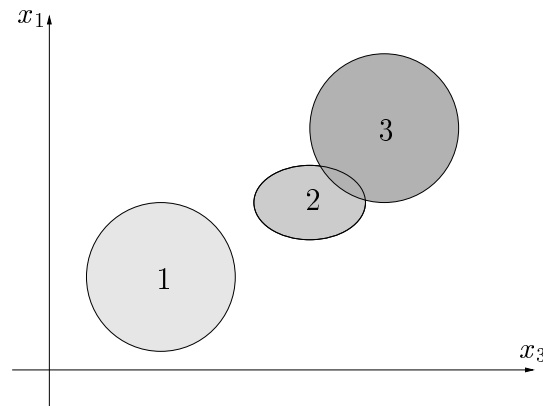


Figura 4.32: Representação simplificada de classes bem comportadas em um domínio 2D.

4.31 ($c = 3$). Uma vez que os dados de saída gerados artificialmente (ver seção 4.3.5) assumem apenas os valores -1 , 0 e 1 associados às classes *sestosa*, *versicolor* e *virginica*, respectivamente, utilizam-se três conjuntos referenciais *fuzzy* do tipo *singleton* para a variável de saída do modelo. Procedimento análogo foi adotado no caso de estudo da seção 4.3.3.

Todos os conjuntos *fuzzy* são mantidos fixos durante o procedimento de otimização do modelo, que é realizado apenas sobre os seus parâmetros relacionais através do método do gradiente projetado. Os parâmetros relacionais são inicializados aleatoriamente no intervalo $[0, 0.1]$, conforme discutido na seção 4.3.4. A figura 4.33-a apresenta a evolução do erro quadrático médio entre os dados de saída (artificiais) e as respectivas saídas do modelo ao longo do procedimento de otimização. As saídas inferidas a partir da aplicação de todo o conjunto de dados de entrada no modelo final obtido (cujo erro associado é de $EQM=0.0334$) são ilustradas na figura 4.33-b. Com base nessas saídas, o procedimento de classificação descrito na seção 4.3.5 produz 145 padrões classificados corretamente e apenas 5 de forma incorreta. Esse resultado é levemente inferior aos obtidos naquela seção a partir dos modelos hierárquicos com submodelos simplificados, basicamente por duas razões: A primeira é que, como o interesse principal no presente contexto diz respeito à modelagem qualitativa e não quantitativa dos dados, os conjuntos referenciais *fuzzy* das variáveis de entrada e saída não são otimizados. A segunda razão é que, pelo mesmo motivo anterior, os parâmetros relacionais estão sujeitos a restrições de não negatividade durante o procedimento de otimização. Nessas circunstâncias, os resultados quantitativos obtidos podem ser considerados satisfatórios.

A partir dos parâmetros otimizados do modelo final, referentes às matrizes relacionais de cada um dos seus submodelos hierárquicos, obtém-se a matriz relacional Ω do seu modelo não hierárquico equivalente utilizando a transformação de equivalência apresentada no teorema 4.1. Em seguida, aplica-se sobre essa matriz a metodologia de extração de regras lingüísticas discutida no capítulo 3. Especificamente, para cada elemento da matriz Ω (5D) calcula-se a medida de representabilidade condicionada (aos

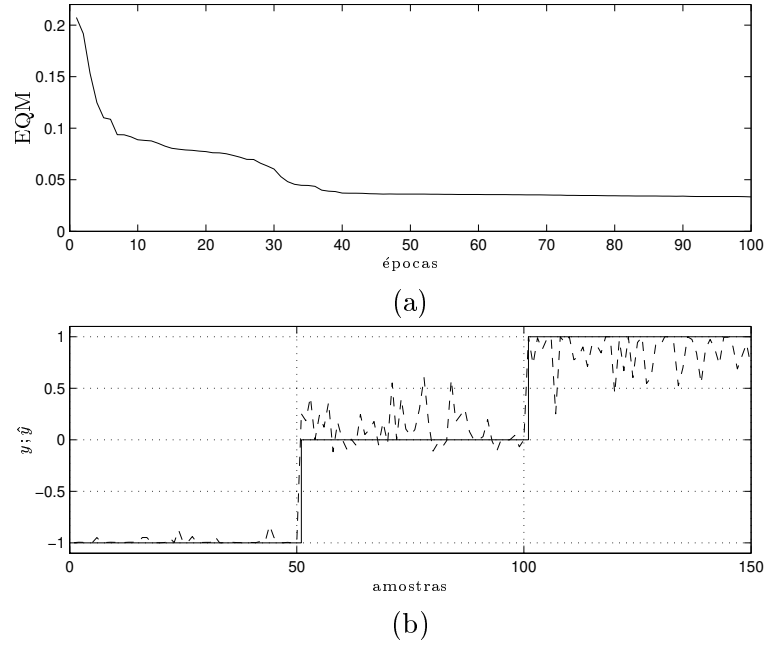


Figura 4.33: (a) Evolução do EQM ao longo do treinamento; (b) Saída do modelo (linha tracejada) e saídas artificiais (linha cheia) associadas às classes *setosa* (-1), *versicolor* (0) e *virginica* ($+1$).

dados *Iris*) definida na seção 3.4.2, lembrando que os dados estão distribuídos igualmente entre as três classes. A matriz de representabilidade condicionada Φ resultante é então normalizada no intervalo $[0, 1]$ para facilitar sua análise. Tem-se que, das 243 regras lingüísticas possíveis ($c^4 \times c$), cada uma representada por um elemento dessa matriz, 184 regras são nulas (representabilidade igual a zero).

Observação 4.4 *Existem interseções entre os conjuntos fuzzy e entre as próprias classes (ver figuras 4.28 a 4.31), de forma que não se pode esperar que apenas 3 regras do modelo, referentes às três classes do problema, possuam representabilidade não nula.*

Aquelas submatrizes 2D de Φ que contém elementos não nulos são mostradas abaixo.

$$\Phi_{i,j,1,1,1} = \begin{bmatrix} \mathbf{0.2506} & 0 & \mathbf{1} \\ 0.0266 & 0 & \mathbf{0.1471} \\ 0.0027 & 0.0002 & 0.0330 \end{bmatrix} \quad (4.76)$$

$$\Phi_{i,j,2,1,1} = \begin{bmatrix} 0.0053 & 0 & 0.0003 \\ 0.0027 & 0 & 0 \\ 0.0016 & 0 & 0.0003 \end{bmatrix} \quad (4.77)$$

$$\Phi_{i,j,3,1,1} = \begin{bmatrix} 0.0002 & 0 & 0.0000 \\ 0.0003 & 0 & 0.0000 \\ 0.0004 & 0.0000 & 0.0002 \end{bmatrix} \quad (4.78)$$

$$\Phi_{i,j,1,1,2} = \begin{bmatrix} 0.0036 & 0 & 0.0021 \\ 0.0005 & 0 & 0 \\ 0.0000 & 0 & 0.0001 \end{bmatrix} \quad (4.79)$$

$$\Phi_{i,j,2,1,2} = \begin{bmatrix} 0.0261 & 0 & 0.0022 \\ 0.0131 & 0 & 0.0004 \\ 0.0089 & 0.0000 & 0.0018 \end{bmatrix} \quad (4.80)$$

$$\Phi_{i,j,1,2,2} = \begin{bmatrix} 0.0173 & 0 & 0.0169 \\ 0.0042 & 0 & 0.0028 \\ 0.0016 & 0.0000 & 0.0012 \end{bmatrix} \quad (4.81)$$

$$\Phi_{i,j,2,2,2} = \begin{bmatrix} \mathbf{0.3203} & 0 & 0.0703 \\ \mathbf{0.3205} & 0 & 0.0168 \\ \mathbf{0.4175} & 0.0012 & \mathbf{0.2306} \end{bmatrix} \quad (4.82)$$

$$\Phi_{i,j,1,3,3} = \begin{bmatrix} 0.0010 & 0 & 0.0009 \\ 0.0003 & 0 & 0.0003 \\ 0.0003 & 0.0000 & 0.0002 \end{bmatrix} \quad (4.83)$$

$$\Phi_{i,j,3,3,3} = \begin{bmatrix} 0.0640 & 0 & 0.0241 \\ \mathbf{0.2119} & 0 & 0.0438 \\ \mathbf{0.6279} & 0.0050 & \mathbf{0.4899} \end{bmatrix} \quad (4.84)$$

Nota-se que todas as regras relacionadas ao conjunto *fuzzy* central da variável de entrada x_2 (conjunto \mathcal{B}_2 , associado às colunas centrais das submatrizes acima) são nulas ou praticamente nulas. Essa condição reflete o fato de que o referido conjunto é redundante, no sentido de que não existe amostra nos dados que apresente pertinência maior a ele que aos demais (\mathcal{A}_2 e \mathcal{C}_2), conforme mostra a figura 4.29. Por ser redundante, esse conjunto *fuzzy* poderia em tese ser desprezado. Esse resultado foi obtido pelo procedimento de otimização do modelo através da anulação dos parâmetros relacionais correspondentes.

Das 59 regras não nulas (25% do total de 243), apenas dez regras (4% do total) podem ser consideradas significativas tomando como critério uma medida de representabilidade maior ou igual a 10% da regra mais representativa (representabilidade unitária). Essas dez regras mais representativas estão em destaque nas submatrizes de (4.76) a (4.84) e serão qualitativamente avaliadas na seqüência.

A principal conseqüência da pouca distinção entre as classes com relação à variável x_2 , conforme ilustra a figura 4.29, que resulta na nulidade do conjunto \mathcal{B}_2 como já foi discutido, é que a terceira regra referencial da tabela 4.5 terminou por ser representada no modelo de forma compartilhada entre as regras vizinhas (com relação a x_2) apresentadas na tabela 4.6. Essas regras correspondem aos elementos $\Phi_{3,1,3,3,3}$ e $\Phi_{3,3,3,3,3}$ em (4.84), isto é, aqueles com segunda e terceira maiores medidas de representabilidade condicionada.

Tabela 4.6: Regras lingüísticas extraídas dos dados *Iris* (2ª e 3ª maiores representabilidades).

x_1	x_2	x_3	x_4	<i>cluster</i>
\mathcal{C}_1	\mathcal{A}_2	\mathcal{C}_3	\mathcal{C}_4	3
\mathcal{C}_1	\mathcal{C}_2	\mathcal{C}_3	\mathcal{C}_4	3

Fato similar ocorre em relação à primeira variável de entrada, x_1 . Nesse caso, a segunda regra referencial da tabela 4.5, que envolve o conjunto *fuzzy* central dessa variável (conjunto \mathcal{B}_1), é representada não apenas pela regra correspondente do modelo, mas também de forma compartilhada pelas regras vizinhas (com relação a x_1). Essas três regras, que correspondem aos elementos $\Phi_{3,1,2,2,2}$, $\Phi_{2,1,2,2,2}$ e $\Phi_{1,1,2,2,2}$ em (4.82), isto é, aqueles com quarta, quinta e sexta maiores medidas de representabilidade condicionada, são apresentadas na tabela 4.7.

Tabela 4.7: Regras lingüísticas extraídas dos dados *Iris* (4ª, 5ª e 6ª maiores representabilidades).

x_1	x_2	x_3	x_4	<i>cluster</i>
\mathcal{C}_1	\mathcal{A}_2	\mathcal{B}_3	\mathcal{B}_4	2
\mathcal{B}_1	\mathcal{A}_2	\mathcal{B}_3	\mathcal{B}_4	2
\mathcal{A}_1	\mathcal{A}_2	\mathcal{B}_3	\mathcal{B}_4	2

Finalmente, tem-se que a regra mais representativa do modelo (elemento $\Phi_{1,3,1,1,1}$ em (4.76)) coincide exatamente com aquela regra referencial que descreve a primeira classe na tabela 4.5, justamente a classe referente à *Iris setosa* que é aquela mais bem definida

(separável) do problema. As demais quatro regras com alguma representabilidade, além das seis já comentadas acima, são todas vizinhas⁵ dessas últimas, o que é justificado através da observação 4.4.

Todos os resultados acima foram reproduzidos de forma muito similar considerando conjuntos *fuzzy* Gaussianos ao invés de triangulares e diferentes inicializações paramétricas, confirmando que os modelos hierárquicos obtidos proporcionam conhecimento qualitativo consistente a respeito do sistema. Também obteve-se o mesmo resultado qualitativo alterando a ordem hierárquica das variáveis de entrada do modelo, isto é, trocando x_1 e x_2 (comprimento e largura das sépalas) por x_3 e x_4 (comprimento e largura das pétalas), respectivamente. Nesse caso, no entanto, o desempenho quantitativo foi inferior (EQM das estimativas igual a 0.0477 e 9 classificações incorretas para uma inicialização/simulação típica). Esse resultado pode ter ocorrido porque nesse caso as variáveis posicionadas nos níveis hierárquicos mais baixos (comprimento e largura das sépalas) concentram a maior complexidade do problema de separação pois as classes não estão tão bem definidas em relação a elas como em relação às demais (ver figuras 4.28 e 4.29).

Uma questão que se coloca é porque não adotar sempre um método simples como aquele que gerou a base referencial de regras da tabela 4.5. Como foi visto, aquela base de regras faz sentido somente a partir da hipótese de que as classes são bem comportadas, como aquelas ilustradas na figura 4.32. Em problemas mais complexos, no entanto, como por exemplo aquele ilustrado na figura 4.34, a abordagem relacional (hierárquica ou não) é mais adequada porque particiona o domínio do problema em diversas regiões difusas através da distribuição dos conjuntos *fuzzy* sobre cada um de seus componentes unidimensionais e então estima os elementos relacionais associados às regras que relacionam cada uma das partições em cada uma das diferentes classes.

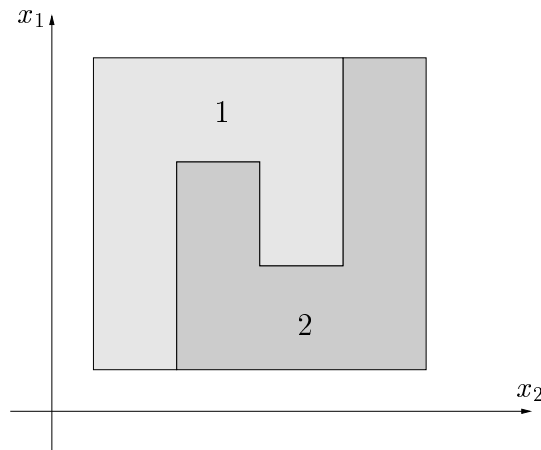


Figura 4.34: Representação simplificada de classes não convexas e não linearmente separáveis em um domínio 2D.

⁵Para duas delas essa afirmação toma como base a nulidade do conjunto *fuzzy* B_2 .

4.4 Aproximação Universal

Nessa seção discute-se a capacidade de aproximação dos modelos relacionais com estrutura hierárquica. Inicialmente analisam-se os modelos apresentados na seção 4.3, com submodelos relacionais convencionais, e em seguida os modelos discutidos na seção 4.2, com submodelos relacionais simplificados.

Estrutura de submodelos relacionais convencionais

A seguir mostra-se que os modelos relacionais hierárquicos com submodelos convencionais e conjuntos referenciais *fuzzy* Gaussianos para as variáveis de entrada, ou seja,

$$\mathcal{X}_{i_j}(x_i) = \exp\left(-\frac{(x_i - \theta_{i_j})^2}{\sigma_{i_j}^2}\right) \quad (4.85)$$

são *aproximadores universais*, isto é, podem aproximar com precisão arbitrária qualquer função real contínua sobre um domínio compacto.

Os parâmetros de projeto da estrutura hierárquica de submodelos relacionais convencionais, descrita pelas equações (4.26) até (4.30) e (4.85), são dados por:

- O número “ c ” de conjuntos referenciais *fuzzy* Gaussianos por variável.
- Os parâmetros relacionais, ou seja, as matrizes R_i ($i = 1, \dots, n - 1$) dos submodelos hierárquicos.
- Os centros θ_{i_j} e as aberturas $\sigma_{i_j} \neq 0$ ($i = 1, \dots, n$; $j = 1, \dots, c$) dos conjuntos referenciais das variáveis de entrada.
- Os valores modais ϕ_h ($h = 1, \dots, c$) dos conjuntos referenciais das variáveis de saída.

Seja \mathcal{M} o conjunto de todos os modelos possíveis com essa arquitetura hierárquica, descritos genericamente em função do seu vetor $\mathbf{x} = [x_1 \cdots x_n]^T$ de entradas como $\hat{y}(\mathbf{x}) : U \subset \mathfrak{R}^n \rightarrow \mathfrak{R}$, onde U é uma região compacta (fechada e limitada) do espaço n -dimensional de entrada. O conjunto \mathcal{M} é dito *denso* no conjunto $\mathcal{C}[U]$ de todas as funções reais contínuas em U se, para qualquer $g(\mathbf{x}) \in \mathcal{C}[U]$ e um escalar arbitrário $\epsilon > 0$, existe $\hat{y}(\mathbf{x}) \in \mathcal{M}$ tal que (Kreysig, 1978; Luenberger, 1997):

$$\sup_{\mathbf{x} \in U} |g(\mathbf{x}) - \hat{y}(\mathbf{x})| < \epsilon \quad (4.86)$$

Essa condição pode ser verificada utilizando o teorema de Stone-Weierstrass (Rudin, 1964), apresentado abaixo.

Teorema de Stone-Weierstrass: *Seja \mathcal{Z} um conjunto de funções reais contínuas sobre um conjunto compacto U . Se: 1) \mathcal{Z} é uma álgebra, i.e., o conjunto \mathcal{Z} é fechado*

sob as operações de adição, multiplicação e produto escalar; 2) \mathcal{Z} separa pontos em U , i.e., para cada $\mathbf{x}', \mathbf{x}'' \in U$, $\mathbf{x}' \neq \mathbf{x}''$, existe $f \in \mathcal{Z}$ tal que $f(\mathbf{x}') \neq f(\mathbf{x}'')$; e 3) \mathcal{Z} não se anula em U , i.e., para cada $\mathbf{x} \in U$ existe $f \in \mathcal{Z}$ tal que $f(\mathbf{x}) \neq 0$, então o fecho uniforme de \mathcal{Z} consiste de todas as funções reais contínuas em U , ou seja, \mathcal{Z} é denso no conjunto $\mathcal{C}[U]$ de todas as funções reais contínuas em U .

Logo, para provar que os modelos hierárquicos apresentam a propriedade de aproximação universal dada em (4.86), deve-se demonstrar que o conjunto \mathcal{M} satisfaz as condições do teorema acima. Por simplicidade e sem perda de generalidade serão utilizados modelos com 3 entradas ($n = 3$) de tal forma que \mathcal{M} nesse caso é o conjunto de todos os modelos hierárquicos $\hat{y}(\mathbf{x}) : U \subset \mathbb{R}^3 \rightarrow \mathbb{R}$ que podem ser escritos a partir das equações (4.26) a (4.30) como

$$\hat{y}(\mathbf{x}) = \frac{\sum_{h=1}^c \hat{Y}_{2h} \phi_h}{\sum_{h=1}^c \hat{Y}_{2h}} \quad (4.87)$$

$$\hat{Y}_{2h} = \sum_{i=1}^c \sum_{j=1}^c \tilde{R}_{2_{i,j,h}} \mathcal{X}_{3_i}(x_3) \hat{Y}_{1_j} \quad (4.88)$$

$$\hat{Y}_{1_j} = \sum_{p=1}^c \sum_{q=1}^c \tilde{R}_{1_{p,q,j}} \mathcal{X}_{2_p}(x_2) \mathcal{X}_{1_q}(x_1) \quad (4.89)$$

sendo os termos $\tilde{R}_{(\cdot)}$ os mesmos já definidos em (4.33). Como \mathcal{M} é por definição um conjunto de modelos reais contínuos em um domínio compacto U , deve-se provar que esse conjunto é uma álgebra, separa pontos e não se anula nesse domínio. Para tanto, sejam dois modelos $\hat{y}'(\mathbf{x}), \hat{y}''(\mathbf{x}) \in \mathcal{M}$. Das equações acima pode-se escrevê-los como:

$$\hat{y}'(\mathbf{x}) = \frac{\sum_{h1=1}^{c1} \sum_{i1=1}^{c1} \sum_{j1=1}^{c1} \sum_{p1=1}^{c1} \sum_{q1=1}^{c1} \phi'_{h1} \tilde{R}'_{2_{i1,j1,h1}} \tilde{R}'_{1_{p1,q1,j1}} \mathcal{X}'_{3_{i1}}(x_3) \mathcal{X}'_{2_{p1}}(x_2) \mathcal{X}'_{1_{q1}}(x_1)}{\sum_{h1=1}^{c1} \sum_{i1=1}^{c1} \sum_{j1=1}^{c1} \sum_{p1=1}^{c1} \sum_{q1=1}^{c1} \tilde{R}'_{2_{i1,j1,h1}} \tilde{R}'_{1_{p1,q1,j1}} \mathcal{X}'_{3_{i1}}(x_3) \mathcal{X}'_{2_{p1}}(x_2) \mathcal{X}'_{1_{q1}}(x_1)} \quad (4.90)$$

$$\hat{y}''(\mathbf{x}) = \frac{\sum_{h2=1}^{c2} \sum_{i2=1}^{c2} \sum_{j2=1}^{c2} \sum_{p2=1}^{c2} \sum_{q2=1}^{c2} \phi''_{h2} \tilde{R}''_{2_{i2,j2,h2}} \tilde{R}''_{1_{p2,q2,j2}} \mathcal{X}''_{3_{i2}}(x_3) \mathcal{X}''_{2_{p2}}(x_2) \mathcal{X}''_{1_{q2}}(x_1)}{\sum_{h2=1}^{c2} \sum_{i2=1}^{c2} \sum_{j2=1}^{c2} \sum_{p2=1}^{c2} \sum_{q2=1}^{c2} \tilde{R}''_{2_{i2,j2,h2}} \tilde{R}''_{1_{p2,q2,j2}} \mathcal{X}''_{3_{i2}}(x_3) \mathcal{X}''_{2_{p2}}(x_2) \mathcal{X}''_{1_{q2}}(x_1)} \quad (4.91)$$

Uma vez que considerou-se o uso de conjuntos *fuzzy* Gaussianos (ver equação (4.85)), tem-se que o conjunto \mathcal{M} é constituído de funções reais contínuas dadas pela média ponderada do produto de funções Gaussianas em cada um dos seus argumentos (variáveis de entrada dos modelos).

Proposição 4.1 *O conjunto \mathcal{M} é uma álgebra.*

Prova:

A soma de dois modelos $\hat{y}'(\mathbf{x}), \hat{y}''(\mathbf{x}) \in \mathcal{M}$ é dada a partir de (4.90) e (4.91) por:

$$\begin{aligned} \hat{y}'(\mathbf{x}) + \hat{y}''(\mathbf{x}) &= \\ &= \frac{\sum_{h1=1}^{c1} \sum_{i1=1}^{c1} \sum_{j1=1}^{c1} \sum_{p1=1}^{c1} \sum_{q1=1}^{c1} \sum_{h2=1}^{c2} \sum_{i2=1}^{c2} \sum_{j2=1}^{c2} \sum_{p2=1}^{c2} \sum_{q2=1}^{c2} (\phi'_{h1} + \phi''_{h2}) \tilde{R}'_{2_{i1,j1,h1}} \tilde{R}'_{1_{p1,q1,j1}}}{\sum_{h1=1}^{c1} \sum_{i1=1}^{c1} \sum_{j1=1}^{c1} \sum_{p1=1}^{c1} \sum_{q1=1}^{c1} \sum_{h2=1}^{c2} \sum_{i2=1}^{c2} \sum_{j2=1}^{c2} \sum_{p2=1}^{c2} \sum_{q2=1}^{c2} \tilde{R}'_{2_{i1,j1,h1}} \tilde{R}'_{1_{p1,q1,j1}}} \quad (4.92) \\ &= \frac{\tilde{R}''_{2_{i2,j2,h2}} \tilde{R}''_{1_{p2,q2,j2}} \mathcal{X}'_{3_{i1}}(x_3) \mathcal{X}''_{3_{i2}}(x_3) \mathcal{X}'_{2_{p1}}(x_2) \mathcal{X}''_{2_{p2}}(x_2) \mathcal{X}'_{1_{q1}}(x_1) \mathcal{X}''_{1_{q2}}(x_1)}{\tilde{R}''_{2_{i2,j2,h2}} \tilde{R}''_{1_{p2,q2,j2}} \mathcal{X}'_{3_{i1}}(x_3) \mathcal{X}''_{3_{i2}}(x_3) \mathcal{X}'_{2_{p1}}(x_2) \mathcal{X}''_{2_{p2}}(x_2) \mathcal{X}'_{1_{q1}}(x_1) \mathcal{X}''_{1_{q2}}(x_1)} \end{aligned}$$

Dado que o produto de duas funções Gaussianas definidas para uma mesma variável x_i como em (4.85) pode ser escrito como o produto entre um escalar e uma outra função Gaussiana definida da mesma forma, tem-se que $\mathcal{X}'_{i_{(\cdot)}}(x_i) \mathcal{X}''_{i_{(\cdot)}}(x_i)$ é também uma função Gaussiana em x_i assim como os termos individuais desse produto, exceto por um fator proporcional constante. Logo, a equação (4.92) é dada por uma média ponderada do produto de funções Gaussianas de cada um dos elementos do vetor \mathbf{x} . Essa equação é portanto equivalente em forma àquelas em (4.90) e (4.91), o que implica $\hat{y}'(\mathbf{x}) + \hat{y}''(\mathbf{x}) \in \mathcal{M}$. É importante observar que a quantidade de termos no numerador e denominador não importa pois constitui um dos parâmetros de projeto da arquitetura.

O produto de dois modelos $\hat{y}'(\mathbf{x})$ e $\hat{y}''(\mathbf{x})$ em \mathcal{M} é obtido a partir de (4.90) e (4.91) de forma similar à soma apresentada em (4.92), apenas substituindo o termo $(\phi'_{h1} + \phi''_{h2})$ naquela equação por $(\phi'_{h1} \cdot \phi''_{h2})$. Analogamente ao caso anterior, tem-se que $\hat{y}'(\mathbf{x}) \cdot \hat{y}''(\mathbf{x}) \in \mathcal{M}$.

Finalmente tem-se que o produto $\alpha \hat{y}'(\mathbf{x})$, onde α é um escalar qualquer, é equivalente à equação (4.90) apenas com esse escalar adicional multiplicando os termos da somatória no numerador daquela equação. Novamente, com base nos mesmos argumentos anteriores, tem-se que $\alpha \hat{y}'(\mathbf{x}) \in \mathcal{M}$.

Portanto, tem-se que o conjunto \mathcal{M} é fechado com relação às operações de adição, multiplicação e produto escalar, o que completa a prova. ■

Proposição 4.2 *O conjunto \mathcal{M} não se anula em U .*

Prova:

Como os conjuntos *fuzzy* (funções Gaussianas) só podem assumir valores positivos para qualquer $\mathbf{x} \in U$, tem-se que qualquer modelo em \mathcal{M} com todos os parâmetros de projeto especificados como sendo positivos só resultará em valores de saída positivos, de tal forma que para $\forall \mathbf{x} \in U$ existe $\hat{y}(\mathbf{x}) \in \mathcal{M}$ tal que $\hat{y}(\mathbf{x}) \neq 0$. ■

Proposição 4.3 *O conjunto \mathcal{M} separa pontos em U .*

Prova:

A prova nesse caso é obtida construindo um modelo específico que garante a propriedade desejada. Primeiro seleciona-se $c = 2$. Escolhendo

$$\tilde{R}_{1p,q,j} = \begin{cases} 1 & \text{para } p = q = j, \\ 0 & \text{caso contrário.} \end{cases} \quad (4.93)$$

então de (4.88) e (4.89) tem-se que

$$\hat{Y}_{2h} = \sum_{i=1}^2 \left(\tilde{R}_{2i,1,h} \mathcal{X}_{11}(x_1) \mathcal{X}_{21}(x_2) + \tilde{R}_{2i,2,h} \mathcal{X}_{12}(x_1) \mathcal{X}_{22}(x_2) \right) \mathcal{X}_{3i}(x_3) \quad (4.94)$$

De forma similar, escolhendo

$$\tilde{R}_{2i,j,h} = \begin{cases} 1 & \text{para } i = j = h, \\ 0 & \text{caso contrário.} \end{cases} \quad (4.95)$$

de (4.94) tem-se

$$\hat{Y}_{2h} = \mathcal{X}_{1h}(x_1) \mathcal{X}_{2h}(x_2) \mathcal{X}_{3h}(x_3) \quad (4.96)$$

que em (4.87) resulta

$$\hat{y}(\mathbf{x}) = \frac{\phi_1 \prod_{l=1}^3 \mathcal{X}_{l1}(x_l) + \phi_2 \prod_{l=1}^3 \mathcal{X}_{l2}(x_l)}{\prod_{l=1}^3 \mathcal{X}_{l1}(x_l) + \prod_{l=1}^3 \mathcal{X}_{l2}(x_l)} \quad (4.97)$$

Dados dois pontos quaisquer $\mathbf{x}' = [x'_1 \ x'_2 \ x'_3]^T$ e $\mathbf{x}'' = [x''_1 \ x''_2 \ x''_3]^T$ em U , com $\mathbf{x}' \neq \mathbf{x}''$, selecionam-se os centros e aberturas dos conjuntos *fuzzy* respectivamente como $\theta_{l_1} = x'_l$, $\theta_{l_2} = x''_l$ e $\sigma_{l_1} = \sigma_{l_2} = 1$ para $l = 1, 2, 3$, de modo que

$$\mathcal{X}_{l_1}(x_l) = \exp(-(x_l - x'_l)^2) \quad (4.98)$$

$$\mathcal{X}_{l_2}(x_l) = \exp(-(x_l - x''_l)^2) \quad (4.99)$$

A partir de (4.97), (4.98) e (4.99), tem-se

$$\hat{y}(\mathbf{x}') = \frac{\phi_1 + \phi_2 \prod_{l=1}^3 \mathcal{X}_{l_2}(x'_l)}{1 + \prod_{l=1}^3 \mathcal{X}_{l_2}(x'_l)} \quad (4.100)$$

$$\hat{y}(\mathbf{x}'') = \frac{\phi_1 \prod_{l=1}^3 \mathcal{X}_{l_1}(x''_l) + \phi_2}{1 + \prod_{l=1}^3 \mathcal{X}_{l_1}(x''_l)} \quad (4.101)$$

com

$$\mathcal{X}_{l_1}(x''_l) = \mathcal{X}_{l_2}(x'_l) = \exp(-(x'_l - x''_l)^2) \quad (4.102)$$

Definindo

$$\alpha \triangleq \frac{1}{1 + \prod_{l=1}^3 \exp(-(x'_l - x''_l)^2)} \quad (4.103)$$

tem-se

$$\hat{y}(\mathbf{x}') = \alpha \phi_1 + (1 - \alpha) \phi_2 \quad (4.104)$$

$$\hat{y}(\mathbf{x}'') = \alpha \phi_2 + (1 - \alpha) \phi_1 \quad (4.105)$$

Finalmente, selecionando $\phi_1 = 0$ e $\phi_2 = 1$, o que completa o modelo, tem-se

$$\hat{y}(\mathbf{x}') = (1 - \alpha) \quad \text{e} \quad \hat{y}(\mathbf{x}'') = \alpha \quad (4.106)$$

Por definição tem-se $\mathbf{x}' \neq \mathbf{x}''$, o que implica

$$\prod_{l=1}^3 \exp(-(x'_l - x''_l)^2) \neq 1 \quad (4.107)$$

ou seja, $\alpha \neq 0.5$. Conseqüentemente,

$$\hat{y}(\mathbf{x}') = (1 - \alpha) \neq \alpha = \hat{y}(\mathbf{x}'') \quad (4.108)$$

Portanto, \mathcal{M} separa pontos em U , i.e., para cada $\mathbf{x}', \mathbf{x}'' \in U$, $\mathbf{x}' \neq \mathbf{x}''$, existe $\hat{y}(\mathbf{x}) \in \mathcal{M}$ tal que $\hat{y}(\mathbf{x}') \neq \hat{y}(\mathbf{x}'')$. ■

Verifica-se assim que o conjunto \mathcal{M} de modelos hierárquicos com submodelos relacionais convencionais e conjuntos referenciais *fuzzy* Gaussianos satisfaz o teorema de Stone-Weierstrass e que, conseqüentemente, esses modelos possuem a propriedade de aproximação universal. É importante observar que a hipótese de conjuntos referenciais Gaussianos⁶ é, em princípio, apenas uma condição suficiente para a aproximação universal uma vez que o teorema de Stone-Weierstrass também representa apenas uma condição de suficiência e não de necessidade (Zeng e Singh, 1994). Um estudo direcionado aos modelos implementados através de outros tipos de conjuntos referenciais (triangulares, trapezoidais, etc.) é uma perspectiva para trabalhos futuros.

Uma questão relevante é a dependência entre a capacidade de aproximação do modelo e a quantidade “ c ” de conjuntos referenciais *fuzzy* definidos por variável. A seguinte proposição apresenta uma propriedade da relação entre essas grandezas.

Proposição 4.4 *A capacidade de aproximação dos modelos é uma função monotônica crescente do número “ c ” de conjuntos referenciais fuzzy por variável.*

Prova:

Sejam \mathcal{M}_1 e \mathcal{M}_2 os conjuntos de todos os modelos \hat{y} com quantidades c' e c'' de conjuntos *fuzzy* por variável, respectivamente. Sejam ainda $\mathcal{C}_1(\epsilon)$ e $\mathcal{C}_2(\epsilon)$ os conjuntos

⁶Essa hipótese foi também adotada em vários outros trabalhos (e.g. (Wang e Mendel, 1992; Oliveira e Lemos, 1997; Güven e Passino, 2001)).

de todas as funções reais contínuas g para as quais existem modelos \hat{y} respectivamente em \mathcal{M}_1 e \mathcal{M}_2 tais que, para um dado $\epsilon > 0$, a desigualdade em (4.86) é verdadeira.

Supondo que $c' < c''$, tem-se $\mathcal{M}_1 \subset \mathcal{M}_2$, pois os modelos em \mathcal{M}_1 podem ser vistos como casos particulares dos modelos em \mathcal{M}_2 nos quais todos os parâmetros relacionais desses últimos associados aos seus conjuntos *fuzzy* adicionais ($c'' - c'$) são nulos. Portanto, tem-se $\mathcal{C}_1(\epsilon) \subset \mathcal{C}_2(\epsilon)$. Uma outra implicação importante pode ser verificada definindo

$$\delta_i(g) = \inf_{\hat{y} \in \mathcal{M}_i} \sup_{\mathbf{x} \in U} |g(\mathbf{x}) - \hat{y}(\mathbf{x})| \quad (4.109)$$

para $i = 1, 2$. Como $\mathcal{M}_1 \subset \mathcal{M}_2$ e $\mathcal{C}_1(\epsilon) \subset \mathcal{C}_2(\epsilon)$ tem-se:

- $\delta_2(g) \leq \delta_1(g) < \epsilon$ se $g \in \mathcal{C}_1(\epsilon)$.
- $\delta_2(g) < \epsilon \leq \delta_1(g)$ se $g \in \mathcal{C}_2(\epsilon) - \mathcal{C}_1(\epsilon)$.
- $\epsilon \leq \delta_2(g) \leq \delta_1(g)$ se $g \notin \mathcal{C}_2(\epsilon)$.

Das condições acima pode-se concluir que a quantidade de funções reais contínuas que os modelos relacionais hierárquicos podem aproximar com determinada precisão, bem como o erro máximo de aproximação de uma determinada função por esses modelos, são funções não decrescentes (monotonicamente crescentes) da granularidade “ c ” das partições *fuzzy* dos universos de discurso das suas variáveis. ■

Outra questão importante é se a capacidade de aproximação universal também é verificada para os modelos com arquiteturas hierárquicas alternativas discutidos na seção 4.2.6. Claramente, a arquitetura apresentada na figura 4.19, dada pela composição de diferentes modelos hierárquicos através de um operador genérico Ξ , requer uma análise individual para cada operador específico adotado. Por esta razão, limita-se a seguir à análise daquelas arquiteturas onde os submodelos não são necessariamente dotados apenas de duas entradas, como aquela ilustrada na figura 4.20. Com base nos desenvolvimentos anteriores, pode-se verificar que a adoção de mais de duas entradas em um ou mais submodelos não afeta a capacidade de aproximação (universal) do modelo resultante. O próprio modelo da figura 4.20 pode ser utilizado como exemplo. Nesse caso, as saídas *fuzzy* dos submodelos 1 e 2 são dadas respectivamente por:

$$\hat{Y}_{1l} = [X_2 \otimes X_1]^T R_1 = \sum_{p=1}^c \sum_{q=1}^c \tilde{R}_{1,p,q,l} \mathcal{X}_{2p}(x_2) \mathcal{X}_{1q}(x_1) \quad (4.110)$$

$$\hat{Y}_{2h} = [X_4 \otimes X_3 \otimes \hat{Y}_1]^T = \sum_{i=1}^c \sum_{j=1}^c \sum_{l=1}^c \tilde{R}_{2,i,j,l,h} \mathcal{X}_{4i}(x_4) \mathcal{X}_{3j}(x_3) \hat{Y}_{1l} \quad (4.111)$$

onde $l, h = 1, \dots, c$, $\tilde{R}_{1,p,q,j} \triangleq R_{1,\rho,j}$ e $\tilde{R}_{2,i,j,l,h} \triangleq R_{2,\bar{\rho},h}$ com $\rho = (p-1)c + q$ e $\bar{\rho} = (i-1)c^2 + (j-1)c + l$. Substituindo a equação (4.110) em (4.111) e esta última em (4.27), que descreve o operador de defuzzificação, obtém-se a saída *crisp* do modelo como:

$$\hat{y}(\mathbf{x}) = \frac{\sum_{h=1}^c \sum_{i=1}^c \sum_{j=1}^c \sum_{l=1}^c \sum_{p=1}^c \sum_{q=1}^c \phi_h \tilde{R}_{2,i,j,l,h} \tilde{R}_{1,p,q,l} \mathcal{X}_{4_i}(x_4) \mathcal{X}_{3_j}(x_3) \mathcal{X}_{2_p}(x_2) \mathcal{X}_{1_q}(x_1)}{\sum_{h=1}^c \sum_{i=1}^c \sum_{j=1}^c \sum_{l=1}^c \sum_{p=1}^c \sum_{q=1}^c \tilde{R}_{2,i,j,l,h} \tilde{R}_{1,p,q,l} \mathcal{X}_{4_i}(x_4) \mathcal{X}_{3_j}(x_3) \mathcal{X}_{2_p}(x_2) \mathcal{X}_{1_q}(x_1)} \quad (4.112)$$

Nota-se que essa equação é análoga àquelas em (4.90) e (4.91). Logo, ela também satisfaz as condições do teorema de Stone-Weierstrass, resultando que o modelo correspondente é um aproximador universal.

Estrutura de submodelos relacionais simplificados

Pode-se escrever um modelo hierárquico com 3 entradas ($n = 3$) e submodelos relacionais simplificados a partir das equações (4.5) a (4.8) como:

$$\hat{y}_2 = \sum_{i=1}^c \sum_{j=1}^c \tilde{\Omega}_{2i,j} \mathcal{X}_{3_i}(x_3) \mathcal{Y}_{1_j}(\hat{y}_1) \quad (4.113)$$

$$\hat{y}_1 = \sum_{p=1}^c \sum_{q=1}^c \tilde{\Omega}_{1p,q} \mathcal{X}_{2_p}(x_2) \mathcal{X}_{1_q}(x_1) \quad (4.114)$$

sendo $\tilde{\Omega}_{i,p,q} \triangleq \Omega_{i,\rho}$, onde $p, q = 1, \dots, c$ e $\rho = (p-1)c + q$. Seja então \mathcal{M} o conjunto de todos os possíveis modelos com relação aos parâmetros de projeto da arquitetura acima. Dados dois modelos em \mathcal{M} com saídas \hat{y}_2' e \hat{y}_2'' , pode-se verificar que $\hat{y}_2' \cdot \hat{y}_2'' \notin \mathcal{M}$ porque os produtos $\mathcal{Y}'_{1_j}(\hat{y}_1') \cdot \mathcal{Y}''_{1_j}(\hat{y}_1'')$ que ocorrem na equação resultante não podem ser escritos na mesma forma matemática dos seus termos individuais, mesmo que esses termos sejam Gaussianos⁷, pois os seus argumentos \hat{y}_1' e \hat{y}_1'' são distintos. Isso significa que o conjunto \mathcal{M} não é fechado para a multiplicação e que, por conseqüência, não é uma álgebra. Logo, o Teorema de Stone-Weierstrass não se aplica nesse caso. Como esse teorema representa apenas uma condição suficiente para a aproximação universal, não é possível afirmar se os modelos hierárquicos com submodelos relacionais simplificados possuem

⁷Condição imposta para provar a suficiência da hipótese de aproximação universal nos modelos hierárquicos com submodelos relacionais convencionais.

ou não essa propriedade sem uma análise particularizada através de alguma estratégia direcionada para esse tipo específico de modelo. Independente dos resultados dessa análise, que pode ainda abranger estudos a respeito da relação precisão \times quantidade de parâmetros necessários e que constitui uma das perspectivas para desenvolvimentos futuros com relação ao presente trabalho, é importante salientar que esses modelos têm apresentado desempenho adequado em diversas aplicações, como aquelas ilustradas nos exemplos da seção 4.2 e nos artigos relacionados.

4.5 Resumo e Discussões Complementares

Uma alternativa para contornar o problema de dimensionalidade em modelos *fuzzy* e neurais é a estrutura hierárquica constituída de um conjunto de submodelos conectados em cascata. Neste capítulo, dois novos tipos de modelos hierárquicos foram propostos: Modelo hierárquico composto de submodelos relacionais simplificados e modelo hierárquico composto de submodelos relacionais convencionais. Foram desenvolvidos conjuntos fechados de equações para o cálculo do gradiente da saída de cada modelo em função dos seus parâmetros, isto é, conjuntos de equações genéricas para modelos hierárquicos com qualquer quantidade de entradas. Essas equações permitem a estimação ou sintonia-fina dos parâmetros desses modelos a partir de algoritmos de otimização não linear. Esses algoritmos podem ser aplicados conjuntamente com metodologias heurísticas para redução dos efeitos indesejados da não convexidade do problema de otimização, específicas para as características dos modelos hierárquicos propostos.

A introdução do conceito de hierarquia adiciona um novo parâmetro de projeto para os modelos *fuzzy* e neurais. Trata-se da disposição hierárquica das suas variáveis de entrada. Mostrou-se que altera-se de forma correlacionada o desempenho geral de modelagem com alterações específicas na ordem hierárquica. Mais especificamente, mostrou-se que o desempenho de modelagem é superior em relação às variáveis de entrada dispostas em níveis hierárquicos mais elevados, i.e., mais distantes da saída. Esses resultados baseiam-se no fato de que quanto mais alto o seu nível hierárquico mais complexo torna-se o mapeamento de uma variável de entrada para a saída do modelo, envolvendo um número maior de parâmetros livres associados. Com base nesses resultados, arquiteturas hierárquicas alternativas foram propostas para permitir a atribuição de níveis hierárquicos equivalentes a diferentes variáveis do modelo.

O desenvolvimento dos modelos relacionais hierárquicos utilizando submodelos simplificados, embora tenha resultado em uma alternativa estrutural simples e eficaz para a redução do problema de dimensionalidade em sistemas *fuzzy* e neurais, não resolveu a questão da impossibilidade de interpretação lingüística intrínseca das arquiteturas hierárquicas. Os modelos compostos de submodelos relacionais convencionais foram propostos com o objetivo de contornar esse problema. Mostrou-se que existe uma transformação unilateral entre esses modelos e os modelos relacionais *fuzzy* não hierárquicos, permitindo a extração de regras lingüísticas através da mesma metodologia apresentada no capítulo 3. Mostrou-se também, através do teorema de Stone-Weierstrass, que esses

modelos hierárquicos apresentam a propriedade de aproximação universal.

Além de permitir a extração de regras lingüísticas e a aplicação do teorema de Stone-Weierstrass para a verificação da propriedade de aproximação universal, o modelo hierárquico composto de submodelos relacionais convencionais apresenta uma vantagem sobre aquele composto de submodelos relacionais simplificados: A inexistência de conjuntos referenciais *fuzzy* nas variáveis intermediárias entre os diferentes blocos hierárquicos. Essa característica reduz a complexidade e a sensibilidade do modelo à inicialização dos seus parâmetros. No caso simplificado essa sensibilidade é maior devido à possibilidade de saturação dos conjuntos referenciais intermediários. Por outro lado, apesar da ausência dos parâmetros referentes a esses conjuntos, o número de parâmetros de projeto livre no caso de submodelos convencionais é superior uma vez que cada submodelo possui um número “ c ” vezes maior de elementos relacionais do que aqueles simplificados, onde “ c ” é a granularidade das partições *fuzzy* dos universos de discurso das variáveis do modelo.

4.6 Contribuições e Perspectivas

As principais contribuições desse capítulo são:

1. Desenvolvimento de duas novas classes de modelos hierárquicos baseadas na utilização de tipos particulares de modelos relacionais *fuzzy*.
2. Desenvolvimento de uma metodologia para a extração de conhecimento lingüístico a partir de uma dessas classes de modelos.
3. Estudo da capacidade de aproximação dos modelos desenvolvidos.
4. Obtenção de um conjunto de equações genéricas, para modelos com qualquer número de entradas, para a otimização dos parâmetros dos modelos hierárquicos desenvolvidos.
5. Proposta de diferentes procedimentos algorítmicos para a estimação dos parâmetros dos modelos desenvolvidos.
6. Proposta para a seleção da ordem hierárquica dos modelos com base em estudos sobre a influência da disposição das suas variáveis de entrada.
7. Proposta de arquiteturas hierárquicas alternativas.

As principais perspectivas a partir dos resultados obtidos e apresentados no presente capítulo são:

1. Aplicação do Filtro de Kalman Estendido para a estimação *on-line* dos modelos.

2. Desenvolvimento de uma estratégia construtiva com a inserção (e eventualmente a poda) gradativa de conjuntos *fuzzy* e elementos relacionais visando a representação dos erros de modelagem remanescentes das iterações anteriores.
3. Estudos direcionados sobre a capacidade de aproximação dos modelos com submodelos simplificados e/ou implementados com conjuntos referenciais *fuzzy* não Gaussianos.

Capítulo 5

Modelos Baseados em Funções de Base Ortonormal (FBO)

Esse capítulo aborda o problema de identificação de sistemas dinâmicos não lineares utilizando modelos baseados em Funções de Base Ortonormal (FBO). Tem-se como tema central o desenvolvimento de modelos *fuzzy* com estrutura FBO e o seu estudo no que tange a diversos aspectos de interesse, tais como a sua formalização e interpretação matemática, capacidade de representação, entre outros. Análises comparativas entre esses modelos e outras classes de modelos FBO são apresentadas tanto no âmbito teórico como em aplicações práticas. Particular ênfase é dada aos modelos FBO do tipo Wiener-Volterra, discutidos no capítulo 2, para os quais desenvolve-se uma metodologia de projeto da base de funções ortonormais segundo a solução de um problema de otimização que visa a redução da ordem dos modelos resultantes.

5.1 Introdução

Modelos de sistemas dinâmicos são usualmente construídos segundo a bem conhecida topologia (N)ARX — (*Nonlinear*) *Auto-Regressive with eXogenous inputs* — em que a saída do sistema em um dado instante de tempo discreto é representada utilizando amostras passadas dos seus sinais de entrada e saída (E/S). Quando os modelos são não lineares, no entanto, o procedimento de determinação dos termos passados relevantes das variáveis de E/S do sistema recai em um problema complexo de seleção estrutural. Esse não é o único problema da topologia NARX. A dizer, sua parte auto-regressiva causa a recursão de erros que por sua vez podem comprometer a qualidade do modelo, especialmente em previsões de horizontes longos. Essa característica é indesejada em diversas situações práticas, como nas aplicações de controle preditivo que serão discutidas no capítulo 6. Por esta razão, estruturas alternativas para a modelagem de sistemas dinâmicos têm sido investigadas.

Uma estratégia eficaz para contornar os problemas mencionados acima no contexto da modelagem de sistemas dinâmicos estáveis é a utilização de funções de ba-

se ortonormal (Ninness e Gustafsson, 1995), como as funções de Laguerre e Kautz (Broome, 1965; Wahlberg, 1994; Wahlberg e Mäkilä, 1996; Oliveira, 1997) preliminarmente discutidas no capítulo 2. Essas funções têm sido amplamente utilizadas para a modelagem e controle de sistemas lineares e não lineares (e.g. ver (Dumont e Fu, 1993; Oliveira et al., 2000) e referências inclusas), especialmente porque os modelos dinâmicos baseados nessas funções exibem várias características desejáveis, tais como (Dumont e Fu, 1993; Oliveira et al., 1996; Campello e Amaral, 2002c): *i*) Ausência de recursão de saída. Conseqüentemente, não existe a realimentação de erros de previsão, o que normalmente leva a modelos mais precisos. A ausência de recursão também implica o desacoplamento natural de múltiplas saídas no caso multivariável bem como um conjunto de propriedades estatísticas favoráveis à estimação numérica de modelos FBO *lineares nos parâmetros* via algoritmo de mínimos quadrados (ver seção 5.4.1); *ii*) Não é necessário conhecer os termos passados relevantes dos sinais de E/S do sistema; *iii*) É possível aumentar a *capacidade de representação* dos modelos meramente incrementando o número de funções ortonormais na base; *iv*) É garantido que a representação de um sistema estável é também estável; *v*) Tolerância a dinâmicas não modeladas e sensibilidade reduzida aos parâmetros estimados; *vi*) Capacidade de representar dinâmicas com atrasos de tempo.

As seções subseqüentes descrevem em detalhes os modelos FBO e aquelas realizações específicas desses modelos inseridas diretamente no escopo do presente trabalho, i.e., modelos lineares, de Volterra, neurais e *fuzzy* com estrutura FBO. As discussões enfocam particularmente os modelos não lineares, nos quais concentram-se as contribuições originais desse capítulo.

5.2 Modelos FBO

Um sistema dinâmico não linear – por simplicidade e sem perda de generalidade SISO¹ – pode ser genericamente representado como:

$$y(k) = \mathcal{F}(y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)) \quad (5.1)$$

onde n_u e n_y são as quantidades de amostras passadas da entrada u e da saída y , respectivamente, e \mathcal{F} é um operador não linear (NARX). Rescrevendo os termos passados da saída nesta equação utilizando a própria equação de forma recursiva obtém-se a seguinte representação:

$$y(k) = \mathcal{G}[u(d)]_{d=-\infty}^{k-1} \quad (5.2)$$

onde \mathcal{G} é o operador não linear resultante. Na modelagem de sistemas com *fading memory* (Boyd e Chua, 1985), para os quais é garantido que o valor de entrada $u(k-\nu)$,

¹Modelos FBO multivariáveis são discutidos na seção 5.10.

com ν suficientemente grande, não mais contribui significativamente para a saída $y(k)$, essa representação pode ser aproximada (truncada) como:

$$\hat{y}(k) = \mathcal{G}(u(k-1), \dots, u(k-\nu)) \quad (5.3)$$

cujas entradas podem ser vistas como o desenvolvimento de u em uma base com funções de transferência pulsadas escritas no operador deslocamento q^{-1} como $\Phi_{\text{fir},i}(q^{-1}) = q^{-i}$, de modo que a i -ésima entrada é dada por $\Phi_{\text{fir},i}(q^{-1})u(k) = q^{-i}u(k) = u(k-i)$. O diagrama de blocos desse modelo, denominado modelo não linear de resposta ao impulso finita (*Nonlinear Finite Impulse Response - NFIR*), é apresentado na figura 5.1.

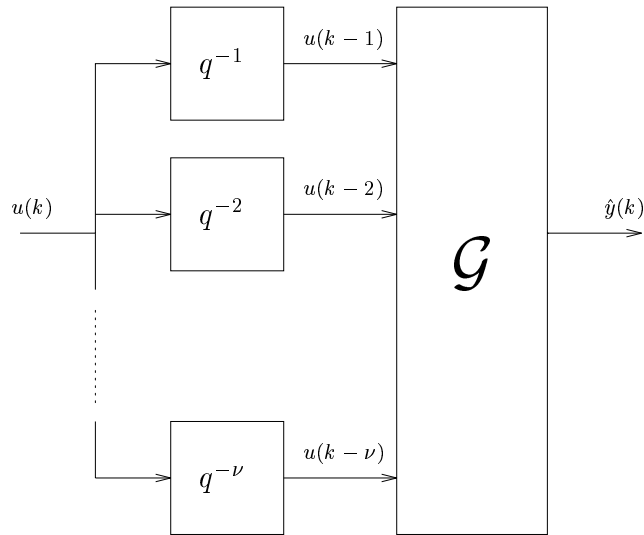


Figura 5.1: Diagrama de blocos do modelo NFIR.

Assim como no caso do modelo linear clássico de resposta ao impulso finita (FIR), pode-se aumentar a capacidade de representação do modelo NFIR em (5.3) simplesmente incrementando a quantidade de funções na base, isto é, a quantidade de termos da entrada. Outra vantagem é a ausência de recursão de saída (parte auto-regressiva), usualmente associada a problemas de convergência nos modelos NARX e nos respectivos algoritmos de identificação. A base $\Phi_{\text{fir},i}(q^{-1})$, porém, geralmente acarreta em modelos com número elevado de termos de entrada, especialmente na representação de sistemas com dinâmicas predominantemente lentas. Uma solução efetiva para contornar esse problema é incorporar na base conhecimento *a priori* a respeito das dinâmicas dominantes do sistema, o que pode ser realizado utilizando as funções de base ortonormal mencionadas na seção 5.1. Nesse caso, o modelo NFIR pode ser reescrito como um modelo FBO, ou seja:

$$\hat{y}(k) = \mathcal{H}(l_1, \dots, l_n) \quad (5.4)$$

onde $l_i(k) = \Phi_{\text{fbo},i}(q^{-1}) u(k)$ é a saída do i -ésimo filtro ortonormal – com função de transferência $\Phi_{\text{fbo},i}(q^{-1})$ – no instante de amostragem k , n é a quantidade de filtros necessários para representar o sistema com uma dada precisão e \mathcal{H} é um operador não linear estático que mapeia $l_i(k)$ ($i = 1, \dots, n$) na saída $\hat{y}(k)$. O diagrama de blocos do modelo FBO é apresentado na figura 5.2.

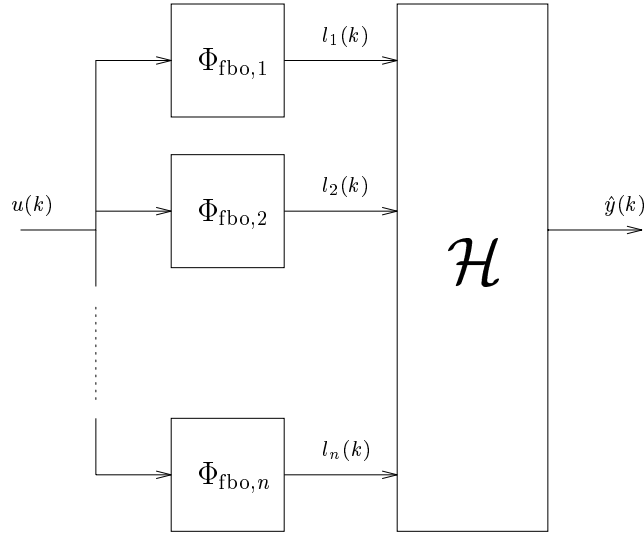


Figura 5.2: Diagrama de blocos do modelo FBO.

O conjunto de funções de transferência dos filtros associados à bem conhecida base de Laguerre (Wahlberg, 1991b), por exemplo, é dado por:

$$\Phi_{\text{fbo},i}(q^{-1}) = \sqrt{1-p^2} \frac{q^{-1}(q^{-1}-p)^{i-1}}{(1-pq^{-1})^i}, \quad i = 1, 2, \dots \quad (5.5)$$

onde $p \in \{\Re : |p| < 1\}$ é o pólo estável que parametriza as funções ortonormais. Note que para $p = 0$ obtém-se a base $\Phi_{\text{fir},i}(q^{-1})$, que é um caso particular da base de Laguerre. A base de Kautz (Wahlberg, 1991a) é uma generalização da base de Laguerre onde as funções são parametrizadas por um par de pólos complexos conjugados. Por esta razão, a base de Kautz requer um número menor de funções para representar sistemas com dinâmicas dominantes mal amortecidas (oscilatórias). As bases de Laguerre e Kautz são realizações particulares das bases ortonormais generalizadas (Ninness e Gustafsson, 1995; Van den Hof et al., 1995), onde as funções não são necessariamente parametrizadas apenas por um único pólo ou par de pólos. Em função da flexibilidade adicional resultante da incorporação de múltiplos modos, as bases generalizadas podem requerer um número menor de funções para a representação de um dado sistema. Por outro lado, requerem uma maior quantidade de informação *a priori* sobre as suas dinâmicas dominantes. Por esta razão, as bases de Laguerre e Kautz ainda são

mais amplamente difundidas e adotadas na literatura; em especial a primeira, que será considerada nos desenvolvimentos subseqüentes do presente capítulo.

As funções ortonormais são recursivas, isto é, a i -ésima função pode ser escrita em função da $(i - 1)$ -ésima. Logo, é possível escrever a dinâmica do conjunto de filtros ortonormais através de uma equação de estados (Oliveira, 1997). Nesse caso, o modelo FBO da figura 5.2 é representado em espaço-de-estados como:

$$\mathbf{l}(k+1) = A\mathbf{l}(k) + \mathbf{b}u(k) \quad (5.6)$$

$$\hat{y}(k) = \mathcal{H}(\mathbf{l}(k)) \quad (5.7)$$

onde $\mathbf{l}(k) = [l_1(k) \cdots l_n(k)]^T$ é o vetor de estados ortonormais² de ordem n . A matriz A e o vetor \mathbf{b} na equação (5.6) dependem apenas da base de funções ortonormais. No caso da base de Laguerre tem-se (Dumont e Fu, 1993; Oliveira, 1997):

$$A = \begin{bmatrix} p & 0 & 0 & \cdots & 0 \\ 1-p^2 & p & 0 & \cdots & 0 \\ (-p)(1-p^2) & 1-p^2 & p & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (-p)^{n-2}(1-p^2) & (-p)^{n-3}(1-p^2) & \cdots & p \end{bmatrix} \quad (5.8)$$

$$\mathbf{b} = \sqrt{1-p^2} [1 \quad -p \quad (-p)^2 \quad \cdots \quad (-p)^{n-1}]^T \quad (5.9)$$

onde p é o pólo de Laguerre. O modelo FBO com dinâmica de Laguerre é ilustrado na figura 5.3, onde observa-se a recursão dos estados através da disposição em cascata dos filtros, representados no plano freqüencial discreto (variável z). Nota-se claramente que esse modelo é, assim como os demais modelos FBO representados genericamente na figura 5.2, descrito por uma dinâmica linear que relaciona a entrada $u(k)$ aos estados de Laguerre $l_i(k)$ seguida de um mapeamento estático entre esses estados e a saída $\hat{y}(k)$, ou seja, um modelo do tipo *Wiener*.

Embora nenhum atraso de tempo esteja explicitamente representado nos modelos FBO apresentados acima, é importante salientar que as funções ortonormais são capazes de representar sistemas com atraso (Mäkilä, 1990; Fu e Dumont, 1993). Não obstante, pode-se incorporar explicitamente ao modelo qualquer informação aproximada disponível sobre o atraso real do sistema, o que permite uma redução no número de funções e respectivos filtros necessários para modelá-lo com uma dada precisão. Para tanto, basta substituir $u(k)$ por $u(k - \tau)$ na equação (5.6), onde τ é o atraso estimado.

²Essa denominação será utilizada na seqüência do texto embora a ortonormalidade seja uma propriedade das funções e não propriamente dos estados.

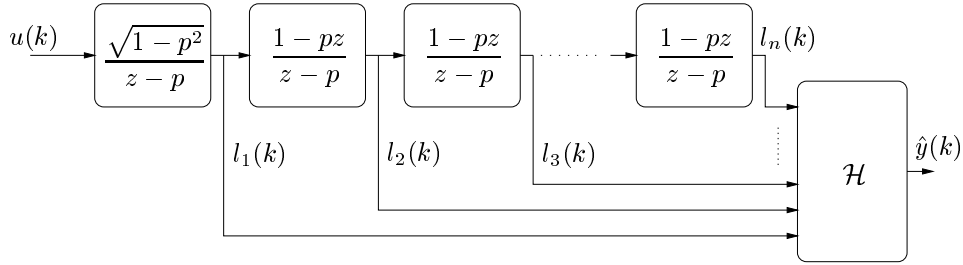


Figura 5.3: Modelo FBO com dinâmica de Laguerre.

Finalmente, é importante observar que a restrição de modelar apenas sistemas estáveis exclui sistemas com ação integral, que são comuns na prática. Nesses casos, no entanto, essa restrição pode ser facilmente contornada através da definição de uma variável de saída auxiliar $\Delta y(k) = y(k) - y(k-1)$, desde que a presença de um integrador no sistema seja conhecida *a priori*.

5.3 Classes de Modelos FBO

A forma do operador genérico \mathcal{H} em (5.7) determina o tipo específico de modelo FBO. No que segue, apresentam-se quatro diferentes tipos de modelos: Modelo linear, modelo de Volterra, modelo relacional *fuzzy* simplificado e modelo *fuzzy* Takagi-Sugeno (TS), todos no contexto da estrutura FBO. Os dois primeiros serão considerados para fins comparativos com relação aos dois últimos, que fazem parte do conjunto de contribuições do presente trabalho. Todos esses modelos podem ser descritos utilizando uma formulação unificada onde o operador \mathcal{H} em (5.7) é representado de forma *linear nos parâmetros*, como:

$$\mathcal{H}(\mathbf{l}(k)) = \boldsymbol{\lambda}(k)^T \boldsymbol{\zeta} \quad (5.10)$$

sendo $\boldsymbol{\zeta} = [\zeta_1 \cdots \zeta_\mu]^T$ o vetor de parâmetros a serem estimados e $\boldsymbol{\lambda}(k)$ um vetor que depende apenas dos estados ortonormais $\mathbf{l}(k)$, como mostrado a seguir.

5.3.1 Modelo Linear

Um modelo FBO linear é dado por uma combinação linear dos estados ortonormais e, opcionalmente, um termo constante adicional (*bias*) utilizado para que o modelo seja capaz de representar adequadamente qualquer nível constante na saída do sistema. Quando esse *bias* é utilizado, no entanto, uma terminologia mais adequada para descrever esse modelo é *modelo afim*. Nesse caso, o vetor $\boldsymbol{\lambda}$ em (5.10) é o próprio vetor de estados ortonormais com um elemento unitário adicional, i.e.

$$\boldsymbol{\lambda}(k) = \begin{bmatrix} 1 \\ \mathbf{1}(k) \end{bmatrix} = [1 \ l_1(k) \ \cdots \ l_n(k)]^T \quad (5.11)$$

Tem-se então que o número de parâmetros de projeto livre em (5.10) é igual à ordem do modelo mais um, isto é, $\mu = n + 1$.

5.3.2 Modelo de Volterra

Modelos de Volterra de 2a. ordem são, por razões práticas, aqueles usualmente adotados em problemas acadêmicos e reais (Billings, 1980; Dumont e Fu, 1993). O desenvolvimento dos kernels de Volterra de 1a. e 2a. ordem em funções ortonormais (Dumont e Fu, 1993; Campello, Amaral e Favier, 2001; Campello e Amaral, 2002a), detalhado no capítulo 2, resulta em um modelo do tipo Wiener-Volterra que pode ser descrito pelas equações (5.6) e (5.7). Especificamente, tem-se que a saída do modelo é dada por³:

$$\hat{y}(k) = \alpha_0 + \sum_{i=1}^{N_1} \alpha_i l_i(k) + \sum_{i=1}^{N_2} \sum_{j=1}^i \alpha_{i,j} l_i(k) l_j(k) \quad (5.12)$$

onde α_i e $\alpha_{i,j}$ são os coeficientes dos desenvolvimentos de 1a. e 2a. ordem, com N_1 e N_2 funções ortonormais respectivamente, e α_0 é um coeficiente de ordem zero adicional (*bias*). Considerando os coeficientes $\alpha_{(\cdot)}$ como parâmetros a serem estimados, pode-se verificar utilizando (5.7), (5.10) e (5.12) que:

$$\boldsymbol{\lambda}(k) = [1 \ l_1(k) \ \cdots \ l_{N_1}(k) \ l_1(k)^2 \ l_2(k)l_1(k) \ l_2(k)^2 \ \cdots \ l_{N_2}(k)l_1(k) \ l_{N_2}(k)l_2(k) \ \cdots \ l_{N_2}(k)^2]^T \quad (5.13)$$

sendo a ordem da dinâmica de estados em (5.6) dada por $n = \max\{N_1, N_2\}$. Nesse caso, que claramente representa uma generalização não linear do modelo apresentado na seção 5.3.1, o número de parâmetros de projeto livre em (5.10) é dado por $\mu = (N_2^2 + N_2 + 2N_1 + 2)/2$.

5.3.3 Modelo Relacional *Fuzzy* / Rede RBF

Uma nova classe de modelo FBO pode ser obtida implementando o operador \mathcal{H} em (5.7) e (5.10) utilizando um modelo *fuzzy* (Oliveira et al., 1999). Adota-se inicialmente o modelo *fuzzy* com estrutura relacional simplificada descrito no capítulo 4 (seção 4.2.1). Foi visto naquele capítulo que esse modelo é completamente equivalente a uma rede neural RBF com funções de ativação Gaussianas sempre que implementado utilizando

³Partindo da hipótese de simetria do termo de 2a. ordem (ver seção 2.4).

conjuntos *fuzzy* também Gaussianos (Campello e Amaral, 1999a). Nesse caso, conforme detalhado no apêndice A, o modelo consiste de uma combinação linear de funções Gaussianas multidimensionais das suas variáveis de entrada (os estados ortonormais, no presente contexto), de modo que o operador \mathcal{H} em (5.7) pode ser escrito como na equação (5.10) com (Campello, Meleiro, Amaral e Maciel Filho, 2001):

$$\boldsymbol{\lambda}(k) = [w_1(\mathbf{1}(k)) \cdots w_\mu(\mathbf{1}(k))]^T \quad (5.14)$$

onde w_i é a i -ésima função de ativação, dada por:

$$w_i(\mathbf{1}(k)) = \exp\left(-(\mathbf{1}(k) - \boldsymbol{\theta}_i)^T \Lambda_i^{-1} (\mathbf{1}(k) - \boldsymbol{\theta}_i)\right) \quad (5.15)$$

O vetor $\boldsymbol{\theta}_i$ e a matriz Λ_i representam respectivamente o centro e as aberturas n -dimensionais da função. Esses termos estão diretamente associados aos centros e aberturas dos conjuntos referenciais *fuzzy* do modelo relacional equivalente (ver figuras 4.1 e 4.2). Conforme discutido na seção 4.2.1, o número de funções de ativação, que é também a quantidade de elementos do vetor de parâmetros $\boldsymbol{\zeta}$ em (5.10), é dado por $\mu = c^n$, onde c é a quantidade de conjuntos referenciais *fuzzy* por variável de entrada.

5.3.4 Modelo *Fuzzy* Takagi-Sugeno

Outra realização de um modelo *fuzzy* FBO é obtida implementando o operador \mathcal{H} em (5.7) e (5.10) utilizando um modelo do tipo Takagi-Sugeno (TS). Os modelos TS, apresentados no capítulo 2, constituem uma generalização dos modelos relacionais discutidos na seção 5.3.3. Conseqüentemente, são também uma generalização das redes RBF equivalentes quando implementados utilizando conjuntos *fuzzy* Gaussianos para as variáveis da premissa (ver apêndice A).

Um modelo TS consiste basicamente de um mapeamento não linear dado por uma interpolação de M modelos locais usualmente lineares ou afins. Logo, a saída do modelo FBO resultante da implementação do operador \mathcal{H} utilizando um modelo desse tipo pode ser descrita como (Campello, Meleiro, Amaral e Maciel Filho, 2001):

$$\hat{y}(k) = \frac{\sum_{i=1}^M w_i(\mathbf{1}(k)) \hat{y}_i(k)}{\sum_{i=1}^M w_i(\mathbf{1}(k))} \quad (5.16)$$

onde

$$\hat{y}_i(k) = q_0^i + q_1^i l_1(k) + \cdots + q_n^i l_n(k) \quad (5.17)$$

é o i -ésimo modelo local (nesse caso afim) com parâmetros de projeto livre q_j^i ($j = 0, \dots, n$) e $w_i(\mathbf{l}(k))$ é o respectivo nível de ativação. Quando o modelo TS é implementado utilizando conjuntos *fuzzy* Gaussianos, o nível de ativação de cada modelo local é descrito pela mesma função dada em (5.15).

Das equações (5.7), (5.10), (5.16) e (5.17) pode-se inferir:

$$\begin{aligned} \boldsymbol{\lambda}(k) = \gamma(k) [& w_1(\mathbf{l}(k)) \quad w_1(\mathbf{l}(k))l_1(k) \quad \cdots \quad w_1(\mathbf{l}(k))l_n(k) \quad \cdots \\ & \cdots \quad w_M(\mathbf{l}(k)) \quad w_M(\mathbf{l}(k))l_1(k) \quad \cdots \quad w_M(\mathbf{l}(k))l_n(k)]^T \end{aligned} \quad (5.18)$$

onde $\gamma(k) = 1 / \sum_{i=1}^M w_i(\mathbf{l}(k))$ é o termo de normalização da equação (5.16). Nesse caso, a quantidade de elementos do vetor de parâmetros $\boldsymbol{\zeta}$ em (5.10) é $\mu = M(n + 1)$. Considerando a mesma parametrização da seção 5.3.3 em relação à quantidade de regras *fuzzy* do modelo, o que significa fazer $M = c^n$, tem-se $\mu = c^n(n + 1)$. Essa condição equivale à hipótese de um modelo com *base de regras completa* (e.g. ver (Güven e Passino, 2001)). Nesse tipo de modelo, uma quantidade fixa de conjuntos referenciais *fuzzy* é atribuída a cada variável da premissa. A base de regras do modelo é denominada *completa* porque é constituída de todas as possíveis regras referentes a combinações desses conjuntos referenciais. Essa hipótese será considerada nos desenvolvimentos subsequentes desse trabalho no que se refere aos modelos FBO do tipo Takagi-Sugeno (FBO-TS).

5.4 Estimação de Parâmetros

Dado que todos os modelos FBO descritos na seção 5.3 são lineares com relação ao vetor de parâmetros $\boldsymbol{\zeta}$ em (5.10), esse vetor pode ser estimado, por exemplo, utilizando o algoritmo dos Mínimos Quadrados (Söderström e Stoica, 1989; Ljung, 1999). Logo, para um conjunto de N pares de dados de E/S medidos a partir de um sistema a ser modelado, i.e., $\{u(k), y(k)\}_{k=1}^N$, a solução ótima global estrita $\boldsymbol{\zeta}^*$ que minimiza o Erro Quadrático Médio (EQM) entre as saídas do sistema e do modelo é:

$$\boldsymbol{\zeta}^* = (L^T L)^{-1} L^T \mathbf{y} \quad (5.19)$$

onde

$$L = [\boldsymbol{\lambda}(1) \quad \cdots \quad \boldsymbol{\lambda}(N)]^T \quad (5.20)$$

e

$$\mathbf{y} = [y(1) \cdots y(N)]^T \quad (5.21)$$

Versões recursivas dessa solução podem também ser utilizadas, tais como o algoritmo RLS (Eykhoff, 1974; Ljung, 1999) combinado com alternativas numericamente robustas, como a decomposição U-D (Wellstead e Zarrop, 1991; Ljung, 1999).

É importante observar que no caso dos modelos FBO *fuzzy* (seções 5.3.3 e 5.3.4) supõe-se que os conjuntos referenciais desses modelos são especificados ou determinados *a priori*, o que pode ser realizado através de alguma das metodologias discutidas nos capítulos anteriores (e.g. *clustering* (Babuška e Verbruggen, 1996)).

5.4.1 Análise de Sistemas com Ruídos de Medida

Assuma que o sistema a ser modelado possa ser representado como nas equações (5.6), (5.7) e (5.10), de tal forma que a sua saída possa ser descrita por $y_0(k) = \boldsymbol{\lambda}(k)^T \boldsymbol{\zeta}_0$, onde $\boldsymbol{\zeta}_0$ é o vetor de parâmetros verdadeiros do sistema. Assuma também que o vetor de dados de saída disponíveis \mathbf{y} em (5.21) esteja contaminado com ruído branco de média zero⁴, de modo que a saída medida no instante de amostragem k seja dada por $y(k) = \boldsymbol{\lambda}(k)^T \boldsymbol{\zeta}_0 + v(k)$, onde $v(k)$ é uma variável aleatória independente caracterizada como:

$$\mathcal{E}\{v(k)\} = 0 ; \quad \mathcal{E}\{v(k)v(j)\} = \begin{cases} \sigma^2 & \text{if } k = j \\ 0 & \text{if } k \neq j \end{cases} \quad (5.22)$$

sendo \mathcal{E} o valor esperado (esperança matemática) e σ^2 a variância do ruído. Como os modelos FBO não possuem recursão da variável de saída, pode-se demonstrar que a estimativa ótima dos mínimos quadrados dada pela equação (5.19) exhibe as seguintes propriedades (Franklin et al., 1990; Ljung, 1999):

1. Não Polarizada (*Unbiased*):

$$\mathcal{E}\{\boldsymbol{\zeta}^*\} = \boldsymbol{\zeta}_0 \quad \text{ou} \quad \mathcal{E}\{\boldsymbol{\zeta}_0 - \boldsymbol{\zeta}^*\} = 0 \quad (5.23)$$

o que significa que o valor esperado para o erro de estimação (média do erro) é nulo.

2. Melhor Estimativa Linear Não Polarizada (*Best Linear Unbiased Estimation - BLUE*):

⁴Qualquer média diferente de zero pode ser estimada e depois subtraída.

$$\zeta^* = \arg \min_{\zeta} \mathcal{E}\{(\zeta_0 - \zeta)^T (\zeta_0 - \zeta)\} \quad (5.24)$$

o que significa que essa estimativa minimiza a variância do erro de estimação.

3. Consistente:

$$\lim_{N \rightarrow \infty} \mathcal{E}\{(\zeta_0 - \zeta^*)^T (\zeta_0 - \zeta^*)\} = 0 \quad (5.25)$$

o que significa que a variância do erro de estimação tende a zero conforme a quantidade de dados N tende ao infinito.

Observação 5.1 *A presença de ruído apenas nos dados de saída não é uma hipótese restritiva no contexto de sistemas controlados por computador, que constituem o principal objetivo de aplicação dos modelos FBO. A razão é que nesses sistemas a entrada (ação de controle) é conhecida, i.e., determinística, enquanto sua saída é medida, e portanto sujeita a ruídos. Contudo, é importante destacar que as propriedades estatísticas relacionadas acima pressupõem que o procedimento de estimação seja realizado em malha aberta, pois, em malha fechada, a entrada do sistema depende da saída mensurada e, conseqüentemente, dos ruídos de medida.*

5.5 Projeto da Base de Funções Ortonormais

Conforme discutido no capítulo 2, as bases de funções ortonormais são completas no espaço $L^2[0, \infty)$, o que resulta em duas condições fundamentais. A primeira é que o número de funções n representa apenas um compromisso entre a capacidade de aproximação e a quantidade de parâmetros do modelo, grandezas monotonicamente crescentes em n . Deve-se enfatizar, porém, que o número de funções necessárias para proporcionar uma dada representação depende da complexidade do problema. Sistemas dinâmicos com múltiplos modos dominantes, por exemplo, requerem uma quantidade maior de funções. A demanda de funções também depende da base utilizada. Na representação de sistemas com dinâmicas dominantes pouco amortecidas, por exemplo, a base de Laguerre requer uma quantidade maior de funções do que a base de Kautz.

A segunda implicação da completude das bases ortonormais é que a parametrização do conjunto de funções não é crítica. Para a base de Laguerre, essa condição significa que para qualquer pólo de Laguerre p estável existe uma quantidade correspondente n de funções que proporciona uma determinada capacidade de aproximação. Dada uma quantidade fixa de funções, o problema que se coloca é como selecionar o pólo de Laguerre para maximizar a precisão do modelo. A solução para esse problema está

bem estabelecida na literatura no contexto de modelos lineares. Em (Masnadi-Shirazi e Ahmed, 1991), por exemplo, propõe-se determinar o pólo minimizando o erro de aproximação do desenvolvimento da resposta ao impulso de uma classe de sistemas lineares em uma quantidade fixa de funções ortonormais. Já em (Fu e Dumont, 1993), o pólo é obtido maximizando a taxa de convergência do desenvolvimento completo (não truncado) da resposta ao impulso do sistema, o que indiretamente minimiza o erro de aproximação associado a um truncamento *a posteriori* desse desenvolvimento em um número finito de funções. Essa solução foi estendida para o caso de modelos de Volterra de segunda ordem em (Campello, Amaral e Favier, 2001; Campello e Amaral, 2002a) e é de particular interesse no presente trabalho, uma vez que a sua generalização para modelos de Volterra de qualquer ordem será discutida na seção 5.12.

Uma alternativa para a determinação automática do pólo de Laguerre em modelos FBO monovariáveis em geral, ou modelos multivariáveis⁵ onde os múltiplos pólos são restritos a um mesmo valor, é a representação do problema de estimação do pólo como um problema de otimização escalar (Oliveira e Amaral, 2000). Nesse caso, o procedimento de otimização pode ser executado utilizando métodos de busca unidimensional (*line search*) (Bazaraa et al., 1993), dentro do intervalo de factibilidade $] -1, 1[$. Pode-se adotar, por exemplo, o método da seção áurea, que não requer qualquer informação sobre a derivada do critério a ser otimizado. O principal inconveniente dessa estratégia é que para cada avaliação do valor do critério, associado a um determinado valor atribuído ao pólo, deve-se estimar todos os demais parâmetros do modelo, i.e., o mapeamento estático \mathcal{H} em (5.7). Esse procedimento pode representar um custo computacional elevado se esse mapeamento for não linear nos parâmetros, como é o caso dos modelos FBO hierárquicos que serão discutidos na seção 5.11. O mesmo problema está associado, por exemplo, à utilização de GAs (Mitchell, 1998) para a otimização do pólo; ou pólos, no caso multivariável.

Independente do tipo de modelo FBO em questão, isto é, da forma do mapeamento \mathcal{H} em (5.7), a forma mais simples porém eficaz de selecionar a parametrização das funções ortonormais é utilizando conhecimento *a priori* a respeito das dinâmicas dominantes do sistema (Ninness e Gustafsson, 1995), o que pode ser obtido através da sua resposta no tempo ou em frequência (Zervos e Dumont, 1988; Wahlberg e Ljung, 1992). Para a base de Laguerre isso equivale basicamente a selecionar o pólo real estável que melhor represente a constante de tempo dominante do sistema (Sentoni et al., 1996; Sentoni et al., 1998). Esse procedimento será utilizado nas aplicações práticas dos modelos FBO descritas na seqüência do trabalho.

5.6 Interpretação dos Modelos FBO

Discutiu-se no capítulo 2 que o modelo FBO linear apresentado na seção 5.3.1 pode ser matematicamente interpretado como o resultado do desenvolvimento da respos-

⁵Ver seção 5.10.

ta ao impulso do modelo linear de convolução em uma base de funções ortonormais. Analogamente, o modelo Wiener-Volterra representa o desenvolvimento dos kernels do modelo de Volterra também em uma base de funções ortonormais. Inspirados na representação de estados dos modelos citados acima, descrita genericamente pelas equações (5.6) e (5.7), foram propostos na literatura modelos FBO baseados na implementação do mapeamento estático \mathcal{H} em (5.7) através de redes neurais do tipo MLP (Back e Tsoi, 1996; Sentoni et al., 1996; Sentoni et al., 1998; Balestrino et al., 1999; Alataris et al., 2000; Vázquez e Agamennoni, 2001; Arto et al., 2001). Embora essa estratégia seja consistente sob o ponto de vista quantitativo, os modelos resultantes não são matematicamente interpretáveis. Diferentemente, os modelos FBO *fuzzy* apresentados nas seções 5.3.3 e 5.3.4 constituem realizações particulares de uma representação FBO generalizada que dispõe de um significado matemático claro, apresentado a seguir.

5.6.1 Modelos FBO-TS Generalizados

A interpolação de um conjunto de modelos locais, que é a idéia central na construção dos modelos *fuzzy* TS, pode ser estendida para a estrutura FBO de duas formas distintas. A primeira consiste simplesmente da implementação do mapeamento \mathcal{H} em (5.7) utilizando um modelo TS, conforme discutido na seção 5.3.4. Nessa abordagem, os modelos locais diferem uns dos outros apenas no que diz respeito às relações estáticas entre o conjunto de estados e a saída do modelo. Os estados em si e a respectiva representação dinâmica FBO dada pela equação (5.7) são considerados comuns a todos os modelos locais. Uma abordagem mais geral e que permite uma interpretação dos modelos *fuzzy* FBO discutidos anteriormente pode ser obtida relaxando esta última hipótese (Campello e Amaral, 2002c). Essa abordagem baseia-se na versão de espaço-de-estados dos modelos TS onde cada regra representa um modelo de estados local, i.e. (Teixeira e Žak, 1999; Teixeira et al., 2000):

$$\begin{aligned} R^i : \text{SE } l_1(k) \text{ é } \mathcal{L}_1^i \text{ E } \dots \text{ E } l_n(k) \text{ é } \mathcal{L}_n^i \\ \text{ENTÃO } \begin{cases} \mathbf{l}_i(k+1) = A_i \mathbf{l}_i(k) + \mathbf{b}_i u(k) \\ \hat{y}_i(k) = f_i(\mathbf{l}_i(k)) \end{cases} \end{aligned} \quad (5.26)$$

onde R^i denota a i -ésima regra. Considerando a utilização da base de Laguerre, por exemplo, tem-se que A_i e \mathbf{b}_i representam a realização em espaço-de-estados da i -ésima dinâmica de Laguerre com pólo p_i e vetor de estados $\mathbf{l}_i(k)$, $f_i(\mathbf{l}_i(k))$ é o mapeamento estático que relaciona a saída $\hat{y}_i(k)$ do i -ésimo modelo local aos seus respectivos estados e \mathcal{L}_j^i é o conjunto *fuzzy* da i -ésima regra associado à j -ésima variável da premissa. As variáveis da premissa são os estados de Laguerre do modelo resultante. Esses estados são calculados utilizando o mecanismo padrão de inferência *fuzzy* do tipo TS, i.e., como uma média ponderada dos estados locais, como segue:

$$\mathbf{1}(k+1) = \frac{\sum_{i=1}^M w_i(\mathbf{1}(k)) \mathbf{1}_i(k+1)}{\sum_{i=1}^M w_i(\mathbf{1}(k))} \quad (5.27)$$

onde $\mathbf{1}(k) = [l_1(k) \cdots l_n(k)]^T$, M é o número total de regras (modelos locais) e $w_i(\mathbf{1}(k))$ é o nível de ativação da i -ésima regra, dado por:

$$w_i(\mathbf{1}(k)) = \prod_{j=1}^n \mathcal{L}_j^i(l_j(k)) \quad (5.28)$$

que pode ser escrito exatamente como na equação (5.15) se os conjuntos *fuzzy* \mathcal{L}_j^i forem Gaussianos.

Analogamente aos estados, a saída global do modelo é calculada como:

$$\hat{y}(k) = \frac{\sum_{i=1}^M w_i(\mathbf{1}(k)) \hat{y}_i(k)}{\sum_{i=1}^M w_i(\mathbf{1}(k))} \quad (5.29)$$

onde $\hat{y}_i(k) = f_i(\mathbf{1}_i(k))$. Se os mapeamentos estáticos locais f_i forem, por exemplo, funções afins dos vetores de estados correspondentes, então o modelo dado pelas equações (5.27), (5.28) e (5.29) é linear nos parâmetros dessas funções e pode ser claramente interpretado como uma interpolação de um conjunto de modelos FBO afins, discutidos na seção 5.3.1.

Um modelo de particular interesse é obtido considerando que os pólos de Laguerre dos M modelos locais são iguais ($p_1 = p_2 = \cdots = p_M = p$), o que resulta em uma dinâmica FBO comum a todos eles, isto é, $\mathbf{1}_1(k) = \cdots = \mathbf{1}_M(k)$. Da equação (5.27) tem-se que essa condição implica $\mathbf{1}(k) = \mathbf{1}_i(k)$ para $i = 1, \dots, M$. Nesse caso, o modelo FBO-TS Generalizado recai na representação das equações (5.6) e (5.7), com \mathcal{H} dado por (5.29). É evidente que esse modelo equivale ao modelo FBO-TS apresentado na seção 5.3.4. Conseqüentemente, trata-se também de uma generalização do modelo FBO relacional/RBF discutido na seção 5.3.3. Logo, esses modelos podem ser interpretados como interpolações TS de diferentes modelos locais do tipo FBO com uma dinâmica de estados comum.

A condição de uma dinâmica FBO única compartilhada por todos os modelos locais não é restritiva uma vez que a completude das bases ortonormais faz com que a sua

parametrização não seja crucial (ver seção 5.5). Logo, a seleção dos pólos de Laguerre dos modelos locais não é crítica. Por outro lado, esta condição simplifica a estrutura do modelo e, em síntese, equivale a supor que os modos *dominantes* do sistema não são *significativamente* distintos para suas diferentes regiões operacionais. Mesmo que essa hipótese não seja verdadeira, a única conseqüência é que uma quantidade maior de funções na base (estados) poderão ser necessárias para proporcionar ao modelo uma dada capacidade de representação. Conforme discutido na seção 5.5, no entanto, esse problema pode ser reduzido através de uma escolha adequada da parametrização da base, isto é, do pólo comum de Laguerre p . Na próxima seção mostra-se que, mesmo com a simplificação discutida acima, os modelos FBO *fuzzy* podem aproximar com precisão arbitrária uma ampla classe de sistemas dinâmicos não lineares.

5.7 Capacidade de Representação dos Modelos FBO

O desenvolvimento em funções ortonormais da resposta ao impulso de um sistema linear (BIBO) estável representado por um modelo de convolução, que resulta no modelo FBO descrito na seção 5.3.1, não altera a capacidade de representação do modelo original. Logo, tem-se que um modelo FBO linear é capaz de representar com precisão desejada qualquer sistema dinâmico linear estável. Em relação ao desenvolvimento de kernels de Volterra, tem-se que os modelos FBO resultantes (Wiener-Volterra) possuem a mesma capacidade de representação dos modelos de Volterra originais, formalizada no trabalho de Boyd e Chua (1985) (ver capítulo 2, seção 2.4). Em síntese, tem-se que esses modelos são capazes de aproximar arbitrariamente bem qualquer sistema dinâmico não linear de tempo discreto que seja causal, contínuo, e que possua *fading memory*⁶, desde que a entrada seja limitada inferior e superiormente.

A capacidade de representação dos modelos FBO *fuzzy* descritos nas seções 5.3.3 e 5.3.4 pode ser analisada diretamente em função dos resultados acima, como segue: Nos modelos Wiener-Volterra o mapeamento \mathcal{H} da representação de estados genérica em (5.6) e (5.7) é dado por um desenvolvimento polinomial como aquele da equação (5.12), não necessariamente de 2a. ordem. Como a entrada é limitada dentro de um intervalo fechado (por hipótese) e a dinâmica de estados FBO em (5.6) é sempre estável, tem-se que os respectivos estados, também limitados, determinam um domínio compacto (hipercubo) para o mapeamento polinomial \mathcal{H} do modelo, que é contínuo. Por outro lado, conforme discutido no capítulo 2, sabe-se que os modelos *fuzzy* são aproximadores universais, isto é, são capazes de aproximar com precisão arbitrária qualquer mapeamento contínuo definido sobre um domínio compacto. Logo, tem-se que um modelo FBO *fuzzy* pode aproximar com precisão arbitrária um modelo Wiener-Volterra e, conseqüentemente, qualquer sistema dinâmico não linear de tempo discreto que seja causal,

⁶Propriedade que remete a sistemas dinâmicos estáveis e com estado estacionário único.

contínuo, com *fading memory* e que possua entrada limitada em um intervalo fechado. Essas características definem uma ampla classe de sistemas encontrados na prática.

É importante mencionar que a hipótese de entrada limitada é essencial para a utilização de qualquer modelo *fuzzy* ou neural, independente da sua configuração dinâmica, uma vez que essa hipótese é necessária para garantir domínios de entrada compactos para esses modelos. Essa hipótese também não é restritiva no contexto de engenharia.

5.8 Modelos FBO: *Fuzzy* \times Volterra

As vantagens dos modelos *fuzzy* FBO em comparação às demais arquiteturas de modelos *fuzzy* de sistemas dinâmicos recaem nas próprias características estruturais da abordagem FBO, destacadas nas seções 5.1 e 5.2. Contudo, não é evidente se existem vantagens nesses modelos quando comparados aos modelos Wiener-Volterra, que também apresentam as mesmas características estruturais. A principal limitação dos modelos Wiener-Volterra é decorrente do truncamento explícito dos mesmos em uma quantidade finita de termos (ordem). Conforme discutido no capítulo 2, esse truncamento é adotado na prática (usualmente além do termo de segunda ordem) para limitar a quantidade de parâmetros a serem estimados, mas também limita a capacidade de representação dos modelos. A questão é até que ponto é possível superar essas limitações substituindo o mapeamento estático polinomial truncado correspondente por um mapeamento *fuzzy* com capacidade de aproximação universal. Embora a aproximação universal suponha a quantidade de conjuntos *fuzzy* do respectivo mapeamento como sendo um parâmetro de projeto livre, a implementação prática desse mapeamento é realizada através de uma quantidade fixa de conjuntos, o que termina por realizar um certo tipo de truncamento (não explícito) da representação. Nesse caso, no entanto, a forma dos conjuntos *fuzzy* pode ser otimizada, enquanto a representação polinomial dos modelos Wiener-Volterra é estruturalmente imutável.

Comparações entre modelos de Volterra e aproximadores universais, viz., redes neurais artificiais, foram descritas na literatura no contexto de identificação de sistemas dinâmicos não lineares (Back e Tsoi, 1996; Marmarelis e Zhao, 1997; Chon et al., 1998). Os critérios utilizados nesses trabalhos, contudo, são questionáveis. Em (Chon et al., 1998), por exemplo, mostrou-se em um caso de estudo que um modelo de Volterra e duas redes MLP com funções de ativação distintas (sigmóides e polinomiais) foram capazes de reproduzir resultados com precisões similares, sendo que o modelo de Volterra possuía aproximadamente metade dos parâmetros das redes. No entanto, o modelo de Volterra utilizado nessa análise era desenvolvido em funções de Laguerre (Wiener-Volterra), enquanto as redes MLP possuíam estrutura NFIR. Conforme discutido na seção 5.2, a estrutura NFIR requer em geral um número significativamente maior de parâmetros. Em (Marmarelis e Zhao, 1997) sugeriu-se, através de exemplos, que redes MLP com funções de ativação sigmóides ou polinomiais seriam mais eficientes do que modelos de Volterra para a modelagem de sistemas fortemente não lineares, porém menos eficazes quando os sistemas apresentam apenas não linearidades brandas.

Essa afirmação deve-se ao fato de que os modelos de Volterra necessitam de uma quantidade elevada de parâmetros necessários para a aproximação de não linearidades fortes enquanto, em contrapartida, a despeito das não linearidades, o esforço computacional de treinamento das redes via *back-propagation* é elevado. Novamente o modelo de Volterra utilizado nessa análise era desenvolvido em funções de Laguerre (Wiener-Volterra), enquanto as redes MLP possuíam estrutura NFIR. Com estruturas e algoritmos de estimação (*back-propagation* e mínimos quadrados) distintos torna-se difícil uma comparação da eficiência relativa de cada estratégia. Em outro trabalho, Back e Tsoi (1996) mostraram um caso de estudo em que uma rede MLP com estrutura FBO foi capaz de produzir aproximações com maior precisão do que um modelo Wiener-Volterra, ambos utilizando dinâmica de Laguerre. Nessa análise, porém, não se consideraram fatores importantes como a quantidade de parâmetros em cada um dos modelos e o esforço computacional requerido pelos diferentes algoritmos (*back-propagation* e mínimos quadrados) na fase de estimação.

Para contornar os vícios presentes nas análises comparativas dos trabalhos descritos acima, mostra-se a seguir através de um caso de estudo que de fato é possível, dependendo da aplicação, superar o desempenho de modelos Wiener-Volterra através de modelos *fuzzy*/neurais com estrutura FBO (Oliveira et al., 1999; Campello et al., 1999). Para esta finalidade, utiliza-se o modelo FBO relacional descrito na seção 5.3.3, que é representante de ambas as classes de modelos *fuzzy* e neurais. Viu-se que a partir da seleção *a priori* dos respectivos conjuntos referenciais/funções de ativação esses modelos tornam-se lineares nos seus demais parâmetros de projeto livre, assim como são os modelos Wiener-Volterra. Essa propriedade permite a estimação de ambos os modelos através de um mesmo algoritmo (mínimos quadrados). Uma mesma base ortonormal de Laguerre também é adotada para a implementação dos modelos.

5.8.1 Caso de Estudo: Tanque de Nível Líquido

Seja o processo real descrito em (Lindskog, 1996; Hellendoorn e Driankov, 1997). Esse processo consiste de um tanque de líquido em escala de laboratório, ilustrado na figura 5.4, onde a saída y é o nível de líquido no tanque (Centímetros) e a entrada u é a tensão (Volts) aplicada ao sistema que bombeia líquido para o tanque.

Dois conjuntos de dados distintos são disponíveis, sendo um para estimação e outro para validação de modelos desse processo. Cada conjunto possui 1000 pares de dados de entrada-saída amostrados a cada 1 segundo, em um total de pouco mais de 16 minutos de operação real do processo, conforme ilustrado na figura 5.5. As principais causas das não linearidades representadas nesses dados são: *i*) O fluxo de saída de líquido é aproximadamente proporcional a \sqrt{y} , segundo a lei de Bernoulli para orifícios pequenos. *ii*) Existe uma saturação em y devido à posição do orifício acima da base do tanque.

Inicialmente, determina-se o valor do pólo p que parametriza a base de funções de Laguerre utilizada para a implementação de ambos os modelos a serem estimados, ou seja, modelo FBO *fuzzy* da seção 5.3.3 e modelo Wiener-Volterra de segunda ordem da

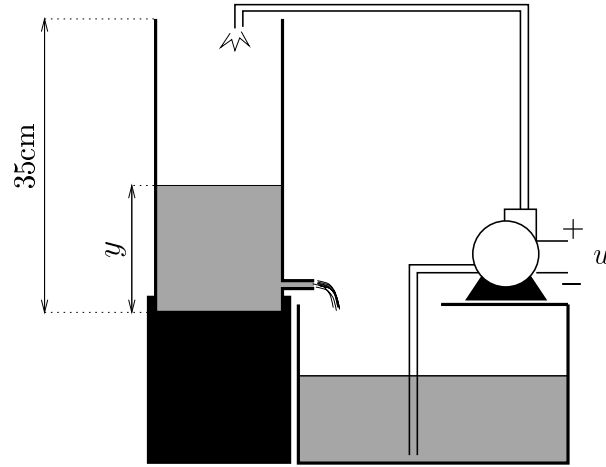


Figura 5.4: Figura esquemática do tanque de líquido em escala de laboratório.

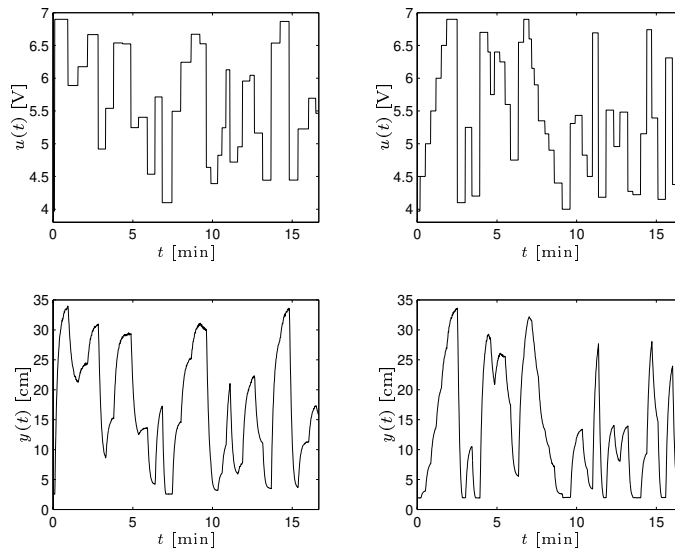


Figura 5.5: Dados de estimação (esquerda) e validação (direita) do tanque de líquido.

seção 5.3.2. Metodologias para a determinação desse pólo são discutidas na seção 5.5. Contudo, é importante lembrar que a escolha desse pólo não é crítica, isto é, uma escolha inadequada implica apenas o requerimento de um número maior de funções ortonormais para uma dada capacidade de aproximação dos modelos. Deve-se considerar também que a utilização de um mesmo pólo para os dois modelos minimiza (se não elimina) a influência da seleção desse pólo na comparação entre os desempenhos dos mesmos. Através de uma análise preliminar do conjunto de dados selecionou-se um pólo $p = 0.7$. Esse pólo aproxima, ao menos de forma grosseira, a dinâmica fundamental do sistema exibida nos dados.

Vários modelos com diferentes dimensões foram identificados. Para o modelo FBO *fuzzy* avaliaram-se diferentes combinações entre a quantidade “ n ” de funções de Laguerre e o número “ c ” de conjuntos referenciais *fuzzy* (Gaussianos) por variável de entrada. Especificamente, variou-se n entre 1 e 4 e c entre 2 e 3, valores além dos quais a razão entre a capacidade de representação e o número de parâmetros do modelo é sensivelmente reduzida. Os conjuntos referenciais foram distribuídos de forma homogênea sobre os universos de discurso das respectivas variáveis (estados de Laguerre), com centros equidistantes e aberturas iguais à distância entre dois centros consecutivos. Conforme discutido no capítulo 4, seção 4.2.4, essa heurística geralmente proporciona resultados satisfatórios, especialmente se considerado que não adiciona qualquer custo computacional ao procedimento de estimação do modelo. O modelo que apresentou a melhor capacidade de representação no que se refere ao conjunto de dados de validação foi aquele com $n = 4$ e $c = 2$, ou seja, com um montante de $\mu = c^n = 16$ parâmetros estimados utilizando a solução analítica dos mínimos quadrados dada pela equação (5.19). O Erro Quadrático Médio de previsão em série sintética desse modelo para os dados de validação é EQM = 1.2551. A simulação correspondente é ilustrada na figura 5.6.

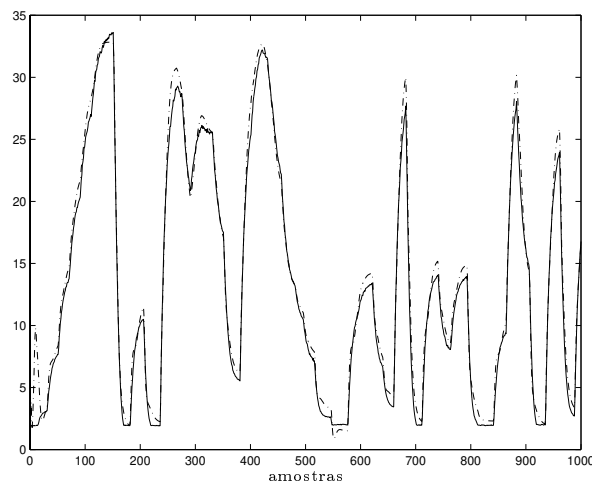


Figura 5.6: Saída do processo real (linha cheia) e previsão em série sintética do modelo FBO *fuzzy* (linha tracejada): Dados de validação.

Na figura 5.6 observa-se um erro mais acentuado no início da simulação, corrigido após algumas amostras. Esse erro é decorrente da ausência de informação sobre as condições iniciais do sistema nos estados dos filtros de Laguerre (inicializados nulos). Como esses filtros são estáveis, o erro tende a desaparecer assintoticamente com o tempo (Chen, 1998), restando apenas os erros de modelagem propriamente ditos. Opcionalmente, poder-se-ia separar uma parcela inicial dos dados de estimação e validação apenas para a geração de condições iniciais adequadas para os estados de Laguerre.

Para o modelo Wiener-Volterra fixou-se o número de funções de Laguerre do termo de segunda ordem em $N_2 = 4$ para limitar a quantidade de parâmetros a serem estimados. Variou-se então a quantidade N_1 de funções no termo de primeira ordem entre 1 e 20. O melhor resultado obtido na representação dos dados de validação refere-se a um erro de EQM = 1.2922, produzido por um modelo com $N_1 = 14$, isto é, com um total de $\mu = (N_2^2 + N_2 + 2N_1 + 2)/2 = 25$ parâmetros estimados pela solução analítica dos mínimos quadrados em (5.19).

Tem-se portanto que o modelo FBO *fuzzy* foi capaz de representar as dinâmicas do processo praticamente com a mesma precisão do modelo Wiener-Volterra, porém utilizando menos do que 2/3 da quantidade de parâmetros. Tem-se também que a qualidade do modelo FBO *fuzzy* poderia ainda ser aprimorada através da otimização *a priori* ou *a posteriori* dos conjuntos referenciais *fuzzy* Gaussianos que foram pré-definidos heurísticamente.

É importante notar que os resultados obtidos dependem diretamente das características da aplicação em questão. Esses resultados qualificam a abordagem FBO *fuzzy* proposta como uma possível alternativa no contexto de modelagem de sistemas dinâmicos não lineares, porém não reduzem a importância dos modelos Wiener-Volterra nesse mesmo contexto.

5.9 Aplicação: Planta para Produção de Etanol

Considera-se a seguir a modelagem de uma planta biotecnológica para produção de álcool etílico (etanol) em escala industrial. Essa planta é descrita em detalhes no apêndice B. No presente contexto, as grandezas de interesse são:

- F_a [m³/h]: Denota a vazão da matéria prima do processo, denominada Mosto. Essa variável constitui uma entrada manipulada cujo intervalo operacional é [50, 150].
- S_0 [kg/m³]: Denota a concentração de açúcar (ART) no Mosto e em condições reais de operação pode apresentar variações de até +/- 5% em torno do seu valor nominal (180kg/m³). Na presente aplicação, essa variável é considerada uma entrada de perturbação não mensurável atuando no processo.
- P_4 [kg/m³]: Denota a concentração de etanol no produto final bruto da planta e constitui a variável de saída de interesse.

5.9.1 Geração e Amostragem de Dados

Uma vez que se dispõe de um simulador validado para condições reais de operação industrial da planta (apêndice B), pode-se gerar de forma irrestrita conjuntos de dados de E/S representativos das sua dinâmicas. Dois conjuntos de dados independentes (I e II) são utilizados. Cada conjunto corresponde a 20 dias de operação da planta, sendo a primeira metade (10 dias/240h) utilizada para a estimação de modelos FBO dessa planta e a segunda metade utilizada para a validação desses modelos. Nesses dados, a variável manipulada F_a é uma seqüência de degraus, cada um dos quais com período de 10h (longo o suficiente para que a saída possa alcançar o estado estacionário⁷) e amplitude aleatória com distribuição de probabilidade uniforme no intervalo operacional [50, 150]. Os dados são amostrados utilizando um período $T = 30\text{min}$, projetado em função das dinâmicas do processo bem como de especificações técnicas dos seus principais instrumentos de medida (Meleiro et al., 2001).

O propósito de utilizar dois conjuntos de dados distintos (I e II) é avaliar o comportamento dos modelos em diferentes condições operacionais do processo. O conjunto de dados I é livre de ruído e foi gerado com S_0 constante em seu valor nominal $S_0 = 180\text{kg/m}^3$. O conjunto II foi gerado com o processo sujeito a uma perturbação Gaussiana em S_0 com média 180 e variância tal que essa perturbação apresenta $\approx 99\%$ de probabilidade de estar contida no intervalo [170, 190]. A parcela de estimação dos dados de saída do conjunto II foi ainda contaminada com ruído branco aditivo, conforme será discutido posteriormente.

5.9.2 Seleção de Estrutura

Estrutura, nesta seção, refere-se a dois conceitos distintos: *i*) O pólo p que parametriza as funções de Laguerre que serão utilizadas para a realização da dinâmica de estados dos modelos FBO; *ii*) As características estruturais internas de cada modelo FBO considerado. Na presente aplicação serão considerados todos os modelos discutidos na seção 5.3.

O pólo p foi selecionado experimentalmente com base em ensaios preliminares de resposta temporal do processo. Esses ensaios mostraram que a sua dinâmica dominante pode ser grosseiramente aproximada por um pólo real discreto dado por $p = 0.65$ (Campello, Meleiro, Amaral e Maciel Filho, 2001). Essa aproximação é válida em média, uma vez que a constante de tempo desse processo depende da amplitude da entrada F_a .

Uma vez que a base FBO esteja parametrizada, o modelo linear (afim) apresentado na seção 5.3.1 pode ser completamente especificado em função da quantidade n de funções/estados de Laguerre. No caso do modelo Wiener-Volterra da seção 5.3.2, pode-se arbitrar $N_1 = N_2$, o que implica $n = N_1 = N_2$. Conseqüentemente, esse modelo pode também ser completamente especificado em função de n . Os modelos FBO *fuzzy*

⁷Caso S_0 seja mantida constante em seu valor nominal.

das seções 5.3.3 e 5.3.4 requerem adicionalmente a especificação da quantidade “ c ” de conjuntos referenciais *fuzzy* por estado de Laguerre. Esses conjuntos são configurados da mesma forma descrita na seção 5.8.1 e, para evitar um número excessivo de parâmetros a serem estimados, o valor $c = 2$ é selecionado. Esse valor é capaz de proporcionar resultados adequados na presente aplicação, conforme mostrado na próxima seção.

5.9.3 Estimação dos Modelos FBO

Os universos de discurso das variáveis de E/S do processo possuem ordens de grandeza muito distintas. Essa característica torna necessária uma normalização dessas variáveis por razões de robustez numérica do procedimento de estimação dos modelos. Essa normalização é realizada de forma que os dados de E/S estejam contidos no intervalo comum $[-1, 1]$. Adota-se inicialmente o conjunto de dados I para a estimação dos diferentes modelos FBO. Os parâmetros desses modelos são estimados através da solução dos mínimos quadrados dada pela equação (5.19). Em seguida, os modelos são simulados utilizando a parcela de validação dos dados. As tabelas 5.1 a 5.4 apresentam os resultados de simulação na forma do erro quadrático médio entre a saída do processo e a simulação em série sintética dos modelos com a quantidade n de funções de Laguerre variando de 1 até 4, número além do qual o desempenho de nenhum deles pode ser melhorado de forma significativa.

Tabela 5.1: Desempenho de simulação do modelo linear para a parcela de validação do conjunto de dados I.

No. Funções n	No. Parâmetros μ	EQM
1	2	0.083075
2	3	0.077555
3	4	0.078434
4	5	0.078441

Pode-se observar nas tabelas que os modelos não lineares, como esperado, apresentaram erros inferiores (ao menos 10 vezes menores) do que aqueles associados ao modelo linear, incapaz de representar as não linearidades do processo. Pode-se observar também que os melhores resultados proporcionados pelos modelos de Volterra e *fuzzy* relacional são muito similares, enquanto os resultados do modelo *fuzzy* TS superou todos os demais. As simulações do melhor modelo de cada abordagem são apresentadas na figura 5.7. Essa figura ilustra que as dinâmicas do processo podem ser representadas de forma adequada pelos modelos FBO não lineares obtidos.

Observação 5.2 *Pode-se notar nas tabelas 5.1 e 5.4 que em alguns casos o acréscimo no número de funções de Laguerre acarretou uma piora no desempenho dos modelos.*

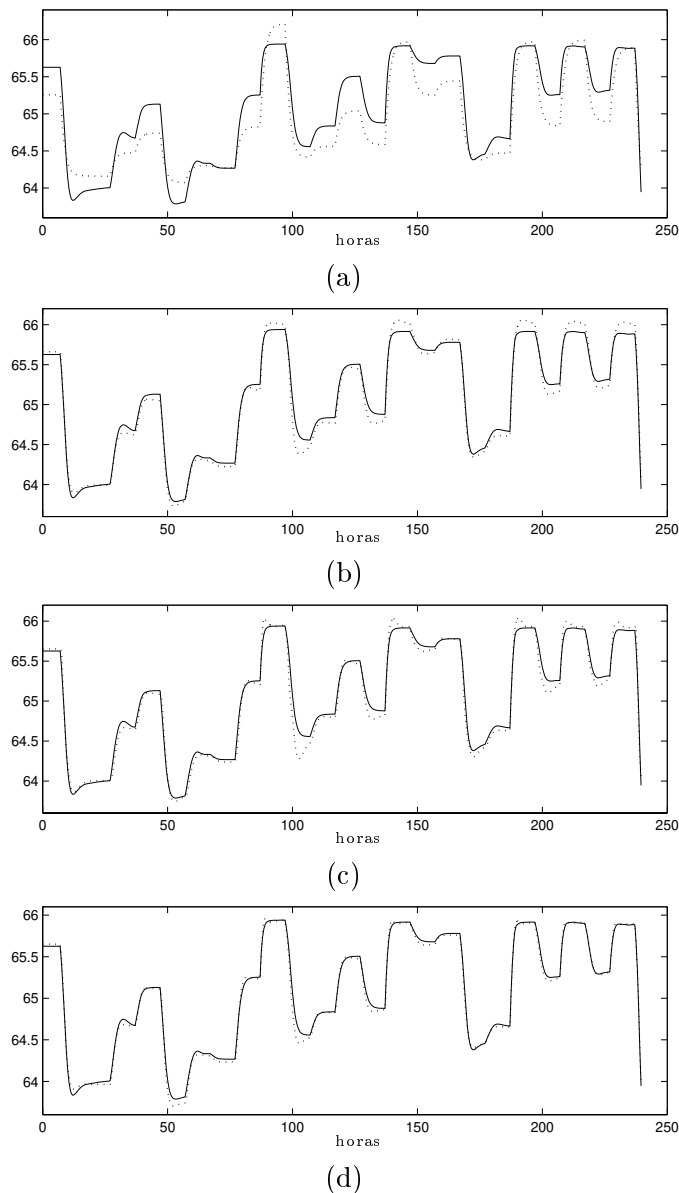


Figura 5.7: Concentração de etanol na saída do último reator (P_4 [kg/m³], linha contínua) e saída do modelo em série sintética (linha pontilhada) para a parcela de validação do conjunto de dados I: (a) Modelo linear, $n = 2$; (b) Modelo de Volterra, $n = 4$; (c) Modelo *fuzzy* relacional/RBF, $n = 4$; (d) Modelo *fuzzy* TS, $n = 3$.

Tabela 5.2: Desempenho de simulação do modelo de Volterra para a parcela de validação do conjunto de dados I.

No. Funções n	No. Parâmetros μ	EQM
1	3	0.013965
2	6	0.011572
3	10	0.007830
4	15	0.006791

Tabela 5.3: Desempenho de simulação do modelo *fuzzy* relacional/RBF para a parcela de validação do conjunto de dados I.

No. Funções n	No. Parâmetros μ	EQM
1	2	0.079631
2	4	0.035824
3	8	0.017794
4	16	0.007676

No caso do modelo linear esse comportamento deve-se predominantemente às dinâmicas não modeladas; por um lado, o modelo não é capaz de representar as não linearidades do processo e, por outro lado, um aumento no número de parâmetros eleva as incertezas estruturais refletidas na variância do estimador. Na abordagem TS este comportamento é menos significativo, dado que o modelo é não linear. Por esta razão, as causas também são menos evidentes. Nesse caso, o aumento do número de parâmetros pode causar um efeito adicional. Trata-se da redução da capacidade de generalização do modelo, isto é, a especialização do mesmo a um conjunto finito de dados utilizados para a estimação dos seus parâmetros com a conseqüente redução da sua capacidade de representar outros conjuntos de dados (validação). Logo, torna-se importante uma adequada distinção entre os conceitos de “capacidade de aproximação” e “desempenho” de um modelo.

Considera-se a seguir o conjunto de dados II para a estimação dos modelos FBO da planta de produção de etanol. Conforme mencionado na seção 5.9.1, esses dados foram gerados sujeitos a uma perturbação na concentração de açúcar do Mosto S_0 . A parcela de estimação dos dados de saída também é contaminada com ruído branco aditivo uniformemente distribuído em $[-0.2, 0.2]$, conforme ilustra a figura 5.8. Pode-se notar que a amplitude do ruído representa aproximadamente 10% do universo de discurso da saída.

Os desempenhos de simulação dos melhores modelos não lineares obtidos utilizando o conjunto de dados II são apresentados na tabela 5.5 e as respectivas curvas de simulação

Tabela 5.4: Desempenho de simulação do modelo *fuzzy* TS para a parcela de validação do conjunto de dados I.

No. Funções n	No. Parâmetros μ	EQM
1	4	0.007817
2	12	0.004638
3	32	0.002220
4	80	0.002664

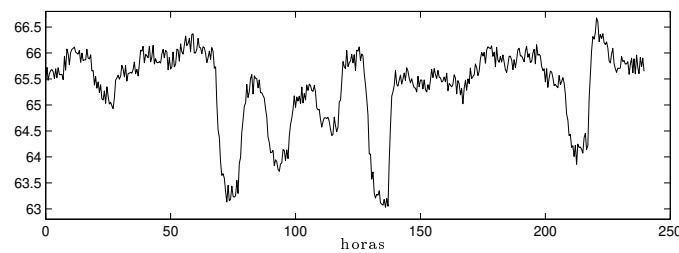


Figura 5.8: Medidas da concentração de etanol P_4 [kg/m³]: Parcela de estimação do conjunto de dados II.

são ilustradas na figura 5.9. O modelo linear foi desconsiderado pois apresentou desempenho muito pobre nessas condições operacionais (melhor modelo: EQM=0.124584).

Pode-se observar na figura 5.9 que os resultados dos diferentes modelos são qualitativamente equivalentes. Pode-se observar também que, apesar da perturbação não modelada S_0 , esses modelos permanecem seguindo as dinâmicas principais do sistema ao longo de um horizonte longo de simulação. Esse resultado é importante considerando a forte influência que S_0 exerce sobre o processo (Dechechi, 1998) e deve-se principalmente à ausência da realimentação dos erros de modelagem resultantes dessa perturbação. A realimentação dos erros poderia degradar ainda mais a previsão dos modelos e até mesmo causar a sua divergência (Campello et al., 1999; Oliveira et al., 1999). Essa análise será relevante na aplicação desses modelos ao controle preditivo do processo de produção de etanol, discutida no capítulo 6.

Observação 5.3 *A capacidade de aproximação dos modelos FBO não lineares poderia ser elevada aumentando a ordem da série de Volterra ou a quantidade de conjuntos referenciais dos mapeamentos fuzzy. Essa estratégia, no entanto, implicaria um crescimento significativo na quantidade de parâmetros dos respectivos modelos. No caso dos modelos FBO fuzzy, poder-se-ia alternativamente elevar essa capacidade através da otimização a priori (via clustering (Bezdek, 1981; Setnes, 2000)) e/ou a posteriori (via back-propagation (Jang, 1993; Wang, 1994)) dos conjuntos referenciais. Contudo, qualquer procedimento de otimização implicaria um custo computacional adicional*

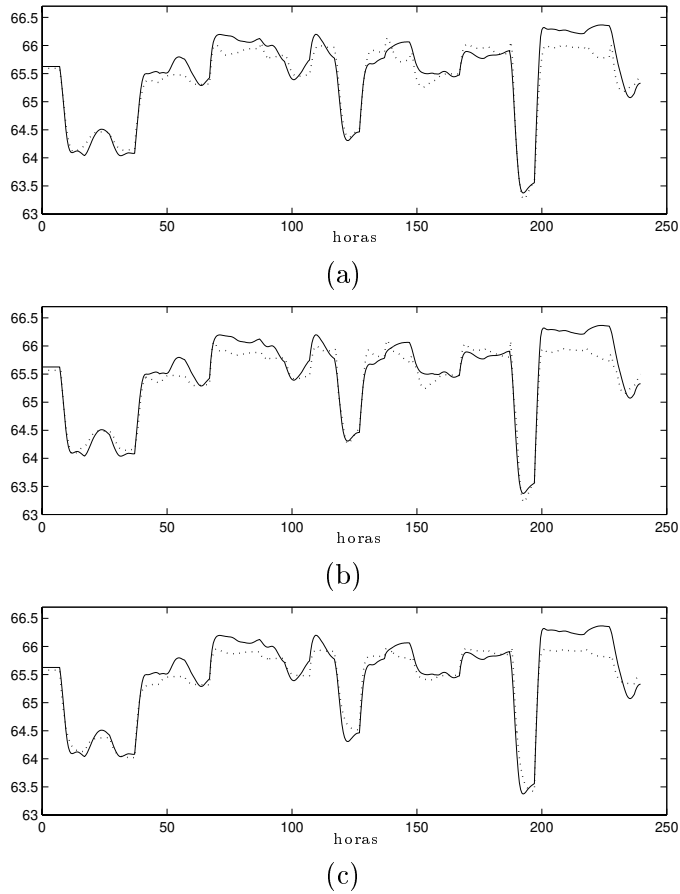


Figura 5.9: Concentração de etanol P_4 [kg/m^3] (linha contínua) e saída do modelo em série sintética (linha pontilhada) para a parcela de validação do conjunto de dados II: (a) Modelo de Volterra, $n = 4$; (b) Modelo *fuzzy* relacional/RBF, $n = 3$; (c) Modelo *fuzzy* Takagi-Sugeno, $n = 2$.

Tabela 5.5: Desempenhos de simulação dos melhores modelos FBO não lineares para a parcela de validação do conjunto de dados II.

Modelo	No. Funções n	No. Parâmetros μ	EQM
Volterra	3	10	0.044701
Volterra	4	15	0.042632
Relacional/RBF	3	8	0.043261
Relacional/RBF	4	16	0.043357
TS	2	12	0.043896
TS	3	32	0.049164

que tornaria complexa uma comparação adequada de desempenho entre os diferentes modelos.

5.10 Modelos FBO Multivariáveis

Um modelo FBO com múltiplas entradas e uma saída (MISO) pode ser genericamente representado como:

$$\hat{y}(k) = \mathcal{H}(\mathbf{l}_1(k), \dots, \mathbf{l}_q(k)) \quad (5.30)$$

onde q é o número de entradas, $\mathbf{l}_j(k) = [l_{j_1}(k), \dots, l_{j_{n_j}}(k)]^T$ é o vetor de estados ortonormais para a j -ésima entrada e \mathcal{H} é um mapeamento estático genérico. Como o modelo MISO em (5.30) não possui qualquer influência (recursão) da variável de saída, um modelo MIMO com s saídas pode ser obtido diretamente através de um conjunto de s modelos MISO independentes⁸. Portanto, por simplicidade e sem qualquer perda de generalidade, considera-se aqui apenas o caso MISO.

O vetor de estados referente a cada entrada do modelo é descrito por uma equação dinâmica independente, isto é:

$$\mathbf{l}_j(k+1) = A_j \mathbf{l}_j(k) + \mathbf{b}_j u_j(k) \quad (5.31)$$

onde u_j é a j -ésima entrada. No caso da base Laguerre, a representação de estados acima é caracterizada por:

⁸Esse desacoplamento natural constitui uma das vantagens da estrutura FBO.

$$A_j = \begin{bmatrix} p_j & 0 & \cdots & 0 \\ 1 - p_j^2 & p_j & \cdots & 0 \\ (-p_j)(1 - p_j^2) & 1 - p_j^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (-p_j)^{n_j-2}(1 - p_j^2) & (-p_j)^{n_j-3}(1 - p_j^2) & \cdots & p_j \end{bmatrix} \quad (5.32)$$

$$\mathbf{b}_j = \sqrt{1 - p_j^2} [1 \quad -p_j \quad (-p_j)^2 \quad \cdots \quad (-p_j)^{n_j-1}]^T \quad (5.33)$$

onde n_j e $p_j \in \{\Re : -1 < p_j < 1\}$ são respectivamente a quantidade de estados e o pólo de Laguerre referentes à j -ésima entrada. A figura 5.10 ilustra um modelo FBO com múltiplas (q) entradas.

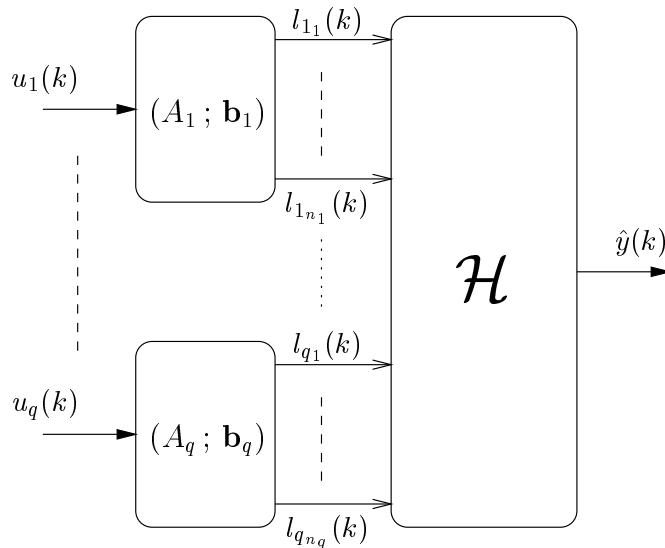


Figura 5.10: Modelo FBO com múltiplas entradas.

5.11 Modelos FBO Hierárquicos

No capítulo 4 discutiu-se o problema da dimensionalidade em modelos *fuzzy* e uma alternativa para a redução desse problema através da implementação desses modelos segundo uma arquitetura hierárquica composta de uma série em cascata de submodelos de dimensão reduzida. O problema da dimensionalidade pode também ser crítico nos modelos *fuzzy* com estrutura FBO, especialmente na representação de sistemas com múltiplas entradas. Torna-se então importante a extensão da estratégia de modelagem hierárquica para o contexto dos modelos com estrutura FBO.

A idéia da arquitetura hierárquica com estrutura dinâmica FBO é implementar o operador \mathcal{H} na equação (5.30) através de um modelo *fuzzy* hierárquico. O objetivo é obter um modelo com um número reduzido de parâmetros de projeto livre. Uma questão fundamental no contexto de modelagem hierárquica diz respeito à seleção da ordem hierárquica das variáveis de entrada do modelo. No capítulo 4 mostrou-se ser mais eficaz posicionar as variáveis associadas às relações de entrada-saída mais complexas e relevantes do sistema a ser modelado nos níveis hierárquicos mais elevados do modelo.

Na estrutura dinâmica FBO sabe-se *a priori* que, a partir da convergência da série de funções ortonormais, a importância de cada função adicional para a capacidade de aproximação do modelo é decrescente. Logo, tem-se que uma escolha natural para a ordem hierárquica no caso monovariável ($q = 1$) é dada pela própria ordem dos estados ortonormais, isto é, $l_{1_1}(k), l_{1_2}(k), \dots, l_{1_{n_1}}(k)$. No caso multivariável, no entanto, existem múltiplas possibilidades. As duas possibilidades mais evidentes são ilustradas nas figuras 5.11 e 5.12, onde $\mathbf{l}_1 = [l_{1_1} \ l_{1_2}]^T$ e $\mathbf{l}_2 = [l_{2_1} \ l_{2_2}]^T$ ($n_1 = n_2 = 2$) são os vetores de estados associados às entradas u_1 e u_2 do modelo ($q = 2$). Na primeira, a ordem hierárquica é determinada pela ordem dos estados. Na segunda, a ordem é determinada pela predominância de uma das entradas, i.e., $u_1(k)$, cujos estados estão posicionados nos níveis hierárquicos mais altos.

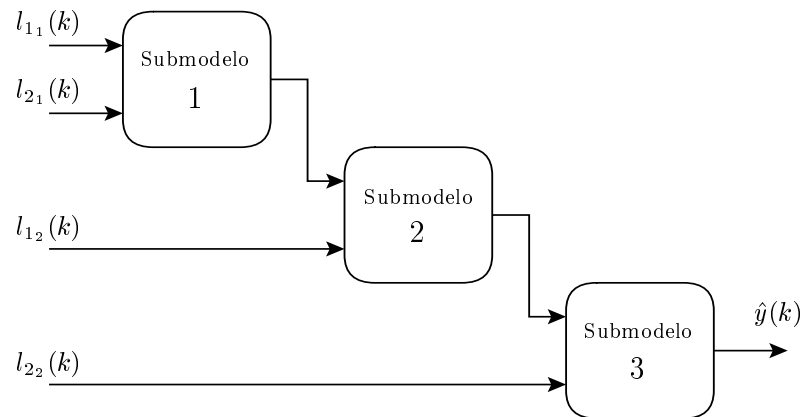


Figura 5.11: Arquitetura A de modelo FBO hierárquico com duas entradas e dois estados ortonormais cada.

Se nenhuma distinção hierárquica for desejada entre as diferentes entradas do modelo, uma alternativa é a utilização de uma estrutura constituída da composição de múltiplos modelos hierárquicos independentes, como ilustrado na figura 5.13. A composição dos diferentes modelos é representada pelo operador genérico Ξ , que pode ser um mapeamento *fuzzy*, neural ou mesmo qualquer operador algébrico simples.

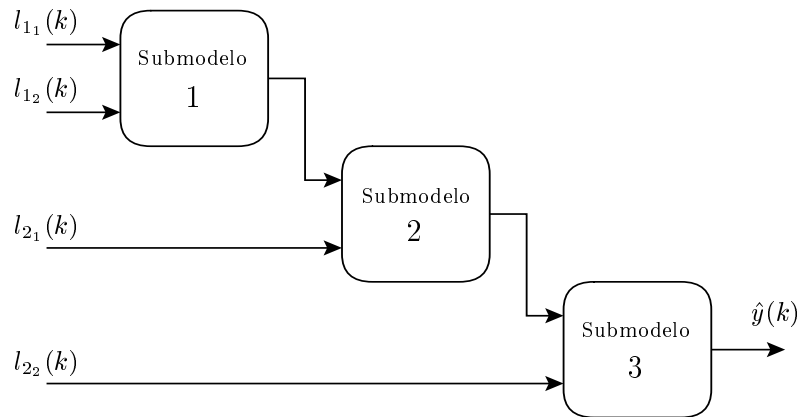


Figura 5.12: Arquitetura B de modelo FBO hierárquico com duas entradas e dois estados ortonormais cada.

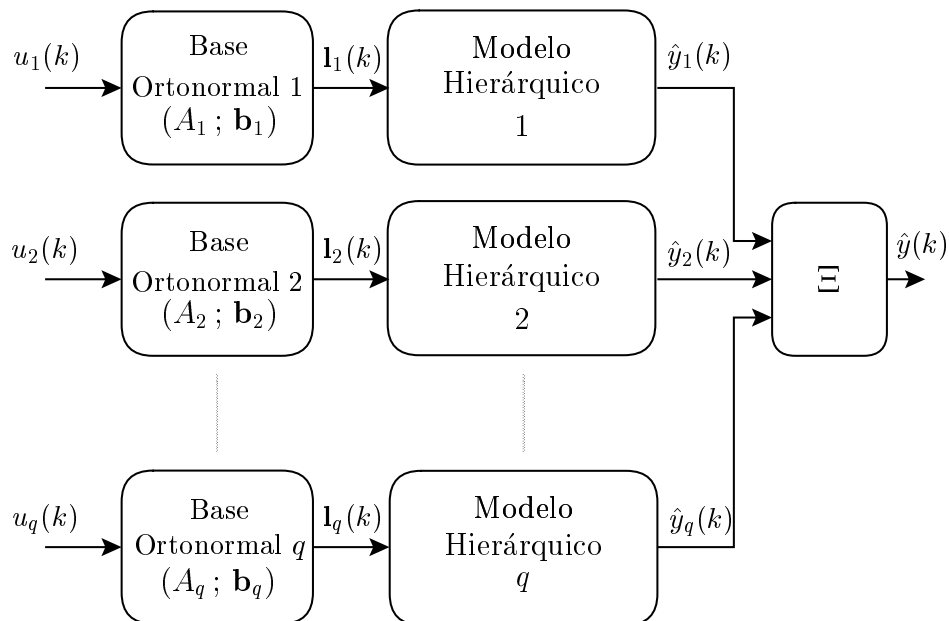


Figura 5.13: Arquitetura alternativa de modelo hierárquico FBO com múltiplas entradas.

5.11.1 Aplicação: Planta para Produção de Etanol

Considera-se aqui o problema de modelagem da planta biotecnológica para produção de etanol descrita em detalhes no apêndice B. Essa mesma planta foi utilizada na seção 5.9 como um exemplo de aplicação dos modelos FBO apresentados na seção 5.3. Naquela aplicação, a concentração de açúcar no mosto S_0 foi tratada como uma perturbação não mensurável/modelada. No presente contexto, no entanto, essa variável será considerada uma perturbação mensurável e inserida como entrada em um modelo FBO hierárquico a ser estimado.

Geração e Amostragem dos Dados

Um conjunto representativo de sinais de E/S do processo referentes a dois meses (1440h) de sua operação simulada é gerado para a identificação do modelo FBO hierárquico. Assim como na seção 5.9, os sinais são amostrados a cada período de 30min. Com essa amostragem, obtém-se um conjunto de dados discretos no tempo com um total de 2880 amostras de E/S, sendo a primeira metade destinada à estimação do modelo FBO hierárquico do processo e a segunda metade destinada à validação desse modelo. Os dados são gerados com a variável manipulada F_a sendo uma seqüência de degraus, cada um dos quais com período de 10h e amplitude aleatória uniformemente distribuída no intervalo operacional $[50, 150]$. A perturbação S_0 apresenta o mesmo comportamento, porém os degraus possuem período de 20h e estão contidos no intervalo $[170, 190]$ (i.e., aproximadamente $\pm 5\%$ em torno do valor nominal $S_0 = 180\text{kg/m}^3$).

Seleção de Estrutura

Estrutura, no presente contexto, refere-se a três conceitos distintos: *i*) Os pólos de Laguerre associados a cada entrada do modelo FBO; *ii*) A ordem hierárquica dos estados de Laguerre como entradas do mapeamento *fuzzy* hierárquico; *iii*) A configuração estrutural interna desse mapeamento.

Os pólos de Laguerre foram selecionados através da mesma estratégia adotada e descrita na seção 5.9, baseada em ensaios preliminares da resposta temporal do processo. Especificamente, o pólo de Laguerre associado à entrada F_a foi selecionado como $p_1 = 0.65$, enquanto o pólo associado à entrada S_0 foi selecionado como $p_2 = 0.9$.

Os submodelos hierárquicos são implementados utilizando a abordagem relacional simplificada discutida na seção 4.2.2 (capítulo 4). A ordem hierárquica dos estados de Laguerre é definida como na figura 5.11, de forma que cada estado associado à variável F_a (a entrada manipulada) seja posicionado em um nível hierárquico imediatamente acima ao nível hierárquico do estado correspondente associado a S_0 .

Uma vez que os pólos de Laguerre e a ordem hierárquica já tenham sido definidos, o modelo pode ser completamente especificado em termos das quantidades n_1 e n_2 de estados de Laguerre por variável de entrada (F_a e S_0) bem como do número “ c ” de conjuntos referenciais *fuzzy* (Gaussianos) por estado. Na seção 5.9 utilizou-se o valor

$c = 2$ para a modelagem das dinâmicas do processo referentes à entrada manipulada F_a . Em função da complexidade das dinâmicas adicionais impostas pela presença da entrada de perturbação modelada S_0 , adota-se aqui um valor maior, dado por $c = 3$. Esse valor é usual no contexto de aplicações de lógica *fuzzy* e, associado a uma quantidade $n_1 = n_2 = 2$ de estados de Laguerre por variável de entrada, é capaz de proporcionar resultados satisfatórios na modelagem do processo de produção de etanol, conforme discutido a seguir.

Estimação e Validação do Modelo

Após concluído o projeto estrutural, os parâmetros desconhecidos do modelo resumem-se aos parâmetros de projeto livre do mapeamento *fuzzy* hierárquico. Esses parâmetros são dados pelos vetores relacionais $\Omega_{(\cdot)}$ na equação (4.5) bem como os centros ($\theta_{(\cdot)}$ e $\phi_{(\cdot)}$) e aberturas ($\sigma_{(\cdot)}$ e $\varphi_{(\cdot)}$) dos conjuntos referenciais *fuzzy* nas equações (4.9) e (4.10).

Na presente aplicação, a estimação desses parâmetros será realizada através do algoritmo híbrido discutido na seção 4.2.7. Nesse algoritmo os parâmetros são otimizados em duas fases: Inicialmente, um modelo inicial aproximado é obtido utilizando um algoritmo genético (detalhado no apêndice C). Em seguida, realiza-se uma sintonia fina dos parâmetros resultantes através do algoritmo do gradiente conjugado de Fletcher-Reeves, cujos detalhes no contexto de modelagem hierárquica foram discutidos na seção 4.2.3

Experimentos preliminares com o GA descrito no apêndice C, baseados em problemas de modelagem mais simples, mostraram que esse algoritmo não é muito sensível à sua configuração caso uma quantidade representativa de indivíduos na população seja utilizada (duas ou três vezes o número de parâmetros a serem otimizados). Uma configuração adequada, por exemplo, é dada por: $P_r = 0.95$, $P_m = 0.1$, $N_e = 1$ e $N_d = 20\%$ de N_I , sendo que esses termos representam respectivamente a probabilidade de reprodução, a probabilidade de mutação, o número de indivíduos para elitismo, o número de indivíduos para diversidade e o número total de indivíduos na população. Essa configuração será utilizada na presente aplicação.

O objetivo da fase inicial de otimização do modelo através do GA é obter uma solução inicial adequada à aplicação do algoritmo de gradiente. Logo, nesta fase pode-se otimizar apenas os vetores $\Omega_{(\cdot)}$ de parâmetros relacionais. Esses vetores são inicializados aleatoriamente no intervalo de normalização $[-1, 1]$, conforme discutido na seção 4.2.4. Os conjuntos referenciais *fuzzy*, por sua vez, podem ser mantidos constantes em sua distribuição inicial, também discutida na seção 4.2.4. Como o modelo hierárquico possui um total de $n = n_1 + n_2 = 4$ entradas, os vetores $\Omega_{(\cdot)}$ possuem $(n-1)c^2 = 27$ parâmetros a serem otimizados pelo GA. Seleciona-se então uma quantidade de $N_I = 100$ indivíduos na população do algoritmo, que é um número em torno de três vezes a quantidade de parâmetros, conforme sugerido anteriormente.

O GA é executado por $N_g = 100$ gerações que representam a primeira fase do algoritmo híbrido. Na segunda fase, o modelo final gerado pelo GA é refinado por 100 épocas

de treinamento utilizando o algoritmo de Fletcher-Reeves. Nessa fase, que também inclui a otimização dos conjuntos referenciais *fuzzy*, o número total de parâmetros otimizados é $\mu = 59$. Um modelo completo não hierárquico com o mesmo número de entradas (4) e conjuntos referenciais por entrada (3) necessitaria otimizar um total de 105 parâmetros (ver seção 4.2.1).

A evolução do erro quadrático médio entre as saídas do processo e do modelo é apresentada na figura 5.14, onde as primeiras 100 iterações referem-se ao GA e as demais referem-se ao algoritmo de Fletcher-Reeves.

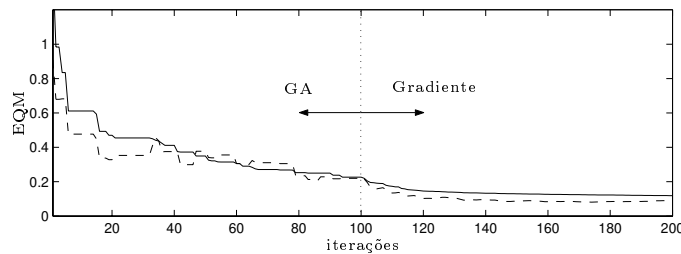


Figura 5.14: Evolução do erro quadrático médio entre as saídas do processo e do modelo ao longo do procedimento híbrido de otimização: Dados de estimação (linha contínua) e validação (linha tracejada).

A simulação do modelo final utilizando os dados de validação é ilustrada na figura 5.15 (EQM=0.089). Pode-se observar que o modelo representa de forma adequada as dinâmicas do processo, especialmente considerando o longo horizonte de previsão envolvido na simulação. O projeto de um sistema de controle preditivo da planta de produção de etanol baseado nesse modelo é descrito em (Campello et al., nd).

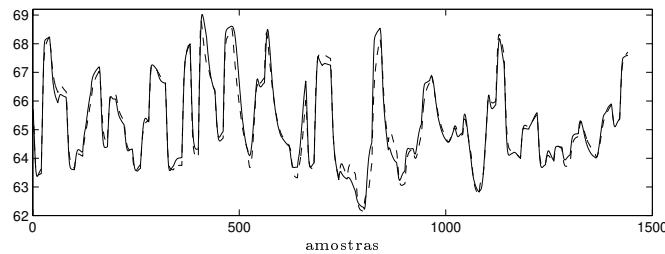


Figura 5.15: Saída do processo (P_4 [kg/m³], linha contínua) e série sintética do modelo (linha tracejada) para dados de validação.

Observação 5.4 *Os parâmetros estruturais do modelo, ou seja, a quantidade e a parametrização das funções de Laguerre bem como a ordem hierárquica e a quantidade de conjuntos referenciais fuzzy, poderiam ter sido incluídos no algoritmo genético para serem determinados de forma automática. Essa estratégia, contudo, elevaria significativamente a complexidade do algoritmo e dos seus requerimentos computacionais,*

especialmente por envolver um problema estrutural, e não está do escopo do presente trabalho.

5.12 Redução de Ordem em Modelos Wiener-Volterra pela Otimização da Base de Laguerre

Nesta seção aborda-se o problema de seleção ótima de bases de Laguerre para o desenvolvimento de modelos de Volterra em funções ortonormais. O objetivo é minimizar o número de funções necessárias para proporcionar a esse desenvolvimento uma dada precisão, simplificando assim os problemas de identificação e controle que poderão estar associados aos modelos Wiener-Volterra resultantes.

5.12.1 Definição do Problema

Seja o seguinte modelo de Volterra de ordem η :

$$y(k) = \sum_{m=1}^{\eta} \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, k_2, \dots, k_m) \prod_{j=1}^m u(k - k_j) \quad (5.34)$$

Viu-se no capítulo 2 que os kernels $h_{(\cdot)}$ em (5.34) podem ser desenvolvidos através de funções ortonormais desde que eles sejam quadraticamente somáveis em $[0, \infty[$. Propõe-se aqui a utilização de uma base independente de funções para o desenvolvimento de cada kernel (Campello, Amaral e Favier, 2001). Nesse caso, tem-se que o desenvolvimento do m -ésimo kernel é dado por:

$$h_m(k_1, \dots, k_m) = \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m} \prod_{j=1}^m \phi_{m, i_j}(k_j) \quad (5.35)$$

onde $\phi_{m, l}$ é a l -ésima função ortonormal da m -ésima base de funções e $\alpha_{(\cdot)}$ são os coeficientes da expansão, dados por:

$$\alpha_{i_1, \dots, i_m} = \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, \dots, k_m) \prod_{j=1}^m \phi_{m, i_j}(k_j) \quad (5.36)$$

A partir das equações (5.34) e (5.35) tem-se:

$$y(k) = \sum_{m=1}^{\eta} \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m} \prod_{j=1}^m \bar{\phi}_{m, i_j}(k) \quad (5.37)$$

onde $\bar{\phi}_{m,l}$ é a saída do l -ésimo filtro de Laguerre (com entrada u) da m -ésima base de funções, como segue:

$$\bar{\phi}_{m,l}(k) = \sum_{\tau=0}^{\infty} \phi_{m,l}(\tau)u(k - \tau) \quad (5.38)$$

As funções de transferência dos filtros de Laguerre dessa base são dadas por:

$$\Phi_{m,l}(z) = Z\{\phi_{m,l}\} = \frac{\sqrt{1-p_m^2}}{z-p_m} \left(\frac{1-p_m z}{z-p_m} \right)^{l-1}; \quad l = 1, 2, \dots \quad (5.39)$$

onde Z denota a transformada z unilateral e $p_m \in (-1, 1)$ é o pólo real estável que parametriza as funções.

O problema que se coloca é como encontrar os pólos p_m ($m = 1, \dots, \eta$) que minimizam os erros de aproximação do desenvolvimento truncado dos respectivos kernels em um número finito de funções de Laguerre. Esse truncamento é necessário pois permite a implementação computacional do modelo.

5.12.2 Desenvolvimento Ótimo do Kernel de Primeira Ordem

O pólo p_1 que parametriza o desenvolvimento do kernel de 1a. ordem h_1 (resposta ao impulso de um sistema linear) pode ser obtido através da solução do seguinte problema de otimização (Fu e Dumont, 1993):

$$\min_{-1 < p_1 < 1} J_1 \triangleq \sum_{i_1=1}^{\infty} i_1 \alpha_{i_1}^2 \quad (5.40)$$

cuja solução força uma rápida convergência da série de funções através de um aumento linear do custo atribuído a cada coeficiente adicional. No trabalho citado acima, os autores mostraram que o problema (5.40) possui uma solução analítica global estrita se h_1 for absolutamente somável em $[0, \infty[$ (sistema linear BIBO estável), porém não nulo, e possuir ao menos uma unidade de atraso, i.e., $h_1(0) = 0$ (sistema estritamente próprio). Em (Campello, Amaral e Favier, 2001) esse resultado foi estendido para modelos de Volterra de 2a. ordem. Basicamente propôs-se a utilização de duas bases de Laguerre independentes para o desenvolvimento dos kernels, o que possibilitou a aplicação da solução de Fu e Dumont para a determinação da base referente ao kernel de 1a. ordem, e obteve-se uma solução estendida para o desenvolvimento do kernel de 2a. ordem. A seguir generalizam-se esses resultados para o desenvolvimento ótimo de kernels de qualquer ordem.

5.12.3 Desenvolvimento Ótimo do Kernel de m -ésima Ordem

Apresenta-se abaixo a generalização dos resultados mencionados na seção 5.12.2 no que diz respeito à determinação do pólo de Laguerre p_m responsável pelo desenvolvimento do kernel de Volterra de ordem m qualquer.

Teorema 5.1 *Assuma que o kernel de Volterra de ordem m apresenta as seguintes propriedades:*

1. $\sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} |h_m(k_1, \dots, k_m)| < \infty$.
2. $h_m(k_1, \dots, k_m) = 0$ para $\forall k_l = 0$ com $l \in \{1, \dots, m\}$.

ou seja, é absolutamente somável em $[0, \infty[$ (kernel estável) e não possui resposta instantânea. Nesse caso, tem-se que a seguinte igualdade é verdadeira:

$$\begin{aligned} J_m &\triangleq \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} (i_1 + \cdots + i_m) \alpha_{i_1, \dots, i_m}^2 \\ &= \left(\frac{p_m Q_{2,m} + (1 - p_m)^2 Q_{1,m} + p_m - p_m^2}{1 - p_m^2} \right) m \|h_m\|^2 \end{aligned} \quad (5.41)$$

onde $Q_{1,m}$, $Q_{2,m}$ e $\|h_m\|$ são constantes que dependem exclusivamente do kernel de ordem m , como segue:

$$\|h_m\|^2 = \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, \dots, k_m)^2 \quad (5.42)$$

$$Q_{j,m} = \frac{1}{m} \sum_{l=1}^m S_{j,l} \quad (5.43)$$

com $j = 1, 2$ e

$$S_{1,l} = \frac{1}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l h_m(k_1, \dots, k_m)^2 \quad (5.44)$$

$$S_{2,l} = \frac{1}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l [\Delta_l h_m(k_1, \dots, k_m)]^2 \quad (5.45)$$

sendo $l = 1, \dots, m$ e o operador Δ_l definido sobre h_m como

$$\Delta_l h_m(k_1, \dots, k_m) = h_m(k_1, \dots, k_l + 1, \dots, k_m) - h_m(k_1, \dots, k_l, \dots, k_m) \quad (5.46)$$

Prova: A prova é apresentada no apêndice D. ■

Observação 5.5 Embora a primeira condição do teorema 5.1 necessariamente implique $\|h_m\|^2 < \infty$ (Desoer e Vidyasagar, 1975), o mesmo não ocorre para os termos $S_{1,l}$ e $S_{2,l}$ em (5.44) e (5.45), respectivamente. A convergência desses termos é necessária para que o critério J_m em (5.41) seja bem definido e possa ser otimizado, sendo garantida apenas para modelos de primeira ordem descrevendo sistemas lineares⁹. No caso geral de ordem m , a convergência depende da taxa de decaimento do kernel, isto é, a taxa com que h_m estável tende a zero conforme k_l ($\forall l \in \{1, \dots, m\}$) tende ao infinito. Uma condição suficiente para a convergência é assumir que o kernel, além de ser não nulo ($\|h_m\|^2 \neq 0$), possua memória finita (Eykhoff, 1974), isto é, $h_m(k_1, \dots, k_m) = 0$ para $k_l > \epsilon_m$ ($\forall l \in \{1, \dots, m\}$) com $\epsilon_m < \infty$. Essa hipótese leva à usual representação truncada do kernel, adotada nas implementações computacionais dos modelos de Volterra (ver capítulo 2).

Supondo, a partir da observação 5.5, que os termos $S_{1,l}$ e $S_{2,l}$ em (5.44) e (5.45) são bem definidos, pode-se otimizar o pólo p_m que parametriza o desenvolvimento do kernel h_m através da solução do seguinte problema:

$$\min_{p_m \in \mathcal{P}} J'_m(p_m) = \frac{(Q_{1,m} - 1)p_m^2 + (Q_{2,m} - 2Q_{1,m} + 1)p_m + Q_{1,m}}{1 - p_m^2} \quad (5.47)$$

onde $J'_m \triangleq \frac{J_m}{m \|h_m\|^2}$ e $\mathcal{P} \triangleq \{p_m \in \mathfrak{R} : |p_m| < 1\}$. Pode-se observar, a partir de (5.41), que o problema em (5.47) é uma generalização para o kernel de ordem m daquele em (5.40), proposto por Fu e Dumont (1993) para o desenvolvimento de uma resposta ao impulso h_1 . Antes de seguir com a solução desse problema, faz-se necessária a apresentação de alguns lemas:

Lema 5.1 Se o kernel h_m satisfizer a segunda condição do teorema 5.1, então $Q_{1,m} - 1 \geq 0$.

Prova: A prova encontra-se no apêndice D. ■

⁹Onde a resposta ao impulso h_1 apresenta decaimento exponencial.

Corolário 5.1 *O lema 5.1 implica $2Q_{1,m} - 1 > 0$.*

Prova: A prova é evidente dado que $Q_{1,m} \geq 1$. ■

Lema 5.2 *Se o kernel h_m satisfizer as condições do teorema 5.1, então $4Q_{1,m}Q_{2,m} - Q_{2,m}^2 - 2Q_{2,m} > 0$.*

Prova: A prova encontra-se no apêndice D. ■

Lema 5.3 *Seja uma função $f : \mathcal{X} \rightarrow \Re$ definida como*

$$f(\mathbf{x}) \triangleq \frac{\omega(\mathbf{x})}{\gamma(\mathbf{x})} \quad (5.48)$$

onde \mathcal{X} é um conjunto convexo aberto tal que $\mathcal{X} \subset \Re^n$ e

- ω é uma função convexa, diferenciável e não negativa para todo $\mathbf{x} \in \mathcal{X}$.
- γ é uma função côncava, diferenciável e positiva para todo $\mathbf{x} \in \mathcal{X}$.

Tem-se então que f é uma função pseudo-convexa em \mathcal{X} , i.e., $\nabla f(\mathbf{x}_1)^T(\mathbf{x}_2 - \mathbf{x}_1) \geq 0$ implica $f(\mathbf{x}_2) \geq f(\mathbf{x}_1)$ para qualquer par de elementos $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ (ver (Bazaraa et al., 1993), cap. 3, ex. 3.60).

Lema 5.4 *A função $J'_m(p_m)$ em (5.47) é pseudo-convexa em $\mathcal{P} = \{p \in \Re : |p| < 1\}$.*

Prova:

1. \mathcal{P} é um conjunto convexo aberto.
2. $\gamma(p_m) \triangleq 1 - p_m^2$ é côncava, diferenciável e positiva para qualquer $p_m \in \mathcal{P}$.
3. $\omega(p_m) \triangleq (Q_{1,m} - 1)p_m^2 + (Q_{2,m} - 2Q_{1,m} + 1)p_m + Q_{1,m}$ é convexa $(Q_{1,m} - 1 \geq 0)^{10}$, diferenciável e não negativa¹¹ para todo $p_m \in \mathcal{P}$.

A conclusão da prova segue então diretamente do lema 5.3. ■

¹⁰Ver lema 5.1.

¹¹Caso contrário J'_m (e J_m) seriam negativos, o que por definição não é possível.

Teorema 5.2 (Teorema Principal) *O problema (5.47) possui uma solução global estrita dada por*

$$p_m^* = \frac{2Q_{1,m} - 1 - Q_{2,m}}{2Q_{1,m} - 1 + \sqrt{4Q_{1,m}Q_{2,m} - Q_{2,m}^2 - 2Q_{2,m}}} \quad (5.49)$$

se o kernel h_m satisfizer as condições do teorema 5.1.

Prova:

Uma vez que J'_m é uma função pseudo-convexa no domínio \mathcal{P} (lema 5.4), qualquer solução para $dJ'_m/dp_m = 0$ em \mathcal{P} é uma solução global para o problema (5.47) (Bazarraa et al., 1993). Calculando a derivada de J'_m com relação a p_m obtém-se

$$\frac{dJ'_m}{dp_m} = \frac{(Q_{2,m} - 2Q_{1,m} + 1)p_m^2 + (2Q_{1,m} - 1)2p_m + (Q_{2,m} - 2Q_{1,m} + 1)}{(1 - p_m^2)^2} \quad (5.50)$$

Supondo que $Q_{2,m} - 2Q_{1,m} + 1 \neq 0$ (caso contrário a solução será claramente $p_m^* = 0$)¹², a derivada em (5.50) será nula se e somente se

$$p_m^2 + \frac{(2Q_{1,m} - 1)2p_m}{(Q_{2,m} - 2Q_{1,m} + 1)} + 1 = 0 \quad (5.51)$$

As soluções da equação (5.51) são

$$p_{m,1} = \frac{2Q_{1,m} - 1 - \sqrt{4Q_{1,m}Q_{2,m} - Q_{2,m}^2 - 2Q_{2,m}}}{2Q_{1,m} - 1 - Q_{2,m}} \quad (5.52)$$

e

$$p_{m,2} = \frac{2Q_{1,m} - 1 + \sqrt{4Q_{1,m}Q_{2,m} - Q_{2,m}^2 - 2Q_{2,m}}}{2Q_{1,m} - 1 - Q_{2,m}} \quad (5.53)$$

Pode-se verificar que $p_{m,1}p_{m,2} = 1$. Logo, tem-se que $p_{m,1}$ e $p_{m,2}$ possuem o mesmo sinal. Tem-se ainda que $2Q_{1,m} - 1 > 0$ (corolário 5.1) e $4Q_{1,m}Q_{2,m} - Q_{2,m}^2 - 2Q_{2,m} > 0$ (lema 5.2), o que implica

$$2Q_{1,m} - 1 + \sqrt{4Q_{1,m}Q_{2,m} - Q_{2,m}^2 - 2Q_{2,m}} > 0. \quad (5.54)$$

¹²Esse resultado pode também ser verificado através da equação (5.49).

Conseqüentemente, o sinal de $p_{m,2}$ e $p_{m,1}$ é determinado pelo denominador comum em (5.52) e (5.53). Ambas as hipóteses $2Q_{1,m} - 1 - Q_{2,m} > 0$ e $2Q_{1,m} - 1 - Q_{2,m} < 0$ implicam

$$2Q_{1,m} - 1 - \sqrt{4Q_{1,m}Q_{2,m} - Q_{2,m}^2 - 2Q_{2,m}} > 0 \quad (5.55)$$

e, portanto, $|p_{m,1}| < |p_{m,2}|$. Como $p_{m,1}p_{m,2} = 1$, tem-se $|p_{m,1}| < 1$ ($p_{m,1} \in \mathcal{P}$) e $|p_{m,2}| > 1$ ($p_{m,2} \notin \mathcal{P}$). Logo, a solução de (5.47) é

$$p_m^* = p_{m,1} = \frac{1}{p_{m,2}} = \frac{2Q_{1,m} - 1 - Q_{2,m}}{2Q_{1,m} - 1 + \sqrt{4Q_{1,m}Q_{2,m} - Q_{2,m}^2 - 2Q_{2,m}}} \quad (5.56)$$

■

5.12.4 Exemplo

Seja um modelo de Volterra de segunda ordem ($\eta = 2$) em (5.34) com os kernels h_1 e h_2 dados por:

$$h_1(k_1) = \begin{cases} 0 & \text{para } k_1 = 0 \\ e^{-\rho k_1} & \text{para } k_1 = 1, 2, \dots \end{cases} \quad (5.57)$$

$$h_2(k_1, k_2) = \begin{cases} 0 & \text{para } k_1 \text{ ou } k_2 = 0 \\ e^{-\rho_1 k_1} e^{-\rho_2 k_2} & \text{para } k_1, k_2 = 1, 2, \dots \end{cases} \quad (5.58)$$

onde $\rho, \rho_1, \rho_2 > 0$.

O kernel de 1a. ordem é equivalente, exceto por um fator proporcional, à resposta ao impulso de um sistema contínuo de primeira ordem estável com função de transferência $\sigma/(s + \sigma)$ amostrada com segurador de ordem zero de período T , onde $\rho = \sigma T$. Sabe-se que o pólo de Laguerre ótimo para o desenvolvimento desse kernel – com relação a (5.40) – é exatamente o pólo do sistema discreto resultante, i.e., $p_1^* = e^{-\rho}$ (Fu e Dumont, 1993).

O kernel de 2a. ordem em (5.58) é uma generalização bidimensional de h_1 . Nesse caso, não é difícil verificar, utilizando as equações (5.42) a (5.46), que

$$Q_{1,2} = \frac{1}{2} \left(\frac{1}{1 - e^{-2\rho_1}} + \frac{1}{1 - e^{-2\rho_2}} \right) \quad (5.59)$$

$$Q_{2,2} = \frac{1}{2} \left(\frac{1 - e^{-\rho_1}}{1 + e^{-\rho_1}} + \frac{1 - e^{-\rho_2}}{1 + e^{-\rho_2}} \right) \quad (5.60)$$

Utilizando as equações (5.59) e (5.60) é possível analisar o comportamento do pólo de Laguerre ótimo em (5.49) para vários valores de ρ_1 e ρ_2 , que especificam completamente as dinâmicas do kernel h_2 . Dado que h_2 é simétrico com relação a ρ_1 e ρ_2 , o pólo de Laguerre ótimo depende desses termos também de forma simétrica. A figura 5.16 ilustra o pólo ótimo como uma função de ρ_1 (ou ρ_2) para alguns valores específicos de ρ_2 (ou ρ_1).

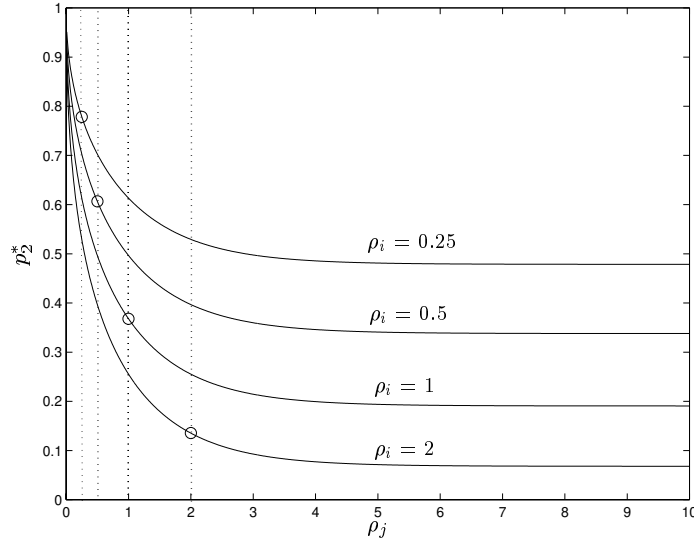


Figura 5.16: Pólo de Laguerre ótimo para o kernel de 2a. ordem como uma função de ρ_j para alguns valores de ρ_i ($i, j = 1, 2$; $i \neq j$).

Os círculos na figura 5.16 destacam os pontos onde $\rho_1 = \rho_2$. Neste caso específico, é simples demonstrar que a equação do pólo ótimo (5.49) torna-se $p_2^* = e^{-\rho_1} = e^{-\rho_2}$ ($\sqrt{4Q_{1,m}Q_{2,m} - Q_{2,m}^2 - 2Q_{2,m}} = 1$). Essa é uma generalização do resultado para o kernel de 1a. ordem discutido no início do exemplo e significa que se $\rho_1, \rho_2 \rightarrow \infty$ então $p_2^* \rightarrow 0$, i.e., o pólo ótimo (discreto) segue as características cada vez mais rápidas do kernel. Pode-se ver também na figura 5.16 que se ρ_1 e/ou ρ_2 tendem a zero então $p_2^* \rightarrow 1$, i.e., o limitante para dinâmicas lentas estáveis. Nesse caso o kernel tende a ser quase constante (i.e., com uma dinâmica muito lenta) na direção k_i quando $\rho_i \rightarrow 0$, como ilustrado na figura 5.17 para $\rho_2 = 0.01$.

Suponha agora que $\rho_1 = 0.5$, $\rho_2 = 0.9$ e que, por razões práticas, o kernel h_2 (figura 5.18) seja truncado de modo que $h_2(k_1, k_2) = 0$ para k_1 ou $k_2 > 50$. Nessas condições, o pólo ótimo para o desenvolvimento desse kernel, calculado computacionalmente através das equações (5.42) até (5.46) e (5.49), é dado por $p_2^* = 0.51349$. Para caracterizar a otimalidade desse pólo no desenvolvimento truncado do kernel utilizando uma quantidade finita de funções, N_2 , varia-se o valor de p_2 dentro do intervalo de factibilidade $] - 1, 1[$ com discretização 0.01. Para cada valor, calculam-se os respectivos coeficientes de Laguerre α_{i_1, i_2} ($i_1, i_2 = 1, \dots, N_2$) através da equação (5.36), lembrando

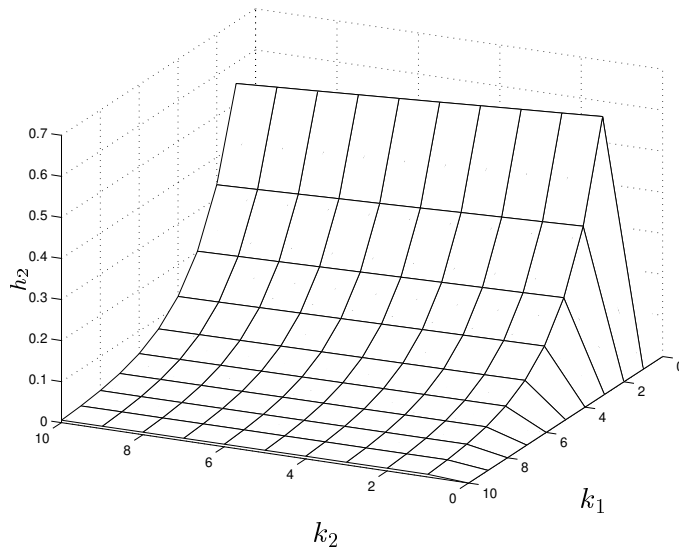


Figura 5.17: Kernel de 2a. ordem com $\rho_1 = 0.5$ e $\rho_2 = 0.01$.

que $h_2(k_1, k_2) = 0$ para k_1 ou $k_2 > 50$, e obtém-se uma estimativa \bar{h}_2 do kernel através da equação (5.35) com as somatórias truncadas em N_2 termos.

O erro absoluto máximo entre o kernel original e o kernel aproximado pelo desenvolvimento de Laguerre truncado com $N_2 = 1, 2$ e 3 funções (norma infinita do erro $E = h_2 - \bar{h}_2$), em função do pólo p_2 , é ilustrado na figura 5.19, onde fica evidenciada a otimalidade da solução p_2^* . Resultado análogo, omitido por simplicidade, é obtido para o kernel h_1 com relação a p_1^* . Esses resultados indicam que a seleção ótima dos pólos permite obter uma dada qualidade de aproximação utilizando um número reduzido de funções de Laguerre, o que significa uma redução de ordem da representação em espaço-de-estados do modelo Wiener-Volterra resultante.

A qualidade do desenvolvimento truncado de Laguerre pode ser avaliada através da figura 5.20, que ilustra o kernel aproximado com $N_2 = 3$ funções e $p_2 = p_2^*$. A superfície de erro entre esse kernel e o kernel original da figura 5.18 é apresentada na figura 5.21.

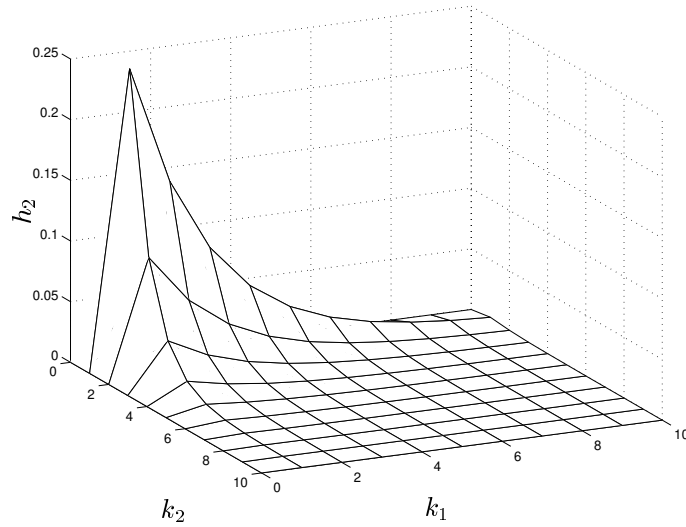


Figura 5.18: Kernel de 2a. ordem com $\rho_1 = 0.5$ e $\rho_2 = 0.9$.

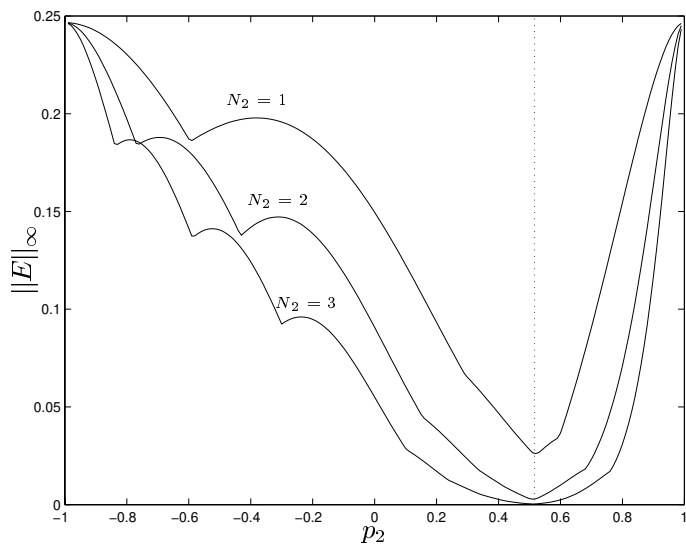


Figura 5.19: Comportamento do erro absoluto máximo entre o kernel original h_2 e o kernel aproximado \bar{h}_2 em função do pólo de Laguerre. A linha pontilhada indica a solução ótima $p_2^* = 0.51349$.

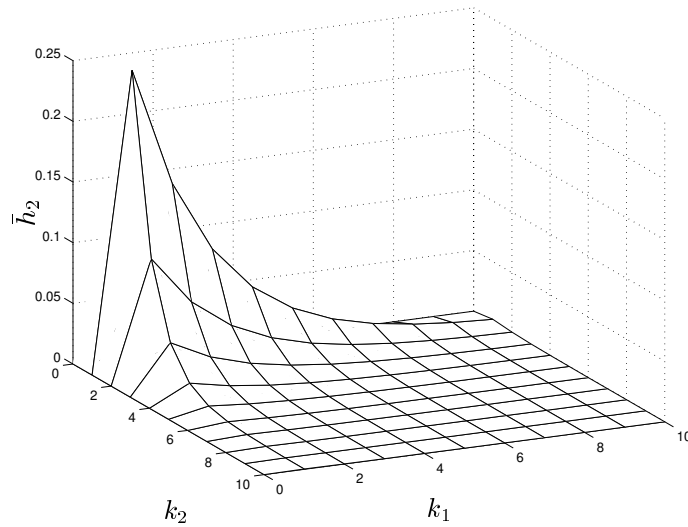


Figura 5.20: Kernel aproximado com $p_2 = 0.51349$ e $N_2 = 3$.

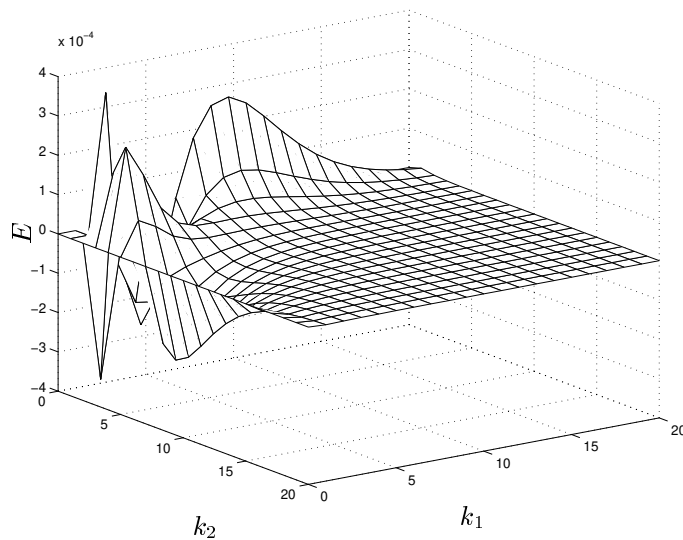


Figura 5.21: Superfície de erro entre os kernels h_2 e \bar{h}_2 .

5.12.5 Desenvolvimento Ótimo dos Kernels com Base de Laguerre Única

Nas seções anteriores tratou-se o problema do desenvolvimento ótimo dos kernels de Volterra utilizando bases de Laguerre independentes otimizadas de forma individual para cada kernel. Outra abordagem possível para a implementação de modelos

Wiener-Volterra é a utilização de uma base de Laguerre única, compartilhada por todos os kernels. Essa abordagem é de particular interesse quando requer-se simplicidade matemática e/ou quando dispõe-se de recursos computacionais limitados, sendo equivalente a fixar $p_1 = \dots = p_\eta \triangleq p$ em (5.39). Nesse caso, o problema de desenvolvimento ótimo dos kernels através dessa base comum pode ser escrito como:

$$\min_{-1 < p < 1} J \triangleq \sum_{m=1}^{\eta} J_m \quad (5.61)$$

onde J é obtido através de (5.41) como:

$$\begin{aligned} J &= \sum_{m=1}^{\eta} \left(\frac{p Q_{2,m} + (1-p)^2 Q_{1,m} + p - p^2}{1-p^2} \right) m \|h_m\|^2 = \\ &= \left(\frac{(\bar{Q}_1 - 1)p^2 + (\bar{Q}_2 - 2\bar{Q}_1 + 1)p + \bar{Q}_1}{1-p^2} \right) \varpi \end{aligned} \quad (5.62)$$

sendo

$$\varpi = \sum_{m=1}^{\eta} m \|h_m\|^2 \quad (5.63)$$

$$\bar{Q}_1 = \frac{1}{\varpi} \sum_{m=1}^{\eta} m Q_{1,m} \|h_m\|^2 \quad (5.64)$$

$$\bar{Q}_2 = \frac{1}{\varpi} \sum_{m=1}^{\eta} m Q_{2,m} \|h_m\|^2 \quad (5.65)$$

Teorema 5.3 *O problema (5.61) possui uma solução global estrita dada por*

$$p^* = \frac{2\bar{Q}_1 - 1 - \bar{Q}_2}{2\bar{Q}_1 - 1 + \sqrt{4\bar{Q}_1\bar{Q}_2 - \bar{Q}_2^2 - 2\bar{Q}_2}} \quad (5.66)$$

se os kernels h_m ($m = 1, \dots, \eta$) satisfizerem as condições do teorema 5.1.

Prova: A prova encontra-se no apêndice D. ■

5.12.6 Estimação Simultânea dos Kernels e dos Pólos de Laguerre Ótimos

Um requisito fundamental da estratégia de desenvolvimento ótimo de modelos de Volterra em funções ortonormais de Laguerre discutida até o presente ponto é o conhecimento *a priori* dos kernels do modelo. Esse requisito pressupõe um procedimento preliminar de estimação dos kernels, e.g., através de alguma das técnicas já discutidas no capítulo 2 (seção 2.4). Assim, o desenvolvimento ótimo dos kernels é realizado *off-line*, o que limita a otimalidade dos modelos Wiener-Volterra resultantes à representação de sistemas invariantes ou cujas dinâmicas dominantes não sofrem variações significativas ao longo do tempo¹³.

Visando contornar as dificuldades mencionadas acima, propõe-se a seguir um algoritmo para estimar simultaneamente, de forma iterativa a partir de um conjunto de dados de E/S de um determinado sistema, os kernels de um modelo de Volterra desse sistema e a solução ótima para o desenvolvimento de Laguerre desses kernels. Antes, porém, é importante mencionar que a idéia do algoritmo é intuitiva, sendo que o estudo de condições formais para a convergência do mesmo é parte do conjunto de perspectivas futuras do presente trabalho.

Algoritmo

1. Selecionar um limitante ϵ para o truncamento dos kernels estimados, i.e., um valor máximo para k_1, \dots, k_m além do qual os kernels $\bar{h}_m(k_1, \dots, k_m)$, onde $m = 1, \dots, \eta$, são considerados nulos. Na prática, ϵ pode ser fixado em função do tempo de estabelecimento do sistema.
2. Selecionar o truncamento do desenvolvimento de Laguerre de cada kernel h_m em uma quantidade N_m de funções, sendo $m = 1, \dots, \eta$.
3. Selecionar valores iniciais para os pólos de Laguerre (ou pólo, no caso de uma base única).
4. Estimar, utilizando o conjunto de dados disponíveis e o algoritmo dos Mínimos Quadrados, os coeficientes de Laguerre $\alpha_{(\cdot)}$ do modelo (5.37) com as somatórias referentes ao termo de ordem m truncadas em N_m elementos.
5. Estimar os kernels de Volterra utilizando a equação (5.35) como somas truncadas de N_m elementos.
6. Calcular os novos pólos de Laguerre utilizando a solução ótima dada pela equação (5.49), ou pela equação (5.66) no caso de um pólo único, e retornar ao passo 4.

¹³Nesse último caso é possível realizar uma compensação *on-line* das variações adaptando-se apenas os coeficientes de Laguerre $\alpha_{(\cdot)}$ na equação (5.37), por exemplo através do algoritmo dos mínimos quadrados recursivos (RLS) (Ljung, 1999).

Observação 5.6 *Após uma execução preliminar off-line do procedimento acima, visando obter estimativas iniciais dos kernels e do(s) respectivo(s) pólo(s) ótimo(s), pode-se implementar o algoritmo de forma adaptativa através dos mínimos quadrados recursivos (RLS) de forma que cada iteração seja executada a cada novo período de amostragem, utilizando o novo conjunto de amostras de E/S correspondente.*

Observação 5.7 *Para efeito da estimação via mínimos quadrados, os coeficientes de Laguerre dos kernels de ordem superior a 1 podem ser considerados simétricos sem qualquer prejuízo para a capacidade de representação do modelo (Dumont e Fu, 1993), conforme foi discutido no capítulo 2. Essa hipótese, que implica arbitrar a igualdade entre os elementos $\alpha_{(\cdot)}$ para qualquer permutação dos índices i_1, \dots, i_m em (5.37), permite reduzir sensivelmente a quantidade de coeficientes a serem estimados, especialmente para valores N_1, \dots, N_m elevados.*

5.12.7 Resultados de Simulação

Considera-se a seguir o modelo de Volterra utilizado na seção 5.12.4, com kernels de 1a. e 2a. ordem dados pelas equações (5.57) e (5.58), respectivamente. Seleciona-se como exemplo os valores $\rho = 0.2$, $\rho_1 = 0.5$ e $\rho_2 = 0.9$. Para fins de implementação computacional, assume-se que os kernels são truncados (considerados nulos) para k_1 ou $k_2 > 50$, garantindo assim o cálculo finito da saída $y(k)$ do modelo em (5.34) bem como dos termos $\|h_1\|$, $\|h_2\|$, $Q_{1,1}$, $Q_{2,1}$, $Q_{1,2}$ e $Q_{2,2}$ nas equações (5.42) a (5.45). Nessas condições, os pólos ótimos para o desenvolvimento de cada um dos kernels através de bases de Laguerre independentes são dados por $p_1^* = 0.81873$ e $p_2^* = 0.51349$, conforme discutido na seção 5.12.3. Para uma base única, tem-se o pólo ótimo calculado através de (5.66) como $p^* = 0.79376$. Esse valor é próximo ao valor ótimo p_1^* para o desenvolvimento independente do kernel de 1a. ordem, o que indica uma predominância da dinâmica desse kernel no cálculo de p^* . De fato, h_1 predomina sobre h_2 nesse caso no sentido que $\|h_1\|^2 = 2.0332$, enquanto $\|h_2\|^2 = 0.1152$.

1º Caso de Estudo

A seguir avalia-se o desempenho do algoritmo iterativo da seção 5.12.6. Assume-se como sistema a ser modelado, por um modelo Wiener-Volterra com base de Laguerre, o próprio modelo de Volterra descrito acima. Os dados de E/S são gerados através da equação (5.34) com uma seqüência de entrada u aleatória com 200 amostras uniformemente distribuídas em $[-1, 1]$, sendo 100 amostras a serem utilizadas pelo algoritmo para a estimação do modelo e as demais para sua validação. Seleciona-se $\epsilon = 100$ (superior aos truncamentos dos kernels originais) e $N_1 = N_2 = 3$, que é uma quantidade de funções suficiente para uma representação precisa do sistema em questão, conforme visto nas figuras 5.18 a 5.21.

A figura 5.22 ilustra a evolução do pólo de Laguerre para três execuções do algoritmo iterativo utilizando um modelo com uma base única de funções comum a ambos os

kernels. Cada execução utiliza uma realização distinta do conjunto de dados de E/S. Em todos os casos inicia-se o pólo em $p = 0$, valor abaixo do qual não existe dinâmica contínua equivalente. Nota-se que nas três execuções o pólo converge praticamente para o valor ótimo, sendo que no pior caso o pólo final obtido é $p = 0.79050$. A evolução do erro quadrático médio entre os dados de validação e a série sintética do modelo Wiener-Volterra resultante a cada iteração da execução intermediária do algoritmo é ilustrada na figura 5.23. A simulação para o modelo final correspondente é ilustrada na figura 5.24, que mostra uma aproximação precisa em termos do sinal de saída.

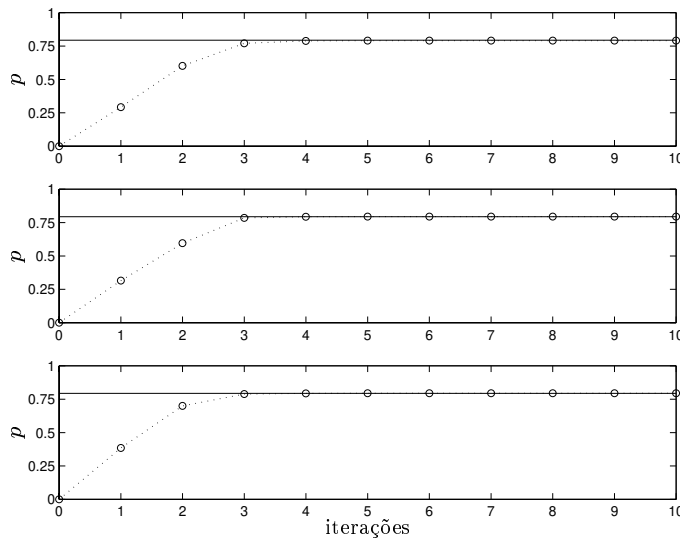


Figura 5.22: Evolução do pólo de Laguerre ao longo do algoritmo iterativo para três conjuntos de dados distintos. As linhas contínuas indicam a solução ótima $p^* = 0.79376$.

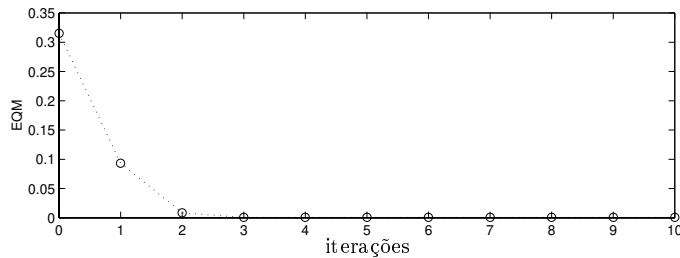


Figura 5.23: Evolução do erro quadrático médio entre os dados de validação e a série sintética do modelo para cada iteração do algoritmo: Base de Laguerre única (EQM final = 8.4294×10^{-4}).

2º Caso de Estudo

Considera-se agora um modelo com bases de Laguerre independentes para cada kernel. Tem-se na figura 5.25 a evolução dos respectivos pólos ao longo de uma execução do algoritmo iterativo. Nota-se que ambos os pólos convergem praticamente para os valores ótimos correspondentes, sendo os valores finais dados por $p_1 = 0.81834$ e $p_2 = 0.51390$. Simulações com conjuntos de dados distintos apresentaram resultados análogos.

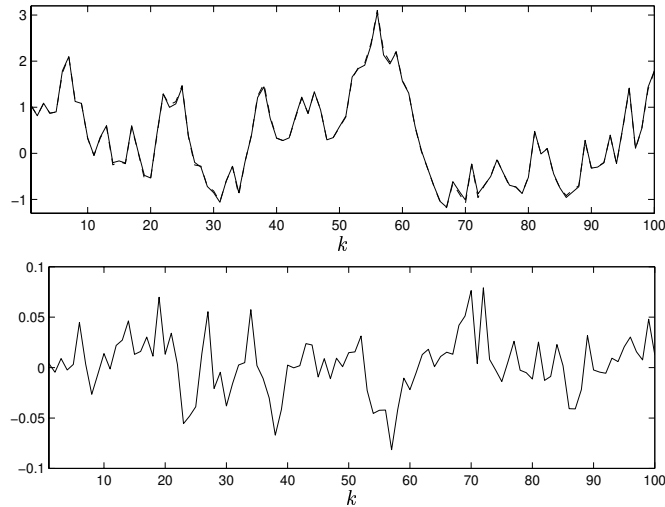


Figura 5.24: Acima: Dados de validação e série sintética do modelo final com base de Laguerre única (curvas sobrepostas); Abaixo: Erro de previsão.

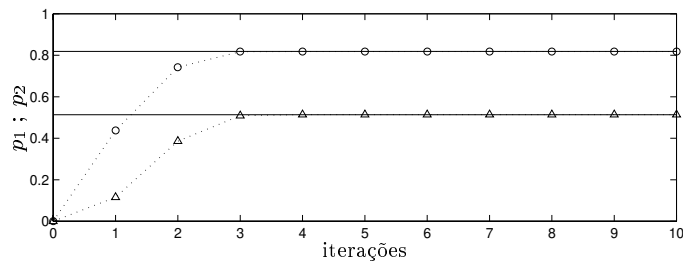


Figura 5.25: Evolução dos pólos p_1 (círculos) e p_2 (triângulos) de Laguerre ao longo do algoritmo. As linhas contínuas indicam as soluções ótimas $p_1^* = 0.81873$ e $p_2^* = 0.51349$.

A evolução do EQM entre os dados de validação e a série sintética do modelo Wiener-Volterra resultante a cada iteração do algoritmo é ilustrada na figura 5.26. A simulação para o modelo final correspondente é ilustrada na figura 5.27 (mesmos dados de validação da figura 5.24). Conforme esperado, a precisão do modelo com bases de Laguerre distintas é superior àquela apresentada pelo modelo com uma base única. É importante notar que essa precisão é obtida por um modelo com apenas 9 coeficientes de Laguerre, dada a hipótese de simetria discutida na observação 5.7.

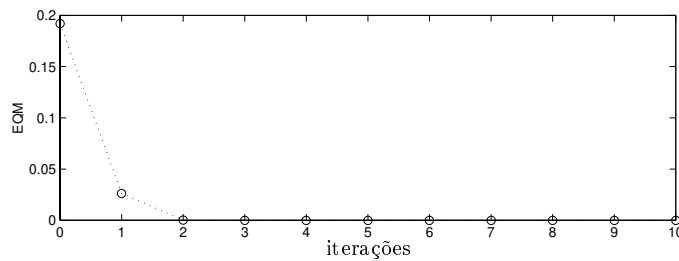


Figura 5.26: Evolução do erro quadrático médio entre os dados de validação e a série sintética do modelo para cada iteração do algoritmo: Bases de Laguerre distintas (EQM final = 2.3826×10^{-5}).

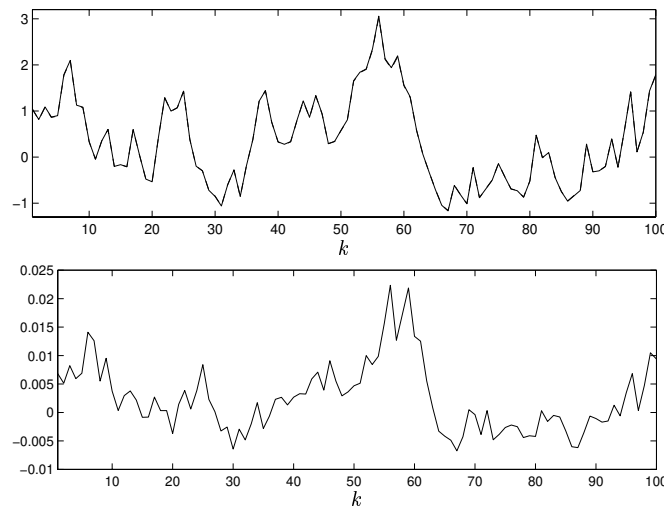


Figura 5.27: Acima: Dados de validação e série sintética do modelo final com bases de Laguerre distintas (curvas sobrepostas); Abaixo: Erro de previsão.

5.13 Resumo e Discussões Complementares

Na maioria das estratégias desenvolvidas para a identificação de sistemas dinâmicos não lineares a determinação de um vetor de regressão adequado contendo amostras passadas dos sinais de entrada e saída do sistema a ser modelado constitui um problema fundamental. Esse é um caso particular de um problema mais geral de seleção de variáveis de entrada que não ocorre apenas no projeto de modelos dinâmicos. No contexto de modelagem *fuzzy*, por exemplo, que constitui um dos assuntos centrais do presente trabalho, a seleção de variáveis de entrada relevantes dentre um conjunto de candidatas tem sido por décadas abordado através de procedimentos de tentativa-e-erro (Pedrycz, 1984a; Lee et al., 1994), métodos heurísticos (Tong, 1978; Takagi e Sugeno, 1985; Sugeno e Yasukawa, 1993) ou, mais recentemente, técnicas de compu-

tação evolutiva (Shimojima et al., 1995). Essas estratégias, no entanto, são geralmente trabalhosas e/ou computacionalmente complexas.

A seleção de amostras passadas relevantes dos sinais de um sistema a ser identificado não é o único problema associado às metodologias usuais para a construção de modelos dinâmicos não lineares. Entre outros, incluem-se problemas estruturais como realimentação de erros de previsão e número excessivo de termos de entrada. Uma estratégia efetiva para contornar esses e outros problemas no contexto de modelagem de sistemas dinâmicos estáveis é a utilização de modelos baseados em funções de base ortonormal (FBO), como as funções de Laguerre e Kautz.

No presente capítulo, os modelos FBO foram analisados utilizando uma abordagem matemática original que resulta em uma representação genérica de espaço-de-estados do tipo Wiener. Foram propostas realizações particulares dessa representação através do uso de sistemas *fuzzy*. Os modelos *fuzzy* FBO propostos foram descritos utilizando uma formulação unificada que também abrange outras classes de modelos FBO, como os modelos lineares e os modelos Wiener-Volterra. Essa formulação permite a interpretação das diferenças entre as diversas classes de modelos FBO analisadas, diferenças essas avaliadas neste capítulo através de estudos teóricos e experimentais. Mostrou-se que os modelos FBO *fuzzy* representam, em determinadas aplicações, um melhor compromisso entre número de parâmetros e precisão quando comparados aos modelos Wiener-Volterra. Esse resultado deve-se à flexibilidade estrutural das arquiteturas utilizadas, i.e., modelos *fuzzy* dos tipos relacional simplificado, equivalente a redes neurais RBF, e Takagi-Sugeno. De fato, mostrou-se que os modelos *fuzzy* com estrutura FBO são capazes de aproximar arbitrariamente bem uma ampla classe de sistemas dinâmicos não lineares.

Um aspecto fundamental para o projeto adequado de um modelo com estrutura FBO é a interpretação matemática do modelo em questão. Os modelos FBO lineares, por exemplo, resultam do desenvolvimento da resposta ao impulso do modelo linear de convolução em uma base de funções ortonormais. Similarmente, os modelos Wiener-Volterra representam o desenvolvimento dos kernels de Volterra nesse mesmo tipo de base. Em princípio, os modelos *fuzzy* FBO recaem em um problema análogo àquele verificado em trabalhos anteriores onde foram desenvolvidas redes neurais do tipo MLP segundo essa mesma estrutura (Back e Tsoi, 1996; Sentoni et al., 1996; Sentoni et al., 1998; Balestrino et al., 1999; Alataris et al., 2000; Vázquez e Agamennoni, 2001; Arto et al., 2001): a ausência de interpretabilidade matemática. Esse problema foi abordado neste capítulo a partir da proposta do Modelo FBO *Fuzzy* Generalizado. Esse modelo consiste de uma interpolação do tipo Takagi-Sugeno de diversos modelos locais (e.g. lineares) com estrutura FBO. Quando adota-se uma dinâmica de estados comum aos diferentes modelos locais, obtém-se a estrutura convencional do tipo Wiener proposta originalmente para os modelos FBO *fuzzy*. Esse resultado permite a interpretação desses modelos como realizações particulares da representação generalizada, matematicamente clara.

O capítulo também analisou modelos FBO multivariáveis. Nesse contexto, propôs-se

o desenvolvimento dos modelos hierárquicos discutidos no capítulo 4 também segundo a estrutura FBO. Conforme discutido naquele capítulo, a arquitetura hierárquica permite uma redução sensível na quantidade de parâmetros do modelo. Por outro lado, a otimização desses parâmetros torna-se não convexa, exigindo a aplicação de algoritmos robustos a problemas de convergência local. Por esta razão, propôs-se a utilização de uma estratégia híbrida para a estimação dos parâmetros dos modelo FBO hierárquicos. Essa estratégia consiste da utilização de um algoritmo genético para a obtenção de um modelo inicial aproximado a ser refinado por um algoritmo baseado em gradiente, tendo sido aplicada com sucesso à modelagem de um processo biotecnológico para produção de álcool etílico (etanol) em escala industrial.

Finalmente, o capítulo discutiu estratégias para o projeto das bases de funções ortonormais utilizadas na implementação computacional dos modelos FBO. O principal resultado nesse contexto consiste da obtenção de uma solução analítica global para o desenvolvimento ótimo de kernels de Volterra em funções ortonormais de Laguerre. Essa solução minimiza, segundo um critério específico, o erro de aproximação associado ao desenvolvimento truncado dos kernels em um número finito de funções, o que permite uma redução de ordem da representação em espaço-de-estados do modelo Wiener-Volterra resultante. A solução ótima foi obtida tanto para modelos com bases de Laguerre independentes para o desenvolvimento de cada kernel como para os modelos usuais em que se utiliza uma base comum ao desenvolvimento compartilhado de todos os kernels. Propôs-se ainda um algoritmo iterativo para a estimação simultânea dos kernels de Volterra e da solução ótima para o desenvolvimento desses kernels em funções de Laguerre utilizando dados de E/S de um sistema a ser modelado.

5.14 Contribuições e Perspectivas

As principais contribuições desse capítulo são:

1. Proposta e desenvolvimento de diferentes tipos de modelos *fuzzy* com estrutura FBO.
2. Estudo da capacidade de representação dos modelos *fuzzy* FBO propostos.
3. Interpretação matemática dos modelos *fuzzy* FBO propostos como realizações específicas de uma representação generalizada dada pela interpolação de diferentes modelos FBO locais.
4. Estudos comparativos entre os modelos *fuzzy* FBO propostos e outras classes de modelos FBO a partir de uma formulação unificada que permite a interpretação das diferenças entre essas diversas classes, diferenças essas avaliadas detalhadamente a partir de argumentos teóricos e experimentos computacionais.
5. Proposta e desenvolvimento de diferentes arquiteturas de modelos *fuzzy* FBO hierárquicos.

6. Estudo de estratégias para o projeto das bases de funções ortonormais utilizadas na implementação computacional de modelos FBO.
7. Obtenção de uma solução analítica global para o desenvolvimento ótimo de kernels de Volterra em funções ortonormais de Laguerre tanto para modelos com bases de Laguerre independentes para o desenvolvimento de cada kernel como para os modelos usuais onde utiliza-se uma base comum ao desenvolvimento compartilhado de todos os kernels.
8. Desenvolvimento de um algoritmo iterativo para a estimação simultânea dos kernels de Volterra e da solução ótima para o desenvolvimento desses kernels em funções de Laguerre utilizando dados de E/S de um sistema a ser modelado.

As principais perspectivas a partir dos resultados obtidos e apresentados no presente capítulo são:

1. Desenvolvimento de soluções ótimas para a seleção de bases de funções ortonormais em modelos FBO *fuzzy*.
2. Desenvolvimento de procedimentos sistemáticos para a estimação simultânea e adaptativa das bases de funções ortonormais e dos demais parâmetros dos modelos FBO, possivelmente através de estratégias de treinamento de redes neurais recorrentes (Medsker e Jain, 1999).
3. Aprimoramento do algoritmo híbrido para a estimação de modelos FBO não lineares nos parâmetros, como os modelos FBO hierárquicos, através da incorporação de estratégias mais elaboradas ao algoritmo genético, tais como: funções de *fitness* alternativas (e.g. logarítmica ou por *rank*), medidas de similaridade alternativas (e.g. com base nos valores de *fitness*), diversidade probabilística, seleção por torneio (competição), *crossover* de múltiplos pontos, entre outros.
4. Investigação sobre as condições de convergência do algoritmo iterativo para a estimação simultânea dos kernels de Volterra e da solução ótima para o desenvolvimento desses kernels em funções de Laguerre.
5. Extensão de todos os resultados decorrentes da hipótese de utilização específica de bases de Laguerre também no que se refere a outras bases ortonormais, tais como as bases de Kautz (Broome, 1965; Wahlberg, 1991a) e as bases generalizadas (Ninness e Gustafsson, 1995; Van den Hof et al., 1995).

Capítulo 6

Aplicações em Controle de Processos Não Lineares

Esse capítulo aborda o problema de controle automático de sistemas não lineares utilizando estratégias de controle baseadas em modelos dos processos a serem controlados. As discussões concentram-se nos modelos *fuzzy* com estrutura FBO desenvolvidos no capítulo 5, por duas razões: Esses modelos constituem uma das contribuições principais do presente trabalho e foram aqueles que apresentaram os melhores resultados em aplicações de modelagem de sistemas dinâmicos. Desenvolve-se uma estratégia de controle preditivo baseada na linearização de modelos *fuzzy* FBO em cada instante de amostragem e aplica-se essa estratégia ao controle de dois processos químicos: Um reator de polimerização e uma planta biotecnológica para produção de etanol.

6.1 Introdução

A utilização dos conceitos de lógica *fuzzy* para o projeto de sistemas de controle automático de processos foi realizada inicialmente por Mamdani (Mamdani, 1974; Mamdani, 1977; King e Mamdani, 1977), tendo desde então sido amplamente difundida. Atualmente, existem várias arquiteturas possíveis para a implementação de malhas de controle a partir de sistemas *fuzzy* (Pedrycz, 1993; Yager e Filev, 1994; Kosko, 1997; Passino e Yurkovich, 1997). Inicialmente, essas arquiteturas podem ser classificadas em duas vertentes principais (Harris et al., 1993): Controle Direto e Indireto. No controle direto, o próprio controlador é constituído de um sistema *fuzzy* (Tanscheit, 1992; Gomide e Gudwin, 1994; Pinheiro, 2000). Nesse caso, o respectivo conjunto de regras descrevendo as possíveis ações de controle pode ser projetado qualitativamente por um especialista (e.g. um operador do processo) (Lee, 1990), obtido em malha fechada de forma adaptativa a partir de uma parametrização das regras (Wang, 1994) ou desenvolvido através de uma combinação dessas duas estratégias. No controle indireto, que é de particular interesse no presente contexto, a lei de controle depende de um modelo *fuzzy* do processo a ser controlado.

O controle *fuzzy* indireto tem sido objeto de extensa investigação na literatura, principalmente devido ao potencial que tem apresentado para a solução de problemas complexos reais e acadêmicos. Nessa abordagem, de caráter amplo e genérico, destacam-se dois grupos importantes. Um desses grupos caracteriza-se pela utilização de um modelo *fuzzy* do processo como base para o projeto paramétrico de um controlador, como em (Teixeira e Žak, 1999; Teixeira et al., 2000). No segundo grupo, a lei de controle é calculada em cada instante de amostragem em função de algum critério envolvendo as equações e/ou previsões de saída de um modelo, que pode ser adaptado (ajustado e/ou linearizado) em tempo real. Nesse caso, uma possibilidade é utilizar uma lei de controle dada pela minimização de uma função de custo que considere explicitamente o desempenho desejado para a malha fechada como uma função do modelo *fuzzy* não linear disponível do processo (Oliveira e Lemos, 1995; Oliveira e Lemos, 1998; Nazzetta et al., 1998; Sing e Postlethwaite, 1997; Kelkar e Postlethwaite, 1998; Meleiro, 2002; Meleiro et al., nd). Essa estratégia, porém, requer a solução *online* de problemas em geral não convexos de programação não linear. A solução desses problemas pode ser crítica em muitas aplicações, especialmente para processos e controladores multivariáveis. Possíveis abordagens nesse contexto baseiam-se na utilização de métodos de gradiente, como em (Oliveira e Lemos, 1995; Oliveira e Lemos, 1998; Nazzetta et al., 1998), busca unidimensional, como em (Sing e Postlethwaite, 1997; Kelkar e Postlethwaite, 1998), e Programação Quadrática Seqüencial (*Sequential Quadratic Programming* – SQP), como em (Meleiro, 2002; Meleiro et al., nd). Outra alternativa, que pode ser mais eficaz (cf. (Cavalcante Júnior, 2000)), é a utilização de controladores ótimos baseados em algum tipo de linearização do modelo *fuzzy* do processo em cada ponto de operação (e.g. ver (Fischer et al., 1998b; Fischer et al., 1998a; Cavalcante Júnior, 2000; Cavalcante Júnior e Amaral, 2000; Campello et al., nd)). A linearização dos modelos faz com que os respectivos problemas de controle ótimo possam ser escritos através de formulações convexas, o que permite a obtenção de soluções globais com esforço computacional reduzido. Por esta razão, discute-se na seção seguinte uma estratégia de controle que se enquadra nessa categoria.

6.2 Controle Preditivo com Base na Linearização de Modelos

Conforme discutido em capítulos anteriores, o controle automático de sistemas dinâmicos é apenas uma das diversas áreas de aplicação para o conjunto de modelos investigados no presente trabalho, ou seja, modelos *fuzzy*, neurais e de Volterra. É uma aplicação em potencial, no entanto, para os modelos FBO, cuja estrutura é específica e particularmente favorável para a representação de sistemas dinâmicos. Por esta razão, esses modelos serão adotados nos desenvolvimentos subseqüentes. Em particular, serão considerados os modelos *fuzzy* com estrutura FBO que constituem uma proposta original cujos resultados têm se mostrado promissores.

Os controladores preditivos (Soeterboek, 1992; Clarke, 1994; Camacho e Bordons, 1999) têm sido amplamente difundidos na área de controle de processos químicos, onde concentram-se as aplicações consideradas neste capítulo. A estratégia de controle proposta aqui consiste da utilização do Controlador Preditivo Generalizado (*Generalized Predictive Controller* – GPC) (Clarke et al., 1987a; Clarke et al., 1987b) associado a um procedimento de linearização, em cada instante de amostragem, do modelo *fuzzy* FBO do processo a ser controlado (Campello e Amaral, 2002c). O procedimento de linearização é coerente com os fundamentos matemáticos do controlador GPC, que utiliza explicitamente as equações de um modelo linear do processo para o desenvolvimento da sua lei de controle ótima. Especificamente, adota-se a versão incremental (variacional) do controlador com a incorporação explícita de restrições operacionais nas variáveis de controle. O respectivo algoritmo é descrito em detalhes no apêndice E. A linearização dos modelos *fuzzy* FBO será discutida na seção seguinte.

6.3 Linearização de Modelos Fuzzy FBO

Os modelos *fuzzy* desenvolvidos segundo a estrutura FBO no capítulo 5 são os modelos relacionais simplificados e os modelos Takagi-Sugeno (TS), sendo que ambos podem ser escritos como¹ (ver seção 5.3):

$$\mathbf{l}(k+1) = \mathbf{A}\mathbf{l}(k) + \mathbf{b}u(k) \quad (6.1)$$

$$\hat{y}(k) = \mathcal{H}(\mathbf{l}(k)) = \boldsymbol{\lambda}(\mathbf{l}(k))^T \boldsymbol{\zeta} \quad (6.2)$$

onde $u(k)$ é a entrada, $\hat{y}(k)$ é a saída, $\mathbf{l}(k) = [l_1(k) \cdots l_n(k)]^T$ é o vetor de estados ortogonais (de Laguerre no presente contexto) e $\boldsymbol{\zeta} = [\zeta_1 \cdots \zeta_\mu]^T$ é um vetor de parâmetros. O vetor $\boldsymbol{\lambda}$ depende do tipo de modelo utilizado. Conforme discutido na seção 5.3, os modelos TS são uma generalização dos modelos relacionais simplificados. Por esta razão, esses modelos serão considerados nos desenvolvimentos subseqüentes, onde a variável de tempo discreto k será omitida para simplificar a notação. Conforme demonstrado na seção 5.3, se os conseqüentes das regras TS forem implementados como funções afins das variáveis da premissa, tem-se que $\boldsymbol{\lambda}$ é dado por:

$$\boldsymbol{\lambda}(\mathbf{l}) = \gamma(\mathbf{l}) [w_1(\mathbf{l}) \quad w_1(\mathbf{l})l_1 \cdots w_1(\mathbf{l})l_n \cdots w_M(\mathbf{l}) \quad w_M(\mathbf{l})l_1 \cdots w_M(\mathbf{l})l_n]^T \quad (6.3)$$

onde $\gamma(\mathbf{l})$ é um termo de normalização dado por:

$$\gamma(\mathbf{l}) = \frac{1}{\sum_{i=1}^M w_i(\mathbf{l})} \quad (6.4)$$

¹Por simplicidade e sem qualquer perda de generalidade, considera-se o caso monovariável (SISO).

e $w_i(\mathbf{l})$, para $i = 1, \dots, M$, são funções n -dimensionais que dependem dos conjuntos *fuzzy* do modelo. Para conjuntos *fuzzy* Gaussianos, essas funções são descritas como:

$$w_i(\mathbf{l}) = \prod_{h=1}^n \exp\left(-\frac{(l_h - \theta_{i_h})^2}{2\sigma_{i_h}^2}\right) = \exp\left(-\frac{1}{2}(\mathbf{l} - \boldsymbol{\theta}_i)^T \Lambda_i^{-1} (\mathbf{l} - \boldsymbol{\theta}_i)\right) \quad (6.5)$$

onde $\boldsymbol{\theta}_i = [\theta_{i_1} \dots \theta_{i_n}]^T$ é o vetor com as coordenadas do centro da i -ésima função e $\Lambda_i = \text{diag}(\sigma_{i_1}^2, \dots, \sigma_{i_n}^2)$ é a matriz cujos elementos diagonais representam a variância dessa função em cada uma das n direções.

Como a dinâmica de estados FBO em (6.1) é linear, deve-se linearizar apenas o mapeamento estático do modelo descrito pelas equações (6.2) a (6.5). A saída do modelo linearizada em um determinado estado \mathbf{l}_0 é obtida utilizando um desenvolvimento de Taylor de primeira ordem, como segue:

$$\hat{y}' = \mathcal{H}'(\mathbf{l}, \mathbf{l}_0) = \mathcal{H}(\mathbf{l}_0) + (\mathbf{l} - \mathbf{l}_0)^T \nabla_1 \mathcal{H}(\mathbf{l}_0) \quad (6.6)$$

onde $\nabla_1 \mathcal{H}(\mathbf{l})$ é o gradiente do mapeamento *fuzzy* \mathcal{H} em (6.2) com relação ao vetor de estados \mathbf{l} , cujo j -ésimo elemento é calculado como:

$$\frac{\partial \mathcal{H}(\mathbf{l})}{\partial l_j} = \boldsymbol{\zeta}^T \frac{\partial \boldsymbol{\lambda}(\mathbf{l})}{\partial l_j} \quad (6.7)$$

sendo o termo $\partial \boldsymbol{\lambda}(\mathbf{l})/\partial l_j$ calculado a partir de (6.3) como:

$$\frac{\partial \boldsymbol{\lambda}(\mathbf{l})}{\partial l_j} = \left[\begin{array}{cccc} \frac{\partial(\gamma(\mathbf{l}) w_1(\mathbf{l}))}{\partial l_j} & \frac{\partial(\gamma(\mathbf{l}) w_1(\mathbf{l}) l_1)}{\partial l_j} & \dots & \frac{\partial(\gamma(\mathbf{l}) w_1(\mathbf{l}) l_n)}{\partial l_j} & \dots \\ \dots & \frac{\partial(\gamma(\mathbf{l}) w_M(\mathbf{l}))}{\partial l_j} & \frac{\partial(\gamma(\mathbf{l}) w_M(\mathbf{l}) l_1)}{\partial l_j} & \dots & \frac{\partial(\gamma(\mathbf{l}) w_M(\mathbf{l}) l_n)}{\partial l_j} \end{array} \right]^T \quad (6.8)$$

Os termos derivativos acima, por sua vez, são dados por:

$$\frac{\partial(\gamma(\mathbf{l}) w_i(\mathbf{l}) l_h)}{\partial l_j} = \frac{\partial(\gamma(\mathbf{l}) w_i(\mathbf{l}))}{\partial l_j} l_h + \frac{\partial l_h}{\partial l_j} \gamma(\mathbf{l}) w_i(\mathbf{l}) \quad (6.9)$$

$$\frac{\partial(\gamma(\mathbf{l}) w_i(\mathbf{l}))}{\partial l_j} = \frac{\partial \gamma(\mathbf{l})}{\partial l_j} w_i(\mathbf{l}) + \frac{\partial w_i(\mathbf{l})}{\partial l_j} \gamma(\mathbf{l}) \quad (6.10)$$

onde

$$\frac{\partial l_h}{\partial l_j} = \begin{cases} 1, & \text{se } h = j \\ 0, & \text{se } h \neq j \end{cases} \quad (6.11)$$

$$\frac{\partial \gamma(\mathbf{l})}{\partial l_j} = -\frac{1}{\gamma(\mathbf{l})^2} \sum_{i=1}^M \frac{\partial w_i(\mathbf{l})}{\partial l_j} \quad (6.12)$$

$$\frac{\partial w_i(\mathbf{l})}{\partial l_j} = -\frac{(l_j - \theta_{ij})}{\sigma_{ij}^2} w_i(\mathbf{l}) \quad (6.13)$$

sendo os termos em (6.12) e (6.13) calculados através de (6.4) e (6.5), respectivamente.

Observação 6.1 *A extensão do equacionamento acima para o caso multivariável é direta. Para modelos MISO basta supor que o vetor de estados \mathbf{l} contém os estados referentes a todas as entradas, o que mantém inalteradas as equações para a linearização desses modelos. No caso MIMO, como as saídas são naturalmente desacopladas, pode-se linearizar cada saída de forma independente.*

Observação 6.2 *A saída do modelo em (6.6) não é de fato linear, mas afim com relação aos estados, dado que os termos \mathbf{l}_0 , $\mathcal{H}(\mathbf{l}_0)$ e $\nabla_1 \mathcal{H}(\mathbf{l}_0)$ são constantes. O controlador, porém, requer um modelo linear para o cálculo da lei de controle ótima, conforme foi discutido na seção 6.2. Esse problema pode ser facilmente contornado através da definição de uma saída auxiliar dada por $\bar{y}' \triangleq \hat{y}' - \mathcal{H}(\mathbf{l}_0) + \mathbf{l}_0^T \nabla_1 \mathcal{H}(\mathbf{l}_0) = \mathbf{l}^T \nabla_1 \mathcal{H}(\mathbf{l}_0)$, que é claramente linear com relação ao vetor de estados \mathbf{l} . Nesse caso, o cálculo do erro $e = w - \hat{y}'$ entre a referência de controle w e a saída prevista do modelo \hat{y}' pode ser rescrito como $e = w - \mathcal{H}(\mathbf{l}_0) + \mathbf{l}_0^T \nabla_1 \mathcal{H}(\mathbf{l}_0) - \mathbf{l}^T \nabla_1 \mathcal{H}(\mathbf{l}_0) = w' - \bar{y}'$, onde $w' \triangleq w - \mathcal{H}(\mathbf{l}_0) + \mathbf{l}_0^T \nabla_1 \mathcal{H}(\mathbf{l}_0)$ é um sinal de referência modificado (artificial).*

6.4 Controle Preditivo de um Processo de Polimerização

Considera-se a seguir o controle de um Reator de Polimerização do tipo CSTR (*Continuous Stirred Tank Reactor*). Um modelo de estados do processo é dado por:

$$\begin{aligned}
\dot{x}_1(t) &= 60 - x_1(t) \left(2.4568\sqrt{x_2(t)} + 10 \right) \\
\dot{x}_2(t) &= 80u(t) - 10.1022x_2(t) \\
\dot{x}_3(t) &= 0.002412x_1(t)\sqrt{x_2(t)} + 0.11218x_2(t) - 10x_3(t) \\
\dot{x}_4(t) &= 245.9811x_1(t)\sqrt{x_2(t)} - 10x_4(t) \\
y(t) &= x_4(t)/x_3(t)
\end{aligned} \tag{6.14}$$

onde a saída $y(t)$ é o número médio do peso molecular (*Number Average Molecular Weight – NAMW*) [kg/kmol] do polímero resultante da reação que é controlada pela manipulação da taxa de fluxo $u(t)$ [m³/h] da substância iniciadora. Mais detalhes sobre esse processo são apresentados em (Maner et al., 1996).

6.4.1 Identificação do Reator CSTR

A seguir descreve-se o procedimento de estimação de um modelo FBO-TS do reator de polimerização. Esse modelo será utilizado no esquema de controle preditivo descrito nas seções anteriores. A estimação do modelo requer a obtenção de um conjunto de dados de E/S representativo das dinâmicas do processo. Para tanto, simula-se o conjunto de equações diferenciais (6.14) utilizando o método de Runge-Kutta de quarta ordem com passo fixo igual a 0.03h (1.8 min) e condições iniciais dadas por $x_1(0) = 5.506774$, $x_2(0) = 0.132906$, $x_3(0) = 0.0019752$ e $x_4(0) = 49.38182$ (Maner et al., 1996). A simulação é executada de $t = 0$ até $t = 24$ h com a entrada u sendo dada por uma seqüência de degraus com período de 1h e amplitude aleatória uniformemente distribuída dentro do intervalo operacional [0.002, 0.02]. O conjunto de dados de E/S resultante é dividido em duas partes: A primeira metade (primeiras 12 horas) é destinada à estimação do modelo FBO-TS do processo e a outra metade é destinada à validação desse modelo. Esses dados são normalizados no intervalo $[-1, 1]$ para evitar a ocorrência de problemas numéricos durante a fase de estimação dos parâmetros do modelo. A primeira 1/2 hora de cada conjunto é utilizada apenas para calcular os estados de Laguerre corretos que são considerados desconhecidos (e configurados nulos) no início das simulações porque, na prática, a história passada do sinal de entrada com relação aos dados disponíveis é usualmente desconhecida. Esse procedimento também faz com que o transitório associado às condições iniciais do processo não exerça influência sobre a estimação dos parâmetros do modelo.

Os parâmetros de projeto do modelo FBO-TS descrito pelas equações (6.1) a (6.5) são: O número de estados de Laguerre n , o pólo de Laguerre p que parametriza a equação dinâmica de estados, o número de regras *fuzzy* M , os conjuntos *fuzzy* (Gaussianos) das variáveis da premissa e o vetor de parâmetros ζ . Os conjuntos *fuzzy* são definidos através da mesma distribuição homogênea adotada nas aplicações do capítulo

5 (seções 5.8.1 e 5.9). Quanto ao número de regras, adota-se a mesma parametrização descrita na seção 5.3.4 e utilizada na aplicação da seção 5.9, referente a um modelo com base de regras completa. Essa parametrização é dada por $M = c^n$, onde c é o número de conjuntos referenciais *fuzzy* por estado de Laguerre. Para evitar um número excessivo $\mu = c^n(n + 1)$ de elementos no vetor de parâmetros ζ a ser estimado pelo algoritmo dos mínimos quadrados, seleciona-se $c = 2$. Esse valor é capaz de proporcionar resultados satisfatórios na presente aplicação, conforme mostrado a seguir.

O comportamento dinâmico do processo de polimerização representado pela sua resposta temporal nos dados disponíveis para a estimação e validação do modelo indica que a dinâmica dominante desse processo pode ser grosseiramente aproximada por um filtro discreto de primeira ordem com pólo real no intervalo $[0.65, 0.85]$. Logo, é adequado que o pólo de Laguerre esteja contido nesse intervalo. Após selecionado o valor exato do pólo e a quantidade n de estados de Laguerre, restará obter os parâmetros do vetor ζ . Conforme discutido na seção 5.4, o modelo é linear nesses parâmetros que podem, portanto, ser estimados utilizando o algoritmo dos mínimos quadrados.

A tabela 6.1 ilustra o efeito de variar o pólo p para uma quantidade fixa de estados n . A segunda e terceira colunas referem-se ao erro quadrático médio entre a saída do sistema e a série sintética do modelo para os conjuntos de dados de estimação e validação, respectivamente. Pode-se observar que os erros de modelagem decrescem até atingirem patamares mínimos com $p = 0.75$, valor a partir do qual os erros voltam a crescer. Seleciona-se então o pólo de Laguerre como $p = 0.75$. A tabela 6.2 apresenta o resultado de variar a quantidade de estados n com o pólo mantido fixo no valor selecionado. A quarta coluna refere-se ao número de parâmetros estimados utilizando o algoritmo de mínimos quadrados. A figura 6.1 ilustra a precisão do modelo com 2 estados de Laguerre na previsão de horizonte longo da saída do processo. Esse modelo exibe um compromisso adequado entre quantidade de parâmetros e precisão, sendo por esta razão selecionado para fins de controle do reator de polimerização.

Tabela 6.1: Desempenho de modelagem em função do pólo de Laguerre p ($n = 2$).

Pólo p	EQM est. ($\times 10^6$)	EQM val. ($\times 10^6$)
0.65	0.58236	0.28159
0.70	0.18720	0.09762
0.75	0.02213	0.02513
0.80	0.06100	0.08900
0.85	0.15033	0.26743

Tabela 6.2: Desempenho de modelagem em função do número de funções e estados de Laguerre n ($p = 0.75$).

n	EQM est. ($\times 10^6$)	EQM val. ($\times 10^6$)	No. Parâmetros
1	4.18181	1.41269	4
2	0.02213	0.02513	12
3	0.00361	0.00452	32

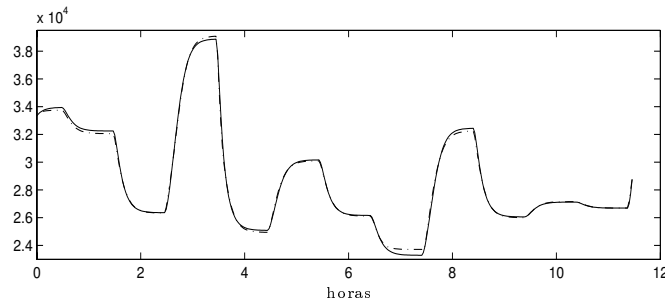


Figura 6.1: Saída do sistema (linha cheia) e série sintética do modelo FBO-TS com $n = 2$ e $p = 0.75$ (linha tracejada) [kg/kmol]: Dados de validação.

6.4.2 Controle do Reator CSTR

O esquema de controle é implementado conforme descrito na seção 6.2. Considera-se explicitamente no controlador GPC as restrições referentes ao intervalo operacional da variável manipulada, i.e., $u \in [0.002, 0.02]$. Os parâmetros do controlador são sintonizados experimentalmente como $N_1 = 1$ (horizonte inicial), $N_u = 1$ (horizonte de controle), $N_y = 4$ (horizonte de previsão) e $\lambda = 0.0001$ (fator de ponderação do sinal de controle). Habilita-se o controle em malha fechada após um período transitório de 1h necessário para a convergência dos estados de Laguerre (configurados nulos inicialmente) e ao longo do qual mantém-se constante a variável manipulada u . Dado que o processo de interesse é invariante no tempo, não se faz necessária qualquer adaptação *on-line* dos parâmetros do modelo ou do controlador. A figura 6.2 apresenta a simulação em malha fechada do CSTR de polimerização e mostra que o esquema proposto, baseado na linearização do modelo FBO-TS, é capaz de controlar esse processo não linear com desempenho adequado.

Em (Oliveira e Amaral, 2000) utilizou-se esse mesmo reator de polimerização para avaliar o desempenho de controladores preditivos baseados em modelos com estrutura FBO. Consideraram-se dois controladores: Um controlador preditivo linear baseado em um modelo de Laguerre também linear do processo e um controlador preditivo não linear baseado em um modelo Wiener-Volterra com base de Laguerre. Mostrou-se que esse último, conforme esperado, foi capaz de controlar o processo com um desempenho

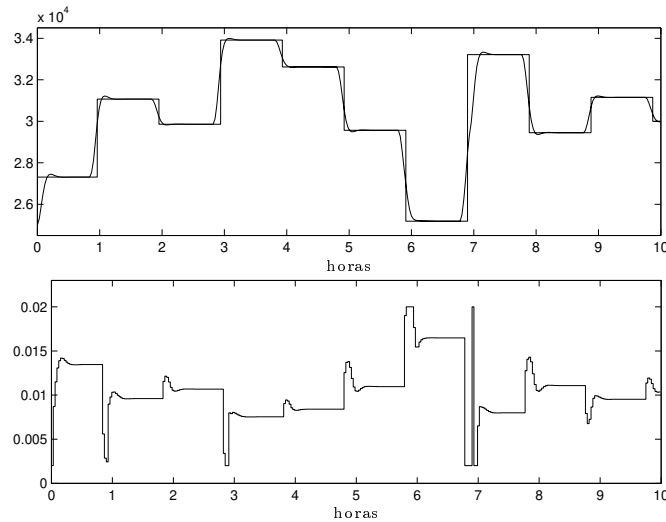


Figura 6.2: Referência e saída do sistema em malha fechada [kg/kmol] (acima); Variável manipulada [m³/h] (abaixo).

muito superior ao primeiro em termos de máximo *overshoot* e tempo de estabelecimento, embora baseado em um modelo Wiener-Volterra com um total de apenas 10 parâmetros estimados pelo algoritmo de mínimos quadrados. O desempenho do controlador GPC apresentado na figura 6.2, baseado em um modelo FBO-TS com um número similar de parâmetros (12), é ainda superior ao mencionado acima, com máximo *overshoot* e tempo de subida/estabelecimento inferiores.

6.5 Controle Preditivo de um Processo Biotecnológico

A seguir considera-se a planta biotecnológica para produção de etanol utilizada para a avaliação dos modelos FBO no capítulo 5, seção 5.9. O objetivo é controlar a saída de interesse P_4 (concentração de álcool no produto final bruto [kg/m³]) manipulando a variável de entrada F_a (vazão do mosto [m³/h]), sendo que a saída é também influenciada por uma perturbação S_0 (concentração de açúcar no mosto [kg/m³]) considerada não mensurável.

O esquema de controle é implementado conforme descrito na seção 6.2. Considera-se explicitamente no controlador GPC as restrições referentes ao intervalo operacional da variável manipulada, i.e., $F_a \in [50, 150]$. Nos experimentos descritos a seguir, mantém-se F_a constante em 100m³/h durante 20 instantes de amostragem para garantir a convergência dos estados de Laguerre antes de se habilitar o controle em malha fechada.

Inicialmente, assume-se que não existe qualquer perturbação atuando no processo,

ou seja, S_0 é mantida fixa em seu valor nominal ($S_0 = 180\text{kg/m}^3$). Essa condição corresponde ao primeiro cenário de modelagem adotado na seção 5.9. Por esta razão, o melhor modelo TS-FBO correspondente² será utilizado para a implementação do controlador, cujos parâmetros são sintonizados experimentalmente como $N_1 = 1$, $N_u = 1$, $N_y = 3$ e $\lambda = 0.0001$. As figuras 6.3 e 6.5 apresentam o comportamento da planta de etanol em malha fechada para diferentes sinais de referência e mostram que o controlador GPC com modelo de previsão FBO-TS linearizado é capaz de controlar esse processo não linear adequadamente. Para fins de ilustração, a figura 6.4 apresenta a previsão do modelo ao longo do primeiro experimento.

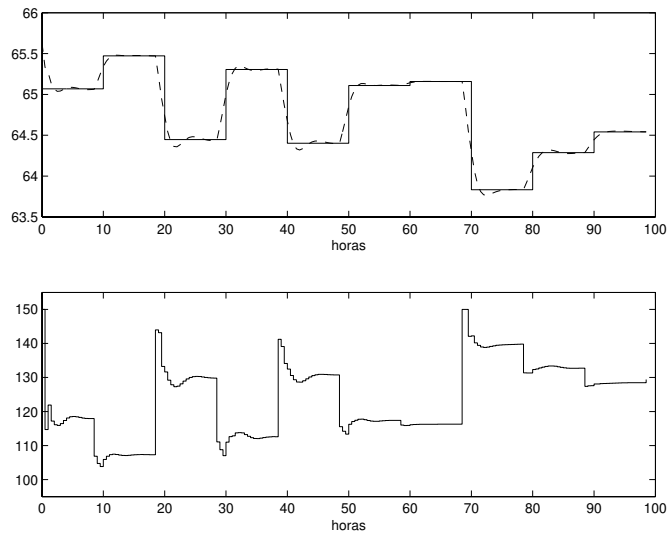


Figura 6.3: Acima: Referência e saída do processo em malha fechada (P_4 [kg/m^3]); Abaixo: Variável manipulada (F_a [m^3/h]).

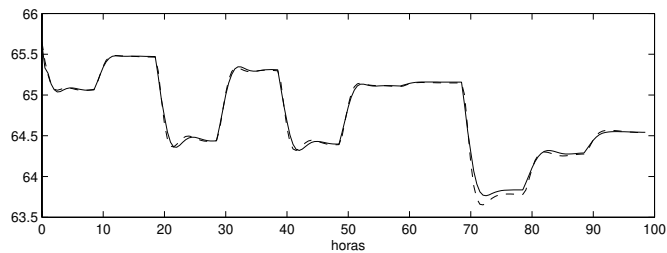


Figura 6.4: Saída do processo (P_4 [kg/m^3], linha contínua) e série sintética do modelo (linha tracejada).

No que segue, considera-se a presença de perturbações atuando sobre o processo e sobre o sistema de controle. Por esta razão, adota-se o melhor modelo FBO-TS obtido

²Identificado utilizando o conjunto de dados I e $n = 3$ funções de Laguerre.

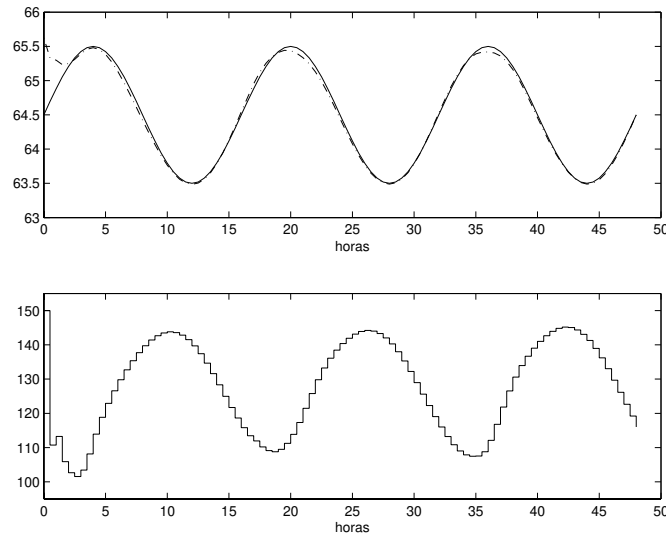


Figura 6.5: Acima: Referência e saída do processo em malha fechada (P_4 [kg/m³]); Abaixo: Variável manipulada (F_a [m³/h]).

no segundo cenário de modelagem apresentado na seção 5.9³, onde a planta esteve sujeita a uma perturbação estocástica em S_0 e a ruído branco aditivo na instrumentação de medida da saída P_4 . No primeiro experimento considera-se apenas a presença de ruído no sensor de saída. Especificamente, um ruído branco com distribuição de probabilidade uniforme no intervalo $[-0.2, 0.2]$ é adicionado ao sinal de P_4 antes que esse sinal seja mensurado pelo controlador. A figura 6.6 apresenta a simulação da planta em malha fechada com a mesma sintonia de controle utilizada nos experimentos anteriores, exceto $\lambda = 0.0005$. Essa figura mostra que, apesar da presença do ruído de medida, o sistema de controle ainda apresenta um desempenho satisfatório.

No experimentos a seguir avalia-se o desempenho do controlador na presença de perturbações em S_0 . Inicialmente, nenhum ruído de medida é utilizado. A figura 6.7 apresenta uma simulação do sistema em malha fechada com a mesma sintonia do experimento anterior, exceto $\lambda = 0.0001$, e uma perturbação do tipo pulso em S_0 . Essa figura mostra que o controlador foi capaz de rejeitar a perturbação agindo sobre o processo. Considera-se então uma perturbação estocástica em S_0 . Especificamente, modifica-se o valor de S_0 aleatoriamente em cada instante de amostragem segundo uma distribuição de probabilidade Gaussiana. O valor é mantido constante entre dois instantes consecutivos apenas para simplificar os procedimentos de simulação. O controlador é re-sintonizado com $\lambda = 0.0005$. A figura 6.8 apresenta a respectiva simulação do sistema em malha fechada mostrando que, apesar da presença da perturbação não modelada, o sistema ainda apresenta um desempenho satisfatório.

Considera-se agora a presença simultânea da mesma perturbação estocástica do

³Identificado utilizando o conjunto de dados II e $n = 2$ funções de Laguerre.

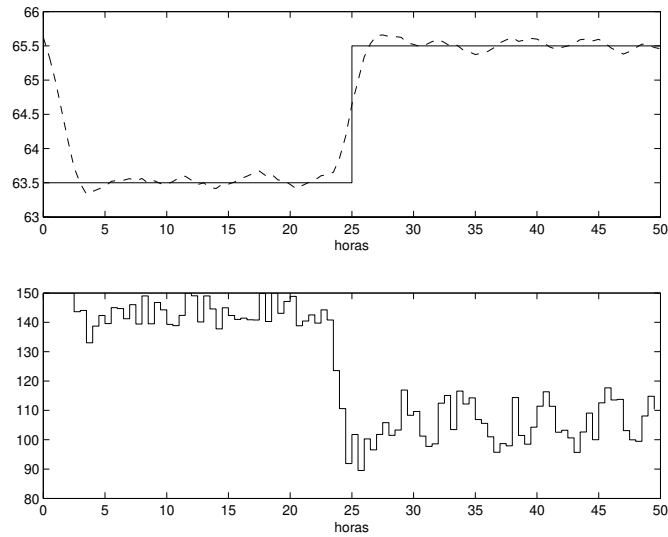


Figura 6.6: Acima: Referência e saída do processo em malha fechada (P_4 [kg/m^3]); Abaixo: Variável manipulada (F_a [m^3/h]).

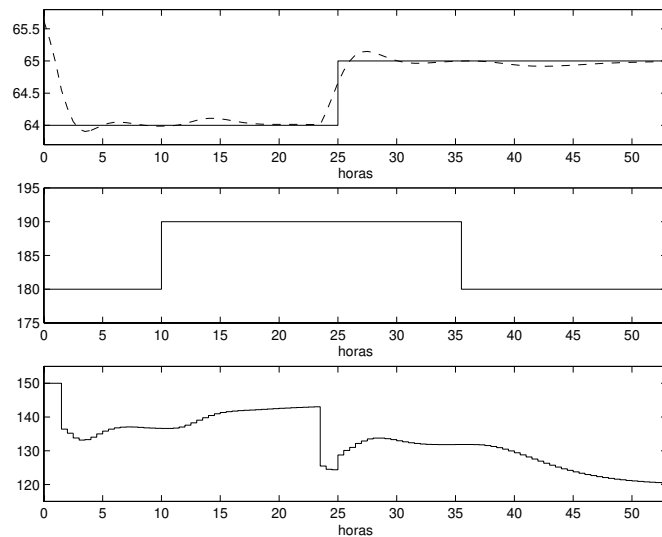


Figura 6.7: Acima: Referência e saída do processo em malha fechada (P_4 [kg/m^3]); Ao centro: Entrada de perturbação (S_0 [kg/m^3]); Abaixo: Variável manipulada (F_a [m^3/h]).

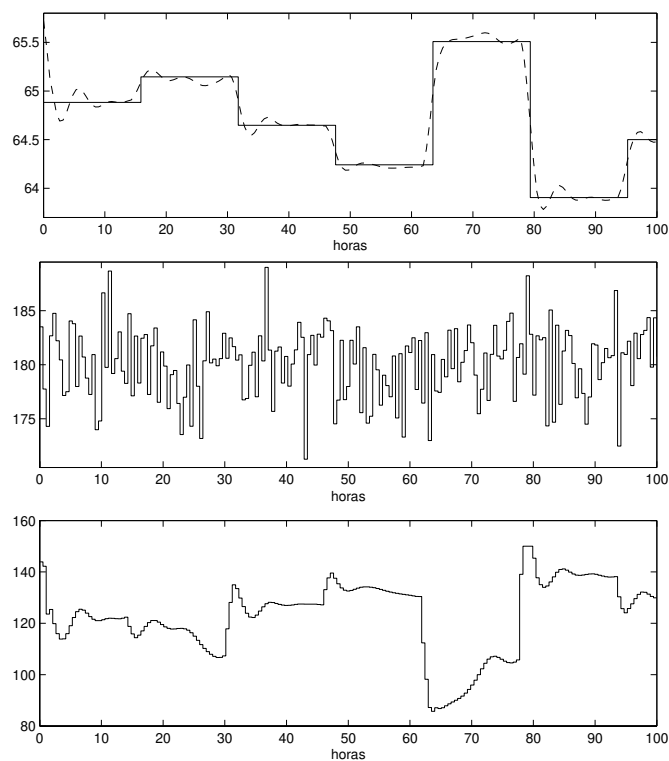


Figura 6.8: Acima: Referência e saída do processo em malha fechada (P_4 [kg/m³]); Ao centro: Entrada de perturbação (S_0 [kg/m³]); Abaixo: Variável manipulada (F_a [m³/h]).

experimento anterior e do mesmo ruído de medida do experimento referente à figura 6.6. Nesse caso, o controlador é re-sintonizado com $\lambda = 0.001$. A figura 6.9 apresenta a simulação correspondente do sistema em malha fechada. Essa figura mostra que, apesar da presença da perturbação não modelada e do forte ruído de medida, o controlador ainda é capaz de controlar o processo.

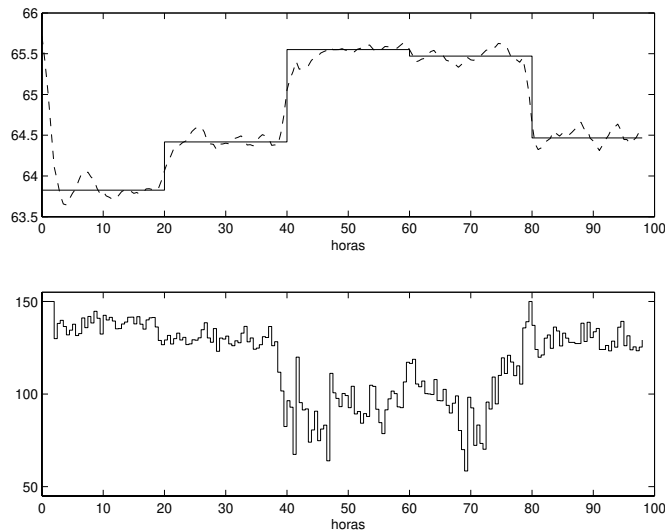


Figura 6.9: Acima: Referência e saída do processo em malha fechada (P_4 [kg/m^3]); Abaixo: Variável manipulada (F_a [m^3/h]).

Os experimentos acima indicam que a utilização de sistemas auxiliares permitem melhorar significativamente a qualidade da malha de controle e, conseqüentemente, o desempenho do processo de produção de etanol. Particularmente, esses sistemas dizem respeito ao aparato de instrumentação bem como ao processo de pré-tratamento que é dado ao meio de alimentação (Mosto) antes da utilização do mesmo como matéria prima para o bioprocessamento em questão. Conforme observado nas simulações, é recomendável um pré-tratamento adequado tal que as características químicas relevantes do Mosto, em especial S_0 , sejam mantidas aproximadamente constantes ao longo do tempo. Vale dizer que uma diferença de $1\text{kg}/\text{m}^3$ na concentração de álcool no produto bruto final do processo representa, a uma vazão média de $100\text{ m}^3/\text{h}$, uma diferença de 100kg de etanol produzidos por hora pela planta.

Finalmente, é importante observar que o modelo FBO-TS utilizado pelo controlador nos experimentos com a presença de perturbações e ruídos de medida foi obtido na seção 5.9 sujeito a essas mesmas adversidades (e sem que os seus conjuntos referenciais *fuzzy* tenham sido otimizados). Naquela seção, evidenciou-se a forte influência da perturbação não modelada S_0 no comportamento dinâmico do processo e conseqüentemente nos erros de previsão do modelo.

6.6 Resumo e Discussões Complementares

Esse capítulo abordou o problema de controle automático de sistemas não lineares através de estratégias de controle preditivo baseadas em modelos dos processos a serem controlados. As análises concentraram-se nos modelos *fuzzy* com estrutura FBO desenvolvidos no capítulo 5, que constituem uma das contribuições centrais do presente trabalho. Propôs-se uma estratégia de controle preditivo para ilustrar o potencial de aplicação desses modelos ao controle de sistemas complexos. Essa estratégia consiste da utilização do Controlador Preditivo Generalizado (GPC) associado a um procedimento de linearização instantânea do modelo *fuzzy* FBO (preditor) do processo a ser controlado. Apresentaram-se duas aplicações do controlador GPC com preditor *fuzzy* FBO linearizado no contexto de processos químicos. Em uma dessas aplicações, referente ao controle de um reator de polimerização do tipo CSTR, verificou-se que o controlador proposto apresentou desempenho em malha fechada superior àquele apresentado por controladores preditivos baseados em modelos lineares e de Volterra com estrutura FBO, cuja aplicação ao controle desse mesmo processo foi descrita na literatura. Em outra aplicação, referente ao controle de uma planta biotecnológica para produção de etanol, mostrou-se que o controlador proposto apresentou desempenho satisfatório mesmo na presença de perturbações agindo sobre o processo. Apesar desses resultados promissores, é importante observar que a estratégia de controle descrita acima foi desenvolvida nesse capítulo apenas em caráter ilustrativo. De fato, o tema merece ainda estudos mais detalhados em tópicos fundamentais como aqueles referentes à questão da estabilidade do sistema em malha fechada.

6.7 Contribuições e Perspectivas

As principais contribuições desse capítulo são:

1. Desenvolvimento de uma estratégia de controle preditivo baseada na utilização do Controlador Preditivo Generalizado (GPC) associado a um procedimento de linearização instantânea de modelos *fuzzy* com estrutura FBO.
2. Aplicação da estratégia proposta ao controle de dois processos químicos: Um reator de polimerização e uma planta biotecnológica para produção de álcool etílico (etanol).

As principais perspectivas a partir dos resultados obtidos e apresentados no presente capítulo são:

1. Estudo de estratégias alternativas para a linearização dos modelos FBO-TS, tal como a utilização dos próprios modelos locais lineares das regras TS estimados de forma específica para esta finalidade através de mínimos quadrados locais (Fischer et al., 1998b; Fischer et al., 1998a; Cavalcante Júnior, 2000).

2. Implementação do controlador GPC utilizando diretamente os modelos não lineares, sem qualquer linearização, para a construção do vetor de previsões e da matriz dinâmica de resposta ao impulso. Comparação dessa estratégia com aquelas baseadas na linearização dos modelos.
3. Implementação do controlador GPC com linearização do modelo ao longo do horizonte de previsão (Cavalcante Júnior, 2000).
4. Comparações entre as estratégias de controle baseadas na linearização dos modelos e estratégias não lineares baseadas em técnicas de otimização não convexa, como SQP (Meleiro, 2002; Meleiro et al., nd).
5. Desenvolvimento de controladores multivariáveis baseados em modelos *fuzzy* com estrutura FBO.
6. Incorporação de restrições de estabilidade e critérios de robustez ao projeto dos controladores.

Capítulo 7

Conclusões e Perspectivas

7.1 Conclusões

Neste trabalho abordou-se o problema da modelagem de sistemas complexos através da utilização conjunta de ferramentas matemáticas e computacionais clássicas e modernas. Por ferramentas clássicas entende-se aquelas bem estabelecidas até os anos 70, tais como estimadores de mínimos quadrados, métodos de otimização não linear, funções ortonormais e séries de Volterra, enquanto ferramentas modernas referem-se àquelas relacionadas ao campo da inteligência computacional, isto é, sistemas *fuzzy*, redes neurais artificiais e computação evolutiva.

O trabalho enfocou inicialmente os problemas de modelagem quantitativa e qualitativa de sistemas complexos através de uma abordagem integrada centrada na utilização de modelos relacionais *fuzzy*. Propôs-se uma metodologia sistemática para o projeto estrutural dos modelos que permite uma formulação convexa para a otimização dos seus parâmetros desconhecidos. Essa formulação admite uma solução ótima global analítica que pode ser obtida numericamente através de mínimos quadrados. A interpretabilidade dos modelos é considerada através da utilização de uma função objetivo *fuzzy* e da inserção de restrições no problema de otimização, que nesse caso pode ser resolvido utilizando programação quadrática. Após a etapa de estimação do modelo, conhecimento lingüístico na forma de um conjunto consistente (não conflitante) de regras *fuzzy* pode ser extraído utilizando os novos conceitos de contribuição e representabilidade que foram introduzidos como um novo paradigma para a interpretação de modelos relacionais. Esses conceitos permitem também a simplificação da base de regras extraída através da eliminação de regras menos representativas. Ilustrou-se o desempenho das metodologias propostas através de exemplos práticos e acadêmicos. Os resultados obtidos mostram que é possível gerar modelos representativos sem necessariamente modificar os conjuntos *fuzzy* pré-definidos pelo usuário/especialista, não comprometendo assim o significado lingüístico dos mesmos. Com isso, obtém-se um melhor compromisso entre os aspectos quantitativo e qualitativo envolvidos no problema de modelagem, refletido na capacidade de aproximação e interpretabilidade dos respectivos modelos.

Outro tema abordado no trabalho foi o problema da dimensionalidade em modelos *fuzzy* e neurais. Uma alternativa para contornar esse problema é a utilização de uma estrutura hierárquica constituída de um conjunto de submodelos conectados em cascata. Nesse contexto, dois novos tipos de modelos *fuzzy*/neurais hierárquicos foram propostos. Desenvolveram-se conjuntos fechados de equações para o cálculo do gradiente da saída de cada modelo em função dos seus parâmetros de projeto livre. Essas equações permitem a estimação ou sintonia-fina desses parâmetros a partir de algoritmos clássicos de otimização não linear. Mostrou-se que o desempenho dos modelos é superior com relação às suas variáveis de entrada dispostas em níveis hierárquicos mais elevados. Com base nesse resultado, arquiteturas hierárquicas alternativas foram propostas para permitir a atribuição de níveis hierárquicos equivalentes a variáveis distintas. Mostrou-se também que uma das classes de modelos hierárquicos proposta apresenta a capacidade de aproximação universal de mapeamentos contínuos em domínios compactos e que existe uma transformação unilateral entre esses modelos e modelos *fuzzy* não hierárquicos. Essa transformação permite a extração de regras lingüísticas a partir do conjunto reduzido de parâmetros da arquitetura hierárquica em questão.

Um terceiro tema investigado foi a identificação de sistemas dinâmicos utilizando modelos com estrutura de funções de base ortonormal (FBO). No presente trabalho, os modelos FBO foram analisados a partir de um ponto de vista matemático original que resulta em uma representação genérica de espaço-de-estados do tipo Wiener. Foram propostas realizações particulares dessa representação através do uso de sistemas *fuzzy*. Os modelos *fuzzy* FBO propostos foram escritos utilizando uma formulação unificada que também abrange outras classes de modelos FBO, como os modelos lineares e os modelos Wiener-Volterra. Essa formulação permite a interpretação das diferenças entre as diversas classes de modelos FBO analisadas, diferenças essas avaliadas através de estudos teóricos e experimentos computacionais. O resultado principal é que os modelos FBO *fuzzy* são capazes de proporcionar, em determinadas aplicações, um melhor compromisso entre número de parâmetros e precisão quando comparados aos modelos Wiener-Volterra, além de aproximarem arbitrariamente bem uma ampla classe de sistemas dinâmicos não lineares. A questão da interpretabilidade matemática desses modelos foi abordada através da proposta de um modelo FBO *fuzzy* generalizado. Esse modelo consiste de uma interpolação do tipo Takagi-Sugeno de diversos modelos locais lineares com estrutura FBO. Quando as dinâmicas de estados dos diferentes modelos locais são iguais, obtém-se os demais modelos FBO *fuzzy* como realizações particulares da representação generalizada, o que permite uma interpretação matemática clara dos mesmos. Dentro do contexto de modelos FBO multivariáveis, foram desenvolvidas diferentes arquiteturas de modelos FBO hierárquicos bem como uma estratégia híbrida para a estimação paramétrica dos mesmos. Essa estratégia consiste da utilização de um algoritmo genético para a obtenção de um modelo inicial aproximado a ser refinado por um algoritmo baseado em gradiente, tendo sido aplicada com sucesso à modelagem de uma planta biotecnológica para produção de álcool etílico (etanol) em escala industrial. Discutiram-se também estratégias para o projeto das bases de funções ortonormais

utilizadas na implementação computacional dos modelos FBO. O principal resultado nesse contexto consiste de uma solução analítica global para o desenvolvimento ótimo de kernels de Volterra em funções ortonormais de Laguerre. Essa solução minimiza, segundo um critério específico, o erro de aproximação associado ao desenvolvimento truncado dos kernels em um número finito de funções, o que permite uma redução de ordem da representação em espaço-de-estados do modelo Wiener-Volterra resultante. A solução ótima foi obtida tanto para modelos com bases de Laguerre independentes para o desenvolvimento de cada kernel como para os modelos usuais onde utiliza-se uma base comum ao desenvolvimento compartilhado de todos o kernels. Propôs-se ainda um algoritmo iterativo para a estimação simultânea dos kernels de Volterra e da solução ótima para o desenvolvimento desses kernels em funções de Laguerre utilizando dados de E/S de um sistema a ser modelado.

Finalmente, o trabalho abordou o problema de controle automático de sistemas dinâmicos não lineares através de estratégias de controle preditivo baseadas em modelos dos processos a serem controlados. As discussões concentraram-se nos modelos *fuzzy* com estrutura FBO, que constituem uma das contribuições centrais do trabalho. Propôs-se uma estratégia de controle preditivo para ilustrar o potencial de aplicação desses modelos ao controle de sistemas complexos. Essa estratégia consiste do uso do Controlador Preditivo Generalizado (GPC) associado a um procedimento de linearização instantânea do modelo *fuzzy* FBO (preditor) do processo a ser controlado. Foram apresentadas duas aplicações do controlador GPC com preditor *fuzzy* FBO linearizado no contexto de processos químicos. Em uma dessas aplicações, referente ao controle de um reator de polimerização do tipo CSTR, mostrou-se que o controlador proposto apresentou um desempenho em malha fechada superior àquele apresentado por controladores preditivos baseados em modelos lineares e de Volterra com estrutura FBO, cuja aplicação a esse mesmo processo foi relatada na literatura. Em outra aplicação, referente ao controle de uma planta biotecnológica para produção de etanol, mostrou-se que o controlador proposto apresentou desempenho satisfatório mesmo na presença de perturbações agindo sobre o processo. Apesar dos resultados promissores, é importante observar que esse tema merece estudos mais detalhados em tópicos fundamentais, que incluem a questão das condições de estabilidade da malha de controle.

7.2 Perspectivas

As perspectivas para pesquisas futuras relacionadas aos temas abrangidos pelo presente trabalho foram apresentadas detalhadamente ao final de cada um dos capítulos. Em resumo, as principais perspectivas são:

- **Modelagem Relacional:**

1. Aprimoramento da metodologia de identificação através de estudos sobre:
 - i) Condições de inversibilidade da matriz Hessiana na solução analítica do

problema de otimização dos modelos; *ii*) Condições de aplicação de algoritmos rápidos para a solução recursiva alternativa dos mínimos quadrados na presença de um número elevado de parâmetros relacionais a serem estimados.

- **Modelagem Hierárquica:**

1. Aplicação do Filtro de Kalman Estendido para a estimação *on-line* dos parâmetros dos modelos.
2. Desenvolvimento de uma estratégia construtiva para a determinação paramétrica e estrutural simultânea dos modelos.

- **Modelagem FBO:**

1. Desenvolvimento de soluções ótimas para a seleção de bases de funções ortonormais em modelos FBO *fuzzy*.
2. Desenvolvimento de procedimentos sistemáticos para a estimação simultânea e adaptativa das bases de funções ortonormais e dos demais parâmetros dos modelos FBO.
3. Estudo sobre as condições de convergência do algoritmo iterativo para a estimação simultânea dos kernels de Volterra e da solução ótima para o desenvolvimento desses kernels em funções de Laguerre, além da extensão do algoritmo e da solução ótima no que se refere a outras bases ortonormais, tais como as bases de Kautz e as bases generalizadas.

- **Controle:**

1. Aprimoramento do sistema de controle via linearização de modelos FBO-TS através de estratégias de linearização alternativas.
2. Desenvolvimento de controladores multivariáveis baseados nesses modelos.
3. Incorporação de restrições de estabilidade e critérios de robustez ao projeto dos controladores.

Apêndice A

Fundamentos em Conjuntos e Sistemas *Fuzzy*

Esse apêndice apresenta um conjunto de fundamentos da teoria de sistemas *fuzzy* que se mostraram de particular relevância ao longo dos desenvolvimentos do presente trabalho.

A.1 Modelos Baseados em Regras *Fuzzy*

A seguir descrevem-se em detalhes os modelos *fuzzy* baseados em regras dos tipos lingüístico (Mamdani) e Takagi-Sugeno, bem como os seus respectivos mecanismos de inferência. Os conceitos e definições básicas envolvidos nas discussões estão convenientemente compilados em (Lee, 1990; Tanscheit, 1992; Yager e Filev, 1994; Mendel, 1995; Pedrycz e Gomide, 1998).

A.1.1 Modelos Lingüísticos (de Mamdani)

Considere o seguinte conjunto de M implicações (regras) *fuzzy*:

$$\begin{aligned} R^1: & \text{SE } x_1 \text{ é } X_1^1 \text{ E } \cdots \text{ E } x_n \text{ é } X_n^1 \text{ ENTÃO } y \text{ é } Y^1 \\ & \text{OU} \\ & \vdots \\ & \text{OU} \\ R^M: & \text{SE } x_1 \text{ é } X_1^M \text{ E } \cdots \text{ E } x_n \text{ é } X_n^M \text{ ENTÃO } y \text{ é } Y^M \end{aligned} \tag{A.1}$$

onde R^i denota a i -ésima implicação, $x_j \in \mathbf{X}_j$ ($j = 1, \dots, n$) são as variáveis de entrada (variáveis da premissa), $y \in \mathbf{Y}$ é a variável de saída e X_j^i e Y^i ($i = 1, \dots, M$) representam conjuntos *fuzzy* definidos sobre os universos de discurso \mathbf{X}_j e \mathbf{Y} das respectivas variáveis, de modo que:

$$X_j^i : \mathbf{X}_j \rightarrow [0, 1], \quad Y^i : \mathbf{Y} \rightarrow [0, 1] \quad (\text{A.2})$$

Cada implicação em (A.1) é descrita matematicamente através de uma relação *fuzzy* R^i , isto é, um conjunto *fuzzy* multidimensional definido sobre o espaço dos Produtos Cartesianos (\times) dos universos de discurso \mathbf{X}_j e \mathbf{Y} , como segue:

$$R^i : \mathbf{X}_1 \times \cdots \times \mathbf{X}_n \times \mathbf{Y} \rightarrow [0, 1] \quad (\text{A.3})$$

Usualmente os universos de discurso são discretizados para fins de implementação computacional. Nesses casos, a relação *fuzzy* torna-se uma matriz $(n+1)$ -dimensional denominada matriz relacional *fuzzy*.

As relações *fuzzy* R^i que descrevem as implicações são calculadas através de uma função de implicação. Uma grande quantidade de funções de implicação foram propostas e analisadas na literatura (Mizumoto, 1982; Kiszka et al., 1985). Genericamente, as implicações podem ser descritas como:

$$R^i(x_1, \dots, x_n, y) = \mathcal{F} (X_1^i(x_1) \& \cdots \& X_n^i(x_n), Y^i(y)) \quad (\text{A.4})$$

onde \mathcal{F} é uma função de implicação, “&” é um operador associado ao conectivo “E” das premissas em (A.1) e $R^i(x_1, \dots, x_n, y)$ é a relação *fuzzy* associada à i -ésima implicação.

O conjunto de implicações em (A.1) é combinado (agregado) como:

$$\begin{aligned} R(x_1, \dots, x_n, y) &= \bigcup \{ R^1, \dots, R^M \} = \\ &= \bigcup \{ \mathcal{F} (X_1^1(x_1) \& \cdots \& X_n^1(x_n), Y^1(y)), \\ &\quad \vdots \\ &\quad \mathcal{F} (X_1^M(x_1) \& \cdots \& X_n^M(x_n), Y^M(y)) \} \end{aligned} \quad (\text{A.5})$$

onde $R(x_1, \dots, x_n, y)$ é a relação *fuzzy* resultante e \bigcup é um operador de agregação associado ao conectivo “OU” do conjunto de implicações.

Um conjunto *fuzzy* de saída Y' pode ser inferido a partir de uma *composição fuzzy* entre a relação R e uma dada coleção de conjuntos *fuzzy* de entrada X'_1, \dots, X'_n , como segue:

$$Y'(y) = (X'_1(x_1) \& \cdots \& X'_n(x_n)) \bullet R(x_1, \dots, x_n, y) \quad (\text{A.6})$$

onde “ \bullet ” representa o operador de composição. Os conjuntos *fuzzy* de entrada são obtidos a partir da *fuzzificação* de um dado conjunto de entradas *crisp* (não *fuzzy*) $x_1 = x'_1, \dots, x_n = x'_n$, usualmente através do método *singleton*, i.e.:

$$X'_j(x_j) = \begin{cases} 1, & \text{se } x_j = x'_j \\ 0, & \text{caso contrário} \end{cases} \quad (\text{A.7})$$

Inferência Min-Max-Gravidade e Produto-Soma-Gravidade

Sejam “&” e “•” em (A.6) dados respectivamente por uma norma triangular (norma-t) (Pedrycz, 1993) e pela composição clássica max-min introduzida por Zadeh (1973). Nesse caso, a equação (A.6) é reescrita como:

$$Y'(y) = \bigvee_{x_1 \in \mathbf{X}_1, \dots, x_n \in \mathbf{X}_n} (X'_1(x_1) \text{ t } \dots \text{ t } X'_n(x_n)) \wedge R(x_1, \dots, x_n, y) \quad (\text{A.8})$$

onde “ \vee ” e “ \wedge ” denotam os operadores max e min, respectivamente, e “t” é uma norma-t. Seja também o procedimento de fuzzificação implementado através do método singleton em (A.7). Logo, dado que $0 \text{ t } \alpha = 0$ e $1 \text{ t } \alpha = \alpha$ para todo $\alpha \in [0, 1]$ (condições limitantes das normas triangulares), a seguinte expressão é válida:

$$\begin{aligned} (X'_1(x_1) \text{ t } \dots \text{ t } X'_n(x_n)) \wedge R(x_1, \dots, x_n, y) &= \\ &= \begin{cases} R(x'_1, \dots, x'_n, y), & \text{se } x_j = x'_j \text{ (} j = 1, \dots, n \text{)} \\ 0, & \text{caso contrário} \end{cases} \end{aligned} \quad (\text{A.9})$$

As equações (A.8) e (A.9) implicam:

$$Y'(y) = R(x'_1, \dots, x'_n, y) \quad (\text{A.10})$$

e as equações (A.5) e (A.10) resultam em:

$$\begin{aligned} Y'(y) &= \bigcup \{ \mathcal{F}(X_1^1(x'_1) \text{ t } \dots \text{ t } X_n^1(x'_n), Y^1(y)), \\ &\quad \vdots \\ &\quad \mathcal{F}(X_1^M(x'_1) \text{ t } \dots \text{ t } X_n^M(x'_n), Y^M(y)) \} \end{aligned} \quad (\text{A.11})$$

Se a função de implicação \mathcal{F} e a norma triangular “t” forem realizadas através do operador min, além do operador de agregação \bigcup implementado utilizando o operador max, então a equação (A.11) é reescrita como:

$$\begin{aligned} Y'(y) &= \bigvee (X_1^1(x'_1) \wedge \dots \wedge X_n^1(x'_n) \wedge Y^1(y), \\ &\quad \vdots \\ &\quad X_1^M(x'_1) \wedge \dots \wedge X_n^M(x'_n) \wedge Y^M(y)) \end{aligned} \quad (\text{A.12})$$

Por outro lado, se a função de implicação e a norma triangular forem realizadas através do produto, bem como o operador de agregação implementado utilizando a soma, então a equação (A.11) é reescrita como:

$$\begin{aligned}
Y'(y) &= X_1^1(x'_1) \cdot \dots \cdot X_n^1(x'_n) \cdot Y^1(y) + \\
&\quad \vdots \\
&+ X_1^M(x'_1) \cdot \dots \cdot X_n^M(x'_n) \cdot Y^M(y)
\end{aligned} \tag{A.13}$$

Finalmente, utilizando o bem conhecido método de *defuzzificação* do Centro de Gravidade, pode-se calcular a saída *crisp* y' do modelo como:

$$y' = \frac{\int_{y \in \mathbf{Y}} Y'(y) y \, dy}{\int_{y \in \mathbf{Y}} Y'(y) \, dy} \tag{A.14}$$

ou ainda, no caso de um universo de discurso discreto $\mathbf{Y} = [y_1 \cdots y_m]$, como:

$$y' = \frac{\sum_{k=1}^m Y'(y_k) y_k}{\sum_{k=1}^m Y'(y_k)} \tag{A.15}$$

As equações (A.12) e (A.14) (ou (A.15)) compõem o método de inferência Min-Max-Gravidade de Mamdani (Mamdani, 1974; Mizumoto, 1991), enquanto as equações (A.13) e (A.14) (ou (A.15)) compõem o método de inferência Produto-Soma-Gravidade (Mizumoto, 1991; Mizumoto, 1996).

Observação *Pode-se observar que o conjunto fuzzy $Y'(y)$ na equação (A.13) pode, devido à operação de soma, exceder o grau de pertinência unitário. Para evitar essa inconsistência teórica, a média aritmética pode ser utilizada em lugar da soma algébrica, embora a saída y' em (A.14) (ou (A.15)) não seja afetada por essa modificação.*

Modelo Tipo Singleton (Caso Especial)

Um caso especial de particular interesse ocorre quando os conjuntos *fuzzy* nos conseqüentes das regras em (A.1) são singletons, i.e.

$$Y^i(y) = \begin{cases} 1, & \text{se } y = q_0^i \\ 0, & \text{caso contrário} \end{cases} \tag{A.16}$$

onde $q_0^i \in \mathbf{Y}$ ($i = 1, \dots, M$) são valores *crisp* no universo de discurso de saída. Nesse caso, considerando apenas por simplicidade e sem qualquer perda de generalidade $q_0^1 \neq q_0^2 \neq \dots \neq q_0^M$, é simples verificar que a equação (A.13) pode ser reescrita como:

$$Y'(y) = \begin{cases} w_i, & \text{se } y = q_0^i \ (i = 1, \dots, M) \\ 0, & \text{caso contrário} \end{cases} \quad (\text{A.17})$$

onde w_i é o nível de ativação (*firing weight*) da i -ésima regra, i.e., o resultado (*truth value*) da proposição “ x'_1 é X_1^i E \dots E x'_n é X_n^i ”, dado por:

$$w_i = X_1^i(x'_1) \cdot \dots \cdot X_n^i(x'_n) \quad (\text{A.18})$$

Conseqüentemente, a defuzzificação de $Y'(y)$ em (A.17) utilizando o centro de gravidade implica:

$$y' = \frac{\sum_{i=1}^M w_i q_0^i}{\sum_{i=1}^M w_i} \quad (\text{A.19})$$

A equação acima é exatamente a mesma que descreve os modelos Takagi-Sugeno quando os conseqüentes das regras desses modelos são constantes, ao invés de funções das variáveis da premissa, como mostrado na seção seguinte.

A.1.2 Modelos Takagi-Sugeno

Os modelos *fuzzy* do tipo Takagi-Sugeno (TS) são constituídos por um conjunto de M implicações (regras) do seguinte formato:

$$R^i: \text{ SE } x_1 \text{ é } X_1^i \text{ E } \dots \text{ E } x_n \text{ é } X_n^i \text{ ENTÃO } y = f_i(x_1, \dots, x_n) \quad (\text{A.20})$$

onde f_i ($i = 1, \dots, M$) são funções que relacionam as entradas do modelo com sua saída. As demais grandezas possuem o mesmo significado que nos modelos lingüísticos discutidos anteriormente.

Nos modelos TS a inferência de um valor de saída $y = y'$ a partir de um dado conjunto de valores de entrada $x_1 = x'_1, \dots, x_n = x'_n$ é calculada como a média ponderada das saídas individuais de cada implicação, como segue:

$$y' = \frac{\sum_i w_i y_i}{\sum_i w_i} \quad (\text{A.21})$$

onde $y_i \triangleq f_i(x'_1, \dots, x'_n)$ e w_i é o nível de ativação da i -ésima implicação, i.e.

$$w_i = X_1^i(x'_1) \text{ t } \dots \text{ t } X_n^i(x'_n) \quad (\text{A.22})$$

sendo “t” uma norma triangular, usualmente o produto algébrico. Sugeno e seus colaboradores (Takagi e Sugeno, 1985; Sugeno e Kang, 1986; Sugeno e Kang, 1988; Sugeno e Tanaka, 1991) propuseram a utilização de funções afins nos conseqüentes das implicações, ou seja:

$$f_i(x_1, \dots, x_n) = q_0^i + \sum_{j=1}^n q_j^i x_j \quad (\text{A.23})$$

Essa escolha permite uma interpretação matemática simples do modelo como uma interpolação de diferentes modelos afins e implica que a saída em (A.21) é *linear nos parâmetros* q_0^i, \dots, q_n^i . Logo, esses parâmetros podem ser estimados utilizando algoritmos de estimação clássicos (Söderström e Stoica, 1989; Ljung, 1999).

Finalmente, é interessante notar que, caso apenas os termos constantes q_0^i sejam considerados em (A.23), então o modelo (A.21) claramente coincide com o modelo lingüístico do tipo singleton dado pela equação (A.19).

A.2 Equivalência entre Modelos *Fuzzy* e Redes Neurais RBF

Considere o modelo *fuzzy* com estrutura relacional simplificada descrito na seção 4.2.1 (capítulo 4), porém com diferentes quantidades c_i ($i = 1, \dots, n$) de conjuntos referenciais *fuzzy* por variável de entrada. A equação (4.2) pode ser rescrita explicitamente como:

$$\hat{y} = \sum_{l_1=1}^{c_1} \dots \sum_{l_n=1}^{c_n} \psi_{l_1, \dots, l_n} \omega_{l_1, \dots, l_n} \quad (\text{A.24})$$

onde ψ_{l_1, \dots, l_n} e ω_{l_1, \dots, l_n} são elementos de Ψ e Ω , respectivamente, de tal modo que $\Psi = [\Psi_1 \dots \Psi_M]^T = [\psi_{1, \dots, 1} \dots \psi_{c_1, \dots, c_n}]^T$ e $\Omega = [\Omega_1 \dots \Omega_M]^T = [\omega_{1, \dots, 1} \dots \omega_{c_1, \dots, c_n}]^T$, com $M = c_1 c_2 \dots c_n$. Cada elemento ψ_{l_1, \dots, l_n} pode ser escrito a partir das equações (4.3) e (4.4) como:

$$\psi_{l_1, \dots, l_n} = \mathcal{X}_{l_1}(x_1) \mathcal{X}_{l_2}(x_2) \cdot \dots \cdot \mathcal{X}_{l_n}(x_n) \quad (\text{A.25})$$

Se conjuntos referenciais *fuzzy* Gaussianos forem utilizados, tem-se que a equação (A.25) pode ser rescrita utilizando (4.9) como:

$$\psi_{l_1, \dots, l_n} = \exp\left(-\frac{(x_1 - \theta_{1l_1})^2}{\sigma_{1l_1}^2}\right) \cdot \dots \cdot \exp\left(-\frac{(x_n - \theta_{nl_n})^2}{\sigma_{nl_n}^2}\right) \quad (\text{A.26})$$

Dado que o produto de n funções Gaussianas unidimensionais é uma função Gaussiana n -dimensional, a equação (A.26) implica:

$$\psi_{l_1, \dots, l_n} = \exp\left(-(\mathbf{x} - \boldsymbol{\theta}_{l_1, \dots, l_n})^T \Lambda^{-1} (\mathbf{x} - \boldsymbol{\theta}_{l_1, \dots, l_n})\right) \quad (\text{A.27})$$

onde

$$\begin{aligned} \mathbf{x} &= [x_1 \ x_2 \ \dots \ x_n]^T \\ \boldsymbol{\theta}_{l_1, \dots, l_n} &= [\theta_{1l_1} \ \theta_{2l_2} \ \dots \ \theta_{nl_n}]^T \end{aligned} \quad (\text{A.28})$$

$$\Lambda = \begin{bmatrix} \sigma_{1l_1}^2 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \sigma_{nl_n}^2 \end{bmatrix} \quad (\text{A.29})$$

Pode-se verificar a partir das equações (A.24) e (A.27) que a saída do modelo é dada por uma soma ponderada de funções Gaussianas multivariáveis, que é precisamente a arquitetura de uma rede neural RBF (Broomhead e Lowe, 1988; Haykin, 1999). Pode-se verificar ainda que essa mesma representação matemática é obtida, exceto por um fator de normalização, no caso dos modelos lingüísticos do tipo singleton em (A.19) e também no caso dos modelos Takagi-Sugeno com conseqüentes constantes¹ em (A.21) sempre que esses modelos forem implementados utilizando conjuntos *fuzzy* Gaussianos para as variáveis da premissa. Maiores detalhes sobre a equivalência entre diferentes classes de modelos *fuzzy* e redes neurais RBF podem ser encontrados em (Hunt et al., 1996; Cho e Wang, 1996; Campello e Amaral, 1999a).

A.3 Discretização Fuzzy

Seja $\mathcal{A}_1, \dots, \mathcal{A}_\eta$ uma coleção de conjuntos referenciais *fuzzy* definidos sobre o universo de discurso \mathbf{A} de uma variável *crisp* “ a ”, ou seja:

¹i.e., $y_i = q_0^i$.

$$\mathcal{A}_i : \mathbf{A} \rightarrow [0, 1], \quad i = 1, \dots, \eta \quad (\text{A.30})$$

sendo que esses conjuntos referenciais satisfazem a seguinte condição de completude:

$$\forall a \in \mathbf{A} \quad \exists i : 1 \leq i \leq \eta : \mathcal{A}_i(a) > 0 \quad (\text{A.31})$$

Tem-se então que, de acordo com o conceito de discretização *fuzzy*, qualquer conjunto *fuzzy* A em \mathbf{A} pode ser representado por um vetor de possibilidade $P = [P_1 \dots P_\eta]^T$ cujo i -ésimo elemento é dado pela medida de possibilidade (Zadeh, 1978) de A com relação ao i -ésimo conjunto referencial \mathcal{A}_i , como segue:

$$P_i = \text{Poss}(A|\mathcal{A}_i) \triangleq \sup_{a \in \mathbf{A}} [A(a) \text{ t } \mathcal{A}_i(a)], \quad (\text{A.32})$$

onde “t” é uma norma triangular.

Se o conjunto *fuzzy* A for definido como um singleton, isto é,

$$A(a) = \begin{cases} 1, & \text{se } a = a' \\ 0, & \text{caso contrário} \end{cases} \quad (\text{A.33})$$

então, devido às condições limitantes das normas triangulares (i.e., $0 \text{ t } \alpha = 0$ e $1 \text{ t } \alpha = \alpha$ $\forall \alpha \in [0, 1]$), as equações (A.32) e (A.33) implicam

$$P_i = \mathcal{A}_i(a') \quad (\text{A.34})$$

e conseqüentemente

$$P = [\mathcal{A}_1(a') \dots \mathcal{A}_\eta(a')]^T \quad (\text{A.35})$$

onde $\mathcal{A}_i(a') \in [0, 1]$ é o grau de pertinência de um dado valor *crisp* a' com relação ao i -ésimo conjunto referencial *fuzzy*. Um exemplo com $\eta = 2$ é ilustrado na figura A.1.

Note que o vetor de possibilidade P é, por definição, um conjunto *fuzzy* com η elementos de discretização. Uma vez que uma variável lingüística (Zadeh, 1975) é usualmente associada com 7 ± 2 conjuntos referenciais *fuzzy* (Pedrycz, 1995), o que implica $\eta \leq 9$, então pode-se concluir que é possível representar lingüisticamente uma variável *crisp* a através de um conjunto *fuzzy* P utilizando um número reduzido de elementos em comparação ao procedimento de fuzzificação e a respectiva discretização dos universos de discurso tipicamente adotada na literatura.

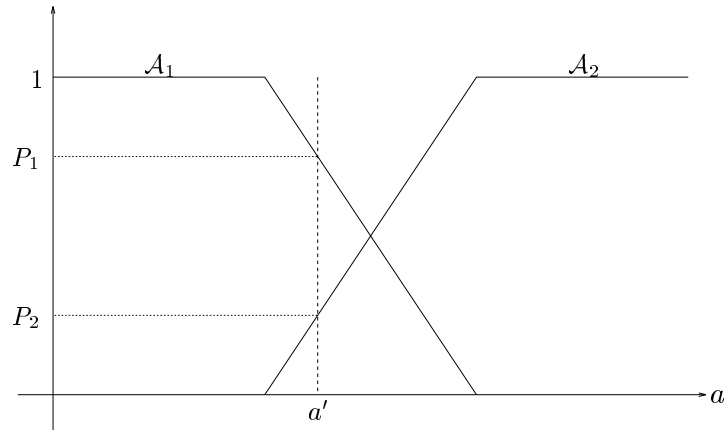


Figura A.1: Discretização *fuzzy* de um valor *crisp* a' utilizando dois conjuntos referenciais $\mathcal{A}_1(a)$ e $\mathcal{A}_2(a)$.

A.4 Estrutura de Cognição

A Estrutura de Cognição (*Frame of Cognition*) (Pedrycz, 1995; Pedrycz e Gomide, 1998) é definida como uma coleção de conjuntos referenciais *fuzzy* $\mathcal{A}_1, \dots, \mathcal{A}_\eta$, sobre um mesmo universo de discurso \mathbf{A} de uma variável *crisp* “a”, satisfazendo as condições necessárias para uma descrição lingüisticamente clara e consistente do comportamento qualitativo dessa variável. Essas condições são apresentadas abaixo:

- **Distinção:** Os rótulos (*labels*) lingüísticos associados aos conjuntos referenciais *fuzzy* devem ser semanticamente claros. Para tanto, os conjuntos referenciais devem ser suficientemente disjuntos (Oliveira, 1995) de forma a representar claramente uma região específica do universo de discurso. A distinção em uma estrutura de cognição está relacionada com a especificidade dos conjuntos *fuzzy* que a compõe. Essa característica pode ser avaliada através de uma medida de especificidade (*specificity measure*) (Pedrycz, 1993). Em uma mesma distribuição dos conjuntos *fuzzy* ao longo do universo de discurso, quanto mais específicos são esses conjuntos mais disjunta é a estrutura de cognição.
- **Convexidade:** A convexidade é uma característica desejada dado que conjuntos *fuzzy* convexos (Zadeh, 1965) possuem notoriamente uma interpretabilidade superior àqueles não convexos. Deve-se ressaltar, no entanto, que em alguns casos específicos essa característica pode não ser verificada. Esse é o caso, por exemplo, de conjuntos gerados a partir da aplicação, em outros conjuntos, do transformador lingüístico “NÃO” (*Hedge NOT*) (Zadeh, 1973; Tanscheit, 1992).
- **Normalização:** Os conjuntos referenciais *fuzzy* devem ser normais, ou seja, para cada conjunto ao menos um elemento do universo de discurso deve possuir uma correspondência máxima (grau de pertinência unitário) a ele.

- **Número justificável de elementos:** A quantidade de conjuntos referenciais *fuzzy* deve ser compatível com a capacidade de compreensão e interpretação humana. Estudos apontam a restrição $\eta \leq 9$ como uma condição limitante adequada (Pedrycz, 1995).
- **Cobertura do universo de discurso:** O universo de discurso \mathbf{A} deve estar totalmente representado (“coberto”) pelos conjuntos referenciais *fuzzy*. Formalmente, pode-se definir a cobertura do universo de discurso através da condição de completude apresentada abaixo:

$$\forall a \in \mathbf{A} : \exists i : 1 \leq i \leq \eta : \mathcal{A}_i(a) > 0 \quad (\text{A.36})$$

Essa condição implica que qualquer valor $a \in \mathbf{A}$ possui um grau de pertinência diferente de zero com relação a pelo menos um conjunto referencial. Não obstante, pode-se redefinir (A.36) de forma mais rigorosa como:

$$\forall a \in \mathbf{A} : \exists i : 1 \leq i \leq \eta : \mathcal{A}_i(a) > \xi \quad (\text{A.37})$$

onde $\xi \in [0, 1)$ é um *limiar de cobertura*.

Apêndice B

Processo Industrial para Produção de Etanol

Esse apêndice descreve a planta para produção de etanol em escala industrial utilizada como aplicação para os algoritmos de modelagem e controle discutidos nos capítulos 5 e 6. Maiores detalhes podem ser encontrados em (Meleiro, 2002).

B.1 Introdução

Os processos fermentativos alcoólicos no Brasil surgiram com a fabricação do aguardente de cana e foram mais tarde aplicados à fabricação de álcool a partir do melaço proveniente da fabricação de açúcar. Na década de 80, a partir dos incentivos do programa Pró-Álcool do governo federal, cresceram as pesquisas enfocando o aumento da produtividade (rendimento) desses processos. Atualmente as pesquisas concentram-se na otimização da operação dos processos em regime contínuo. A motivação e a importância dessas pesquisas advém principalmente da complexidade desses processos cuja principal força motriz são seres vivos, isto é, microrganismos que são muito sensíveis a quaisquer variações ambientais em seu meio, tais como temperatura, substrato (alimento), pH, entre outros. Devido a essas características, esses processos constituem ainda nos dias de hoje tema relevante e atual de pesquisa (Meleiro e Maciel Filho, 2000).

No presente trabalho considera-se particularmente uma planta industrial para produção de álcool etílico (etanol). Devido às dificuldades em trabalhar diretamente com uma planta em operação na indústria, como os altos custos envolvidos em eventuais interrupções na operação para a realização de experimentos, optou-se por trabalhar com um simulador já desenvolvido e validado com base em uma planta real operando em uma usina de açúcar e álcool. Esse simulador foi desenvolvido por Andrietta (Andrietta, 1994; Andrietta e Maugeri, 1994), que modelou o conjunto de reações bioquímicas do processo através de um conjunto de equações diferenciais ordinárias não lineares e otimizou uma região de operação, implementada posteriormente no processo real, de forma a levar o mesmo a valores aceitáveis de produtividade sem compro-

meter sua viabilidade econômica e operacional. Dentro dessa região pré-otimizada de operação, deve-se então aplicar controladores atuando nas variáveis de manipulação do processo de forma a otimizar sua produção em tempo real (regime contínuo de operação) frente às perturbações às quais este está sujeito. Controladores de particular interesse para esse tipo de aplicação, que caracteriza-se por dinâmicas lentas e condições restritas de operação, são os Controladores Preditivos (*Model Predictive Controllers - MPC*) (Clarke, 1994), conforme discutido no capítulo 6.

B.2 Planta para Produção de Etanol

O processo fermentativo para produção de etanol em estudo é ilustrado na figura B.1 (Meleiro et al., 2001). Esse processo consiste de quatro reatores (tanques) ligados em série onde a entrada é um caldo de cana de açúcar previamente tratado consistindo de um xarope açucarado rico em glicose denominado industrialmente de Mosto. Esse mosto é, seqüencialmente ao longo dos reatores, convertido em álcool (etanol) através do processo fermentativo realizado pela levedura *Saccharomyces cerevisae*, basicamente uma colônia de microrganismos denominada genericamente de células de levedura. Junto a esse mosto adiciona-se um reciclo de células que são centrifugadas a partir do produto final bruto (saída do quarto tanque) e previamente submetidas a um tratamento ácido e de diluição.

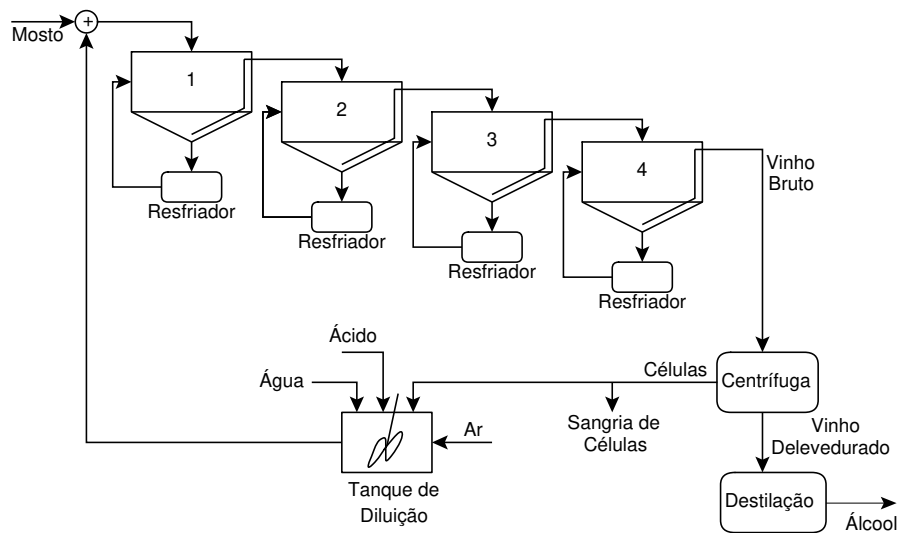


Figura B.1: Figura esquemática da planta industrial para produção de etanol.

O produto final do último tanque é denominado Vinho Bruto. Esse produto carrega entre 30 a 45g de células por litro e, por esta razão, passa por um processo mecânico de centrifugação para a separação dessas células. Uma vez que a reprodução dos microrganismos é muito sensível às condições do meio em que estão inseridos, as células são

recicladas para equilibrar a população. Antes do tratamento para a reciclagem no tanque de diluição as células mortas são retiradas em um processo denominado sangria de células. O procedimento de reciclagem é importante pois a geração de novas colônias, caso necessária, é um processo caro e dispendioso.

O conjunto de centrífugas separa o vinho bruto, que consiste de água, CO₂, açúcar, células (30-45kg/m³) e álcool, em duas fases de produtos. A fase pesada contém a grande maioria das células, em uma concentração de 160-200kg/m³. Já a fase leve contém no máximo 3kg/m³ de células e entre 9 a 12% de álcool. Esta última é finalmente levada ao processo de destilação que separa o álcool da água.

Na figura B.1 observa-se a presença de resfriadores em cada um dos tanques. Esses resfriadores, cujo princípio é a troca de calor a base de água, constituem malhas de controle independentes da malha principal de controle da produção (não representada na figura). Essas malhas independentes objetivam a manutenção da temperatura nos tanques em um nível constante (*set-point*) ideal para o processo de fermentação. Esse *set-point* foi otimizado por Andrietta (1994) de forma a maximizar a eficiência das reações (conversão) na planta industrial considerada por ele, e é assumido no modelo cinético daquela planta cujo simulador é utilizado no presente trabalho. Esse simulador também considera outras simplificações plausíveis de aparatos não ilustrados na figura. Um deles é uma malha de controle independente para a regulação dos volumes de matéria nos tanques, representada no simulador pela condição de vazões iguais para entrada e saída de cada tanque. Outra simplificação refere-se às válvulas de controle de vazão do mosto e do reciclo. Esses dispositivos podem, por simplicidade e sem perda de precisão, terem suas dinâmicas desprezadas por serem extremamente mais rápidas do que as demais dinâmicas do processo. Uma outra simplificação diz respeito à hipótese de tanques perfeitamente agitados (Andrietta, 1994), através da qual supõe-se que as reações ocorrem de forma homogênea no interior dos tanques. Pode-se demonstrar que essa é uma boa aproximação no que diz respeito ao modelo cinético validado por Andrietta.

A planta industrial para produção de etanol é um processo altamente não linear. As principais não linearidades advêm do comportamento biológico dos microrganismos. Aumentando a vazão do mosto, por exemplo, aumenta-se a concentração de açúcar¹ nos tanques, que é o alimento das células. Nessa condição a produção de etanol tende a crescer a partir da conversão biológica desse açúcar. No entanto, uma eventual quantidade de açúcar que supere a capacidade de processamento dos microrganismos (superalimentação) não será convertida em etanol aparecendo assim no produto final, o que significa uma queda de eficiência da conversão. A superalimentação pode ainda causar um efeito inibitório, onde não apenas a conversão mas também a reprodução dos microrganismos é inibida, sendo refletida diretamente na produção do álcool. Esse efeito inibitório pode também ser causado pelo excesso de etanol, que pode ainda causar a morte de células. A morte de células, por sua vez, pode também ser causada por uma subnutrição da colônia. Todos esses fatores influenciam as dinâmicas e a

¹Tecnicamente denominado ART (Açúcares Redutores Totais).

eficiência do processo fermentativo. Maiores detalhes desse processo podem ser encontrados em (Andrietta, 1994) e um conjunto de ensaios ilustrando em detalhes o seu comportamento dinâmico é apresentado em (Dechechi, 1998).

B.2.1 Variáveis Manipuladas, de Perturbação e de Saída

Dada a pré-otimização da planta realizada por Andrietta, discutida anteriormente, tem-se que as variáveis de entrada do processo, manipuladas ou de perturbação, são:

- *Vazão de Alimentação do Mosto* (F_a [m³/h]): Essa é a principal variável de entrada manipulada. O universo de discurso para essa variável é o intervalo [50, 150]. Esse intervalo é conservativo no sentido de garantir a viabilidade econômica e operacional da planta, já abrangendo com alguma folga as limitações relativas aos acionamentos de válvulas e volumes/bombeamento dos tanques, bem como as restrições de alimentação dos microrganismos.
- *Taxa de Reciclo* (t_r [adimensional]): Essa variável relaciona a vazão de alimentação do mosto F_a com a vazão do reciclo de células F_r e, conseqüentemente, com a vazão real de entrada do primeiro tanque (F_0), que é dada pela soma das duas anteriores (ver figura B.1), como segue:

$$F_0 = F_a + F_r = \frac{F_a}{(1 - t_r)} \quad (\text{B.1})$$

Dessa forma, uma taxa de 0.3, por exemplo, implica $F_a = 0.7F_0$ e $F_r = 0.3F_0$, isto é, F_a e F_r representam respectivamente 70% e 30% da vazão F_0 . O valor $t_r = 0.3$ é precisamente o valor nominal pré-otimizado para a operação industrial da planta. Em um sistema de controle multivariável, no entanto, essa variável pode vir a ser utilizada como uma entrada manipulada pelo controlador.

- *Concentração de Açúcar (ART) no Mosto* (S_0 [kg/m³]): O valor nominal dessa variável nas condições reais de operação da planta é 180kg/m³ mas, como essa concentração depende da cana utilizada e do pré-tratamento do mosto, é importante considerar possíveis perturbações de até +/- 5% em torno desse valor. Nesse caso essa variável torna-se uma entrada de perturbação aproximadamente no intervalo [170, 190].

As possíveis variáveis do processo a serem consideradas como saídas de interesse são:

- *Concentração de Etanol na Saída do Quarto Tanque* (P_4 [kg/m³]).
- *Concentração de ART na Saída do Quarto Tanque* (S_4 [kg/m³]).

- *Concentração de Células na Saída do Quarto Tanque (X_4 [kg/m³]).*

onde o produto de saída do quarto tanque, referenciado pelo índice 4 nas variáveis, é o vinho bruto (ver figura B.1).

Apêndice C

Algoritmo Genético

Esse apêndice descreve em detalhes o algoritmo genético utilizado na metodologia híbrida para a estimação de modelos hierárquicos FBO discutida no capítulo 5.

C.1 Introdução

Algoritmos Genéticos (GAs) são algoritmos de busca estocástica compostos de operações inspiradas no processo natural de evolução. Essas operações são aplicadas a um conjunto de variáveis codificadas, mais especificamente um vetor de atributos, obtido através de alguma forma de codificação das variáveis originais do problema a serem otimizadas. O conjunto codificado de variáveis é chamado de *cromossomo* e cada elemento do cromossomo (atributo) é denominado *gene*. Uma realização específica do cromossomo com relação aos valores (*alelos*) assumidos pelos genes é definida como um *indivíduo*.

As operações básicas de um GA são: *Seleção*, *Reprodução (Crossover)* e *Mutação*. Inicialmente, um conjunto de indivíduos representando a condição inicial para o problema de otimização é gerado aleatoriamente ou com base em algum procedimento de inicialização. Esses indivíduos constituem a “população inicial” $P(k = 0)$ do GA. Na iteração (*geração*) k do algoritmo cada indivíduo da população $P(k)$ é avaliado de acordo com um critério de desempenho denominado *fitness*. A partir de uma analogia com sistemas biológicos, o critério de *fitness* pode ser visto como uma medida de desempenho da representação fenotípica da informação genotípica codificada no cromossomo. Utilizando algum procedimento (usualmente estocástico) baseado no conceito Darwiniano da *sobrevivência do mais forte* (indivíduos com maiores valores de *fitness* possuem maiores probabilidades de serem selecionados), um conjunto de indivíduos da população é selecionado para produzir novos indivíduos através de *crossover*. Esses novos indivíduos (prole) são então sujeitos a mudanças aleatórias (mutações) em seus genes, usualmente com baixa probabilidade de ocorrência (assim como na natureza). Os indivíduos resultantes constituem ou são parte da nova população $P(k + 1)$. Para assegurar que essa nova população não contenha apenas soluções para o problema inferiores àquelas da

população anterior, os indivíduos de $P(k)$ com os melhores (maiores) valores de *fitness* podem ser incluídos diretamente em $P(k + 1)$. Esse procedimento é conhecido como *elitismo*. Um conjunto de indivíduos com o menor grau de similaridade (sob o ponto de vista fenotípico ou genotípico) com relação ao melhor indivíduo da população corrente $P(k)$ pode também ser incluído diretamente em $P(k + 1)$ para garantir uma representatividade adequada da população dentro do domínio de busca. Esse procedimento é conhecido como *diversidade*.

O algoritmo é executado por um número fixo (pré-especificado) de gerações ou até que algum critério de parada seja satisfeito. O melhor indivíduo da última geração é tomado como a solução do GA para o problema de otimização. Na prática, devido aos seus elevados requerimentos computacionais, GAs são usualmente utilizados como parte de esquemas híbridos nos quais são responsáveis por encontrar soluções aproximadas para serem refinadas por métodos tradicionais baseados em gradiente.

C.2 Algoritmo

O GA descrito a seguir utiliza um método de *codificação direta* para codificar os parâmetros desconhecidos do modelo na forma de um cromossomo para o algoritmo. Nessa estratégia, freqüentemente adotada quando existe um número elevado de parâmetros a serem otimizados, o cromossomo consiste simplesmente de um vetor onde cada elemento (gene) representa um desses parâmetros (Michalewicz, 1996). O critério de *fitness* é selecionado como $F = 1/\text{EQM}$, onde EQM denota o erro quadrático médio entre um conjunto de dados de saída do sistema a ser identificado e as respectivas estimativas do modelo a ser otimizado. Logo, quanto mais precisa for uma dada realização do modelo (indivíduo), mais elevado será o valor de *fitness* associado a ela. Finalmente, o critério de parada é selecionado simplesmente como um número fixo de gerações a serem executadas.

O algoritmo iterativo é descrito a seguir:

1. **(Configuração)** Selecione os parâmetros do algoritmo, ou seja, a probabilidade de reprodução (P_r), a probabilidade de mutação (P_m), o número de indivíduos para elitismo (N_e), o número de indivíduos para diversidade (N_d), o número total de indivíduos na população (N_I) e a quantidade final de gerações (N_g).
2. **(Inicialização)** Faça $k = 0$ e gere uma população inicial $P(0)$ com N_I indivíduos.
3. **(Condição de Parada)** Pare se $k > N_g$. Caso contrário, vá para o próximo passo.
4. **(Fitness)** Calcule o *fitness* $F(I_j)$ de cada indivíduo I_j ($j = 1, \dots, N_I$) da população corrente $P(k)$.

5. **(Probabilidade de Seleção)** Calcule a probabilidade de seleção $P_s(I_j)$ de cada indivíduo I_j ($j = 1, \dots, N_I$) da população corrente como $P_s(I_j) = F(I_j) / \sum_{l=1}^{N_I} F(I_l)$. Note que $P_s(I_j) \in [0, 1]$ é proporcional a $F(I_j)$.
6. **(Elitismo e Diversidade)** Tome aqueles N_e indivíduos com os maiores valores de *fitness* e inclua uma cópia de cada um deles em uma população auxiliar $\bar{P}(k)$. Faça o mesmo com aqueles N_d indivíduos menos similares ao indivíduo de maior *fitness*. Utilize a distância Euclidiana entre dois vetores como medida de similaridade.
7. **(Seleção e Reprodução)** Execute os subpassos abaixo até que $N_I - N_e - N_d$ novos indivíduos sejam obtidos para completar a população auxiliar $\bar{P}(k)$:
 - **(Seleção)** Retire dois indivíduos da população corrente $P(k)$ de acordo com as probabilidades de seleção calculadas no passo 5. O bem conhecido método da roleta (*roulette wheel method*) (Mitchell, 1998), por exemplo, pode ser utilizado para esta finalidade.
 - **(Reprodução)** Selecione aleatoriamente um valor $\xi \in [0, 1]$. Se $\xi > P_r$, então simplesmente inclua uma cópia de cada indivíduo selecionado no passo anterior na população auxiliar $\bar{P}(k)$. Caso contrário, aplique *crossover* a esses indivíduos (pais) e inclua os dois indivíduos resultantes (prole) em $\bar{P}(k)$. A operação de *crossover* (de ponto simples) funciona da seguinte forma: Selecione aleatoriamente uma posição do cromossomo. Tome essa posição como referência e construa os dois novos indivíduos utilizando a parte esquerda de um dos pais e a parte direita do outro, e vice-versa.
 - Retorne os dois indivíduos originais (pais) para a população corrente $P(k)$.
8. **(Mutação)** Aplique mutação a cada indivíduo de $\bar{P}(k)$, exceto ao melhor indivíduo, como segue: Selecione aleatoriamente um valor $\xi \in [0, 1]$ para cada gene do indivíduo. Se $\xi \leq P_m$, então adicione um valor δ ao valor corrente daquele gene. Caso contrário, mantenha o gene inalterado. Selecione δ aleatoriamente, com distribuição de probabilidade Gaussiana de média zero e variância igual à variância do valor assumido pelo gene em questão nos diferentes indivíduos de $P(k)$ ¹. Um limitante inferior para a variância (e.g. 1% da respectiva variância associada à população inicial) deve ser utilizado para evitar a convergência precoce do algoritmo e problemas de ordem numérica.
9. **(Nova Geração)** Faça $P(k+1) = \bar{P}(k)$, $k = k + 1$ e volte ao passo 3.

¹Esse procedimento representa um dispositivo de auto-ajuste do algoritmo implementado de forma independente para cada gene (Fogel, 2000).

Apêndice D

Provas de Lemas e Teoremas

Esse apêndice apresenta as provas dos lemas e teoremas utilizados para o desenvolvimento ótimo de modelos de Volterra em funções ortonormais de Laguerre, discutido na seção 5.12 (capítulo 5).

D.1 Prova do Teorema 5.1

A prova será dividida em itens, como segue:

Item A

Da definição de norma 2 tem-se:

$$\begin{aligned} \|h_m\|^2 &\triangleq \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, \dots, k_m)^2 \\ &= \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, \dots, k_m) \left(\sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m} \prod_{j=1}^m \phi_{m, i_j}(k_j) \right) \\ &= \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} \left(\sum_{i'_1=1}^{\infty} \cdots \sum_{i'_m=1}^{\infty} \alpha_{i'_1, \dots, i'_m} \prod_{j=1}^m \phi_{m, i'_j}(k_j) \right) \cdot \\ &\quad \cdot \left(\sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m} \prod_{j=1}^m \phi_{m, i_j}(k_j) \right) \end{aligned} \tag{D.1}$$

Logo, tem-se:

$$\|h_m\|^2 = \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m} \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} \phi_{m, i_1}(k_1) \cdots \phi_{m, i_m}(k_m) \cdot \left(\alpha_{i_1, \dots, i_m} \prod_{j=1}^m \phi_{m, i_j}(k_j) + \sum_{\substack{i'_1=1 \\ i'_1 \neq i_1}}^{\infty} \cdots \sum_{\substack{i'_m=1 \\ i'_m \neq i_m}}^{\infty} \alpha_{i'_1, \dots, i'_m} \prod_{j=1}^m \phi_{m, i'_j}(k_j) \right) \quad (\text{D.2})$$

e utilizando a propriedade de ortonormalidade das funções, isto é:

$$\sum_{\kappa=0}^{\infty} \phi_{m, i}(\kappa) \phi_{m, j}(\kappa) = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases} \quad (\text{D.3})$$

tem-se de (D.2) que:

$$\|h_m\|^2 = \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m}^2 \quad (\text{D.4})$$

Item B

Os sinais de Laguerre $\phi_{(\cdot)}$ no domínio do tempo possuem a seguinte propriedade (Fu e Dumont, 1993):

$$p_m k_j \phi_{m, i_j}(k_j + 1) - (1 + p_m^2) k_j \phi_{m, i_j}(k_j) + p_m (k_j - 1) \phi_{m, i_j}(k_j - 1) = (p_m^2 i_j - p_m^2 - i_j) \phi_{m, i_j}(k_j) \quad (\text{D.5})$$

o que leva à igualdade:

$$p_m k_1 \phi_{m, i_1}(k_1 + 1) \prod_{j=2}^m \phi_{m, i_j}(k_j) - (1 + p_m^2) k_1 \phi_{m, i_1}(k_1) \prod_{j=2}^m \phi_{m, i_j}(k_j) + p_m (k_1 - 1) \phi_{m, i_1}(k_1 - 1) \prod_{j=2}^m \phi_{m, i_j}(k_j) = (p_m^2 i_1 - p_m^2 - i_1) \phi_{m, i_1}(k_1) \prod_{j=2}^m \phi_{m, i_j}(k_j) \quad (\text{D.6})$$

que será utilizada nos desenvolvimentos a seguir. Partindo então de:

$$\begin{aligned} \mathcal{U}_1 \triangleq & \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, \dots, k_m) \left(p_m k_1 h_m(k_1 + 1, k_2, \dots, k_m) - \right. \\ & \left. (1 + p_m^2) k_1 h_m(k_1, \dots, k_m) + p_m (k_1 - 1) h_m(k_1 - 1, k_2, \dots, k_m) \right) \end{aligned} \quad (\text{D.7})$$

e utilizando (5.35), obtém-se:

$$\begin{aligned} \mathcal{U}_1 = & \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} \left(p_m k_1 \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m} \phi_{m, i_1}(k_1 + 1) \prod_{j=2}^m \phi_{m, i_j}(k_j) \right. \\ & - (1 + p_m^2) k_1 \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m} \phi_{m, i_1}(k_1) \prod_{j=2}^m \phi_{m, i_j}(k_j) \\ & \left. + p_m (k_1 - 1) \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m} \phi_{m, i_1}(k_1 - 1) \prod_{j=2}^m \phi_{m, i_j}(k_j) \right) h_m(k_1, \dots, k_m) \end{aligned} \quad (\text{D.8})$$

Das equações (D.6) e (D.8) tem-se que:

$$\begin{aligned} \mathcal{U}_1 = & \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} \left(\sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m} (p_m^2 i_1 - p_m^2 - i_1) \cdot \right. \\ & \left. \phi_{m, i_1}(k_1) \prod_{j=2}^m \phi_{m, i_j}(k_j) \right) h_m(k_1, \dots, k_m) \end{aligned} \quad (\text{D.9})$$

e substituindo (5.35) em (D.9) produz-se:

$$\begin{aligned} \mathcal{U}_1 = & \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m} (p_m^2 i_1 - p_m^2 - i_1) \cdot \\ & \cdot \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} \left(\prod_{j=1}^m \phi_{m, i_j}(k_j) \right) \sum_{i'_1=1}^{\infty} \cdots \sum_{i'_m=1}^{\infty} \alpha_{i'_1, \dots, i'_m} \prod_{j=1}^m \phi_{m, i'_j}(k_j) \end{aligned} \quad (\text{D.10})$$

Manipulando algebricamente a equação acima de forma similar àquela realizada no Item A, fazendo uso da propriedade de ortonormalidade das funções em (D.3), obtém-se:

$$\begin{aligned}
\mathcal{U}_1 &= \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m}^2 (p_m^2 i_1 - p_m^2 - i_1) \\
&= \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m}^2 (i_1(p_m^2 - 1) - p_m^2)
\end{aligned} \tag{D.11}$$

Da segunda condição do teorema, que implica $h_m(0, k_2, \dots, k_m) = 0$, e das equações (D.7) e (D.11), tem-se:

$$\begin{aligned}
\mathcal{U}_1 &= \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1 + 1, k_2, \dots, k_m) \left(p_m (k_1 + 1) h_m(k_1 + 2, k_2, \dots, k_m) \right. \\
&\quad \left. - (1 + p_m^2) (k_1 + 1) h_m(k_1 + 1, k_2, \dots, k_m) + p_m k_1 h_m(k_1, \dots, k_m) \right) \\
&= \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m}^2 (i_1(p_m^2 - 1) - p_m^2)
\end{aligned} \tag{D.12}$$

Analogamente, para $\forall l \in \{1, \dots, m\}$ tem-se:

$$\begin{aligned}
\mathcal{U}_l &= \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} \left(p_m (k_l + 1) h_m(k_1, \dots, k_l + 2, \dots, k_m) \right. \\
&\quad \left. - (1 + p_m^2) (k_l + 1) h_m(k_1, \dots, k_l + 1, \dots, k_m) \right. \\
&\quad \left. + p_m k_l h_m(k_1, \dots, k_l, \dots, k_m) \right) h_m(k_1, \dots, k_l + 1, \dots, k_m) \\
&= \sum_{i_1=1}^{\infty} \cdots \sum_{i_l=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m}^2 (i_l(p_m^2 - 1) - p_m^2)
\end{aligned} \tag{D.13}$$

Item C

Da definição de $S_{2,l}$ em (5.45) pode-se escrever:

$$\begin{aligned}
p_m S_{2,l} \|h_m\|^2 &= \\
&= p_m \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l [\Delta_l h_m(k_1, \dots, k_l, \dots, k_m)]^2 \\
&= p_m \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=-1}^{\infty} \cdots \sum_{k_m=0}^{\infty} (k_l + 1) [\Delta_l h_m(k_1, \dots, k_l + 1, \dots, k_m)]^2
\end{aligned} \tag{D.14}$$

Fazendo $k'_l = k_l + 1$, o que implica $k_l = k'_l - 1$, rescreve-se a equação acima como:

$$\begin{aligned}
p_m S_{2,l} \|h_m\|^2 &= \\
&= p_m \sum_{k_1=0}^{\infty} \cdots \sum_{k'_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} q(k_1, \dots, k'_l + 1, \dots, k_m) \Delta_l h_m(k_1, \dots, k'_l, \dots, k_m)
\end{aligned} \tag{D.15}$$

onde

$$\begin{aligned}
q(k_1, \dots, k'_l, \dots, k_m) &\triangleq (k'_l - 1) \Delta_l h_m(k_1, \dots, k'_l - 1, \dots, k_m) \\
&= k_l \Delta_l h_m(k_1, \dots, k_l, \dots, k_m)
\end{aligned} \tag{D.16}$$

Não é difícil demonstrar que para duas seqüências $a(\kappa)$ e $b(\kappa)$, com $\kappa \in \mathbb{Z}_+$, vale a seguinte igualdade (Fu e Dumont, 1993):

$$\sum_{\kappa=0}^{\infty} a(\kappa + 1) \Delta b(\kappa) = b(\infty) a(\infty) - b(0) a(0) - \sum_{\kappa=0}^{\infty} b(\kappa) \Delta a(\kappa) \tag{D.17}$$

onde o operador Δ é definido como $\Delta b(\kappa) = b(\kappa + 1) - b(\kappa)$. Lembrando que $h_m(k_1, \dots, k'_l, \dots, k_m) \rightarrow 0$ para $k'_l \rightarrow \infty$, o que é uma implicação direta da primeira condição do teorema (Kreysig, 1993), e que $h_m(k_1, \dots, k'_{l-1}, 0, k'_{l+1}, \dots, k_m) = 0$ (segunda condição do teorema), tem-se de (D.15) e (D.17) que:

$$\begin{aligned}
p_m S_{2,l} \|h_m\|^2 &= \\
&= -p_m \sum_{k_1=0}^{\infty} \cdots \sum_{k'_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, \dots, k'_l, \dots, k_m) \Delta_l q(k_1, \dots, k'_l, \dots, k_m) \\
&= -p_m \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=-1}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, \dots, k_l + 1, \dots, k_m) \cdot \Delta_l [\Delta_l h_m(k_1, \dots, k_l, \dots, k_m) k_l]
\end{aligned} \tag{D.18}$$

Mas para $k_l = -1$ tem-se que $h_m(k_1, \dots, k_l + 1, \dots, k_m) = 0$, o que implica:

$$p_m S_{2,l} \|h_m\|^2 = -p_m \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, \dots, k_l + 1, \dots, k_m) \cdot \Delta_l \left[(h_m(k_1, \dots, k_l + 1, \dots, k_m) - h_m(k_1, \dots, k_l, \dots, k_m)) k_l \right] \quad (\text{D.19})$$

e após algumas manipulações algébricas elementares:

$$p_m S_{2,l} \|h_m\|^2 = - \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, \dots, k_l + 1, \dots, k_m) \cdot \left(p_m (k_l + 1) h_m(k_1, \dots, k_l + 2, \dots, k_m) - p_m (2k_l + 1) \cdot h_m(k_1, \dots, k_l + 1, \dots, k_m) + p_m k_l h_m(k_1, \dots, k_l, \dots, k_m) \right) \quad (\text{D.20})$$

A partir das equações (D.13) e (D.20) pode-se inferir:

$$p_m S_{2,l} \|h_m\|^2 = - \sum_{i_1=1}^{\infty} \cdots \sum_{i_l=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m}^2 (i_l (p_m^2 - 1) - p_m^2) - \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, \dots, k_l + 1, \dots, k_m)^2 \left[(1 - p_m)^2 (k_l + 1) + p_m \right] \quad (\text{D.21})$$

e como $h_m(k_1, \dots, k_{l-1}, 0, k_{l+1}, \dots, k_m) = 0$ tem-se:

$$p_m S_{2,l} \|h_m\|^2 = - \sum_{i_1=1}^{\infty} \cdots \sum_{i_l=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \alpha_{i_1, \dots, i_m}^2 (i_l (p_m^2 - 1) - p_m^2) - \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, \dots, k_l, \dots, k_m)^2 \left[(1 - p_m)^2 k_l + p_m \right] \quad (\text{D.22})$$

Utilizando então a equação (D.4) e a definição de $S_{1,l}$ em (5.44), rescreve-se (D.22) da seguinte forma:

$$p_m S_{2,l} \|h_m\|^2 = \sum_{i_1=1}^{\infty} \cdots \sum_{i_l=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \left[(p_m^2 + (1 - p_m^2)i_l - p_m) \alpha_{i_1, \dots, i_m}^2 \right] - (1 - p_m)^2 S_{1,l} \|h_m\|^2 \quad (\text{D.23})$$

que é válida para $\forall l \in \{1, \dots, m\}$.

Item D

Da equação (D.23) tem-se que:

$$\begin{aligned} p_m \left(\sum_{l=1}^m S_{2,l} \right) \|h_m\|^2 &= \sum_{l=1}^m \left(\sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \left[(p_m^2 + (1 - p_m^2)i_l - p_m) \alpha_{i_1, \dots, i_m}^2 \right] \right. \\ &\quad \left. - (1 - p_m)^2 S_{1,l} \|h_m\|^2 \right) \\ &= \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \left[\left(m p_m^2 - m p_m + (1 - p_m^2) \sum_{l=1}^m i_l \right) \alpha_{i_1, \dots, i_m}^2 \right] \\ &\quad - (1 - p_m)^2 \left(\sum_{l=1}^m S_{1,l} \right) \|h_m\|^2 \end{aligned} \quad (\text{D.24})$$

Utilizando então a equação (D.4) e a definição de $Q_{j,m}$ em (5.43), pode-se rescrever (D.24) após algumas manipulações algébricas como:

$$\begin{aligned} p_m m Q_{2,m} \|h_m\|^2 + (1 - p_m)^2 m Q_{1,m} \|h_m\|^2 &= \\ &= m (p_m^2 - p_m) \|h_m\|^2 + (1 - p_m^2) \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \left(\sum_{l=1}^m i_l \right) \alpha_{i_1, \dots, i_m}^2 \end{aligned} \quad (\text{D.25})$$

e portanto:

$$\begin{aligned} J_m &\triangleq \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} (i_1 + \cdots + i_m) \alpha_{i_1, \dots, i_m}^2 \\ &= \left(\frac{p_m Q_{2,m} + (1 - p_m)^2 Q_{1,m} + p_m - p_m^2}{1 - p_m^2} \right) m \|h_m\|^2 \end{aligned} \quad (\text{D.26})$$

■

D.2 Prova do Lema 5.1

A partir da definição de $Q_{1,m}$ em (5.43) tem-se que:

$$Q_{1,m} - 1 = \left(\frac{1}{m} \sum_{l=1}^m S_{1,l} \right) - 1 = \frac{1}{m} \sum_{l=1}^m (S_{1,l} - 1) \quad (\text{D.27})$$

sendo o termo $S_{1,l} - 1$ calculado a partir da definição de $S_{1,l}$ em (5.44) como:

$$\begin{aligned} S_{1,l} - 1 &= \frac{1}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l h_m(k_1, \dots, k_m)^2 - \frac{\|h_m\|^2}{\|h_m\|^2} \\ &= \frac{1}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} (k_l - 1) h_m(k_1, \dots, k_m)^2 \end{aligned} \quad (\text{D.28})$$

Dado que $h_m(k_1, \dots, k_m) = 0$ para $\forall k_j = 0$ com $j \in \{1, \dots, m\}$ (segunda condição do teorema 5.1), tem-se que a equação acima pode ser rescrita como:

$$S_{1,l} - 1 = \frac{1}{\|h_m\|^2} \sum_{k_1=1}^{\infty} \cdots \sum_{k_m=1}^{\infty} (k_l - 1) h_m(k_1, \dots, k_m)^2 \quad (\text{D.29})$$

que não pode ser negativa ($S_{1,l} - 1 \geq 0$). Logo, tem-se de (D.27) que:

$$Q_{1,m} - 1 \geq 0 \quad (\text{D.30})$$

■

D.3 Prova do Lema 5.2

Através de manipulações algébricas elementares pode-se verificar que:

$$\begin{aligned} \frac{1}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l \left(h_m(k_1, \dots, k_l, \dots, k_m) + \right. \\ \left. + h_m(k_1, \dots, k_l + 1, \dots, k_m) \right)^2 = S_{2,l} + \Gamma \end{aligned} \quad (\text{D.31})$$

sendo o termo $S_{2,l}$ definido em (5.45) e Γ dado por:

$$\Gamma = \frac{4}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l h_m(k_1, \dots, k_l, \dots, k_m) \cdot h_m(k_1, \dots, k_l + 1, \dots, k_m) \quad (\text{D.32})$$

A equação acima pode ser rescrita como:

$$\Gamma = \frac{4}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l h_m(k_1, \dots, k_l + 1, \dots, k_m) \cdot \left(h_m(k_1, \dots, k_l + 1, \dots, k_m) - \Delta_l h_m(k_1, \dots, k_l, \dots, k_m) \right) \quad (\text{D.33})$$

que implica:

$$\Gamma = \frac{4}{\|h_m\|^2} \left[\sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l h_m(k_1, \dots, k_l + 1, \dots, k_m)^2 - \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l h_m(k_1, \dots, k_l + 1, \dots, k_m) \Delta_l h_m(k_1, \dots, k_l, \dots, k_m) \right] \quad (\text{D.34})$$

ou equivalentemente:

$$\Gamma = \frac{4}{\|h_m\|^2} \left[\sum_{k_1=0}^{\infty} \cdots \sum_{k_l=1}^{\infty} \cdots \sum_{k_m=0}^{\infty} (k_l - 1) h_m(k_1, \dots, k_l, \dots, k_m)^2 \right] - \Phi \quad (\text{D.35})$$

onde

$$\Phi = \frac{4}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l h_m(k_1, \dots, k_l + 1, \dots, k_m) \cdot \Delta_l h_m(k_1, \dots, k_l, \dots, k_m) \quad (\text{D.36})$$

Dado que $h_m(k_1, \dots, k_m) = 0$ para $\forall k_j = 0$ com $j \in \{1, \dots, m\}$ (segunda condição do teorema 5.1), tem-se que a equação (D.35) pode ser rescrita como:

$$\begin{aligned}
\Gamma &= \left. \frac{4}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l h_m(k_1, \dots, k_l, \dots, k_m)^2 \right\} 4S_{1,l} \\
&- \left. \frac{4}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h_m(k_1, \dots, k_l, \dots, k_m)^2 \right\} -4 \frac{\|h_m\|^2}{\|h_m\|^2} \\
&- \Phi
\end{aligned} \tag{D.37}$$

ou seja,

$$\Gamma = 4S_{1,l} - 4 - \Phi \tag{D.38}$$

O termo Φ pode ser rescrito a partir de (D.36) como:

$$\begin{aligned}
\Phi &= \frac{4}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l \Delta_l h_m(k_1, \dots, k_l, \dots, k_m) \cdot \\
&\quad \cdot \left(h_m(k_1, \dots, k_l, \dots, k_m) + \Delta_l h_m(k_1, \dots, k_l, \dots, k_m) \right)
\end{aligned} \tag{D.39}$$

que equivale a:

$$\begin{aligned}
\Phi &= \left. \frac{4}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l h_m(k_1, \dots, k_l, \dots, k_m) \cdot \right. \\
&\quad \left. \cdot h_m(k_1, \dots, k_l + 1, \dots, k_m) \right\} \Gamma \\
&- \left. \frac{4}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l h_m(k_1, \dots, k_l, \dots, k_m)^2 \right\} -4S_{1,l} \\
&+ \left. \frac{4}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l \left(\Delta_l h_m(k_1, \dots, k_l, \dots, k_m) \right)^2 \right\} 4S_{2,l}
\end{aligned} \tag{D.40}$$

ou seja,

$$\Phi = \Gamma - 4S_{1,l} + 4S_{2,l} \tag{D.41}$$

sendo que Γ foi obtido a partir da própria definição em (D.32). Das equações (D.38) e (D.41) tem-se então:

$$\Gamma = 4S_{1,l} - 2S_{2,l} - 2 \quad (\text{D.42})$$

que em (D.31) produz:

$$\frac{1}{\|h_m\|^2} \sum_{k_1=0}^{\infty} \cdots \sum_{k_l=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} k_l \left(h_m(k_1, \dots, k_l, \dots, k_m) + h_m(k_1, \dots, k_l + 1, \dots, k_m) \right)^2 = 4S_{1,l} - S_{2,l} - 2 \quad (\text{D.43})$$

A equação (D.43) claramente implica $4S_{1,l} - S_{2,l} - 2 \geq 0$, mas como o kernel h_m é por hipótese não nulo e estável¹, tem-se necessariamente que:

$$4S_{1,l} - S_{2,l} - 2 > 0 \quad (\text{D.44})$$

Essa desigualdade implica:

$$\frac{1}{m} \sum_{l=1}^m (4S_{1,l} - S_{2,l} - 2) > 0 \quad (\text{D.45})$$

ou seja,

$$\frac{4}{m} \sum_{l=1}^m S_{1,l} - \frac{1}{m} \sum_{l=1}^m S_{2,l} - \frac{1}{m} \sum_{l=1}^m 2 > 0 \quad (\text{D.46})$$

que pela definição em (5.43) equivale a:

$$4Q_{1,m} - Q_{2,m} - 2 > 0 \quad (\text{D.47})$$

Tem-se ainda que, por definição, $Q_{2,m}$ não pode ser negativo. Mais além, como o kernel h_m é por hipótese não nulo e estável², esse termo é sempre positivo ($Q_{2,m} > 0$). Logo, a desigualdade (D.47) necessariamente implica:

$$4Q_{1,m}Q_{2,m} - Q_{2,m}^2 - 2Q_{2,m} > 0 \quad (\text{D.48})$$

■

¹Absolutamente somável, portanto convergente, o que descarta a possibilidade de um kernel oscilatório não amortecido na direção k_l .

²Hipótese que descarta a possibilidade de um kernel constante.

D.4 Prova do Teorema 5.3

Assim como para o teorema 5.1, a prova nesse caso também será organizada em itens:

Item A

A equação (D.30) implica a seguinte desigualdade:

$$\sum_{m=1}^{\eta} m (Q_{1,m} - 1) \|h_m\|^2 \geq 0 \quad (\text{D.49})$$

que pode ser facilmente rescrita utilizando a definição de ϖ em (5.63) como:

$$\left(\frac{1}{\varpi} \sum_{m=1}^{\eta} m Q_{1,m} \|h_m\|^2 \right) - 1 \geq 0 \quad (\text{D.50})$$

lembrando que $\varpi > 0$, pois os kernels são não nulos por hipótese. A desigualdade acima, por sua vez, pode ser rescrita utilizando a definição de \bar{Q}_1 em (5.64) como:

$$\bar{Q}_1 - 1 \geq 0 \quad (\text{D.51})$$

que implica diretamente:

$$2\bar{Q}_1 - 1 > 0 \quad (\text{D.52})$$

Item B

A equação (D.47) implica a seguinte desigualdade:

$$\frac{1}{\varpi} \sum_{m=1}^{\eta} m (4Q_{1,m} - Q_{2,m} - 2) \|h_m\|^2 > 0 \quad (\text{D.53})$$

que pode ser rescrita utilizando as definições de ϖ , \bar{Q}_1 e \bar{Q}_2 em (5.63), (5.64) e (5.65), respectivamente, como:

$$4\bar{Q}_1 - \bar{Q}_2 - 2 > 0 \quad (\text{D.54})$$

Por outro lado, mostrou-se como parte da prova do lema 5.2 que $Q_{2,m} > 0$ para $\forall m \in \{1, \dots, \eta\}$, o que implica:

$$\frac{1}{\varpi} \sum_{m=1}^{\eta} m Q_{2,m} \|h_m\|^2 = \bar{Q}_2 > 0 \quad (\text{D.55})$$

Logo, tem-se de (D.54) e (D.55) que:

$$4\bar{Q}_1\bar{Q}_2 - \bar{Q}_2^2 - 2\bar{Q}_2 > 0 \quad (\text{D.56})$$

Item C

Pode-se observar que o critério de custo em (5.62) é equivalente em forma àquele em (5.41), exceto pelo termo constante ϖ que pode ser ignorado para fins do problema de otimização correspondente em (5.61). Conseqüentemente, pode-se demonstrar o teorema 5.3 utilizando exatamente o mesmo procedimento adotado na prova do teorema 5.2, apenas substituindo os termos J'_m , p_m , $Q_{1,m}$ e $Q_{2,m}$ nesse último por J , p , \bar{Q}_1 e \bar{Q}_2 , respectivamente. Essa substituição direta, também válida para o lema 5.4, é possível porque as propriedades relevantes dos termos $Q_{1,m}$ e $Q_{2,m}$ utilizadas na prova são também verificadas para \bar{Q}_1 e \bar{Q}_2 . Especificamente, as relações apresentadas no lemas 5.1 e 5.2, assim como no corolário 5.1, são generalizadas para \bar{Q}_1 e \bar{Q}_2 em (D.51), (D.56) e (D.52), respectivamente. O lema 5.3 também permanece válido pois não é específico para qualquer dos termos mencionados acima. ■

Apêndice E

Controlador Preditivo Generalizado (GPC)

Esse apêndice descreve o algoritmo de controle utilizado nos desenvolvimentos do capítulo 6, isto é, o Controlador Preditivo Generalizado (GPC).

E.1 Introdução

Controladores baseados em modelo (*Model-Based Controllers*) que têm se mostrado de particular interesse para aplicações industriais, especialmente no contexto de processos químicos de grande porte, são os Controladores Preditivos (Soeterboek, 1992; Clarke, 1994; Camacho e Bordons, 1999). Esses controladores caracterizam-se por utilizar previsões do comportamento futuro do sistema controlado para realizar o cálculo da lei ótima de controle com base em um determinado critério de desempenho determinístico ou estocástico. As previsões são obtidas através de um modelo do sistema. A diferença entre as classes distintas de controladores preditivos está basicamente no tipo de modelo adotado. Nas seções seguintes considera-se um dos controladores preditivos mais difundidos na literatura: o Controlador Preditivo Generalizado (GPC) (Clarke et al., 1987a; Clarke et al., 1987b).

E.2 O Controlador GPC

Suponha que um sistema que se pretenda controlar, por simplicidade e sem perda de generalidade monovariável (SISO), possa ser descrito como:

$$A(q^{-1})y(t) = q^{-\tau}B(q^{-1})u(t-1) \quad (\text{E.1})$$

onde q é o operador deslocamento à frente, $u(t)$ e $y(t)$ são respectivamente a entrada e a saída do sistema no instante discreto de tempo t , $\tau \geq 0$ é o atraso de tempo e $A(q^{-1})$

e $B(q^{-1})$ são polinômios dados por

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_{N_a} q^{-N_a} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \cdots + b_{N_b} q^{-N_b} \end{aligned} \quad (\text{E.2})$$

O Controlador Preditivo Generalizado (GPC) utiliza um modelo do sistema, conhecido como CARIMA (*Controlled Auto-Regressive Integrating Moving-Average*), que é basicamente uma versão incremental (variacional) da representação dada pela equação (E.1), ou seja:

$$\tilde{A}(q^{-1})\hat{y}(t) = B(q^{-1})\Delta u(t-1) \quad (\text{E.3})$$

onde $\Delta = (1 - q^{-1})$ e $\tilde{A}(q^{-1}) = \Delta A(q^{-1}) = 1 + \tilde{a}_1 q^{-1} + \cdots + \tilde{a}_{N_{\tilde{a}}} q^{-N_{\tilde{a}}}$, sendo $\tilde{a}_j = a_j - a_{j-1}$ e $N_{\tilde{a}} = N_a + 1$. Note que, nesse modelo, qualquer atraso de tempo deve ser representado através de coeficientes nulos no polinômio $B(q^{-1})$.

O controlador preditivo calcula a seqüência dos sinais de controle futuros¹ $u(t)$, $u(t+1)$, \dots , $u(t+N_y-1)$ que minimiza o erro quadrático entre uma seqüência de saída desejada $w(t+1)$, \dots , $w(t+N_y)$ (referência de controle) e as saídas previstas $\hat{y}(t+1)$, \dots , $\hat{y}(t+N_y)$ no horizonte de previsão N_y . Dos sinais de controle calculados, somente o primeiro, ou seja $u(t)$, é aplicado ao sistema. No instante de tempo seguinte todo o procedimento é repetido, em uma estratégia conhecida como “Receding Horizon”. Portanto, a estratégia de controle é constituída das seguintes etapas:

- Cálculo da previsão da saída em um horizonte de tempo à frente, utilizando um modelo do sistema.
- Cálculo da lei de controle minimizando um critério dado por uma função do erro entre a saída prevista e a referência especificada.

O critério a ser minimizado é dado por:

$$J = \sum_{j=N_1}^{N_y} (\hat{y}(t+j|t) - w(t+j))^2 + \sum_{j=1}^{N_u} \lambda (\Delta u(t+j-1))^2 \quad (\text{E.4})$$

onde $N_1 \geq 1$ é o horizonte inicial de previsão, N_y é o horizonte de previsão ($N_y \geq N_1$), N_u é o horizonte de controle ($N_u \leq N_y$), $\lambda \geq 0$ é a ponderação do sinal de controle e $|t$ indica que as respectivas previsões da saída em instantes futuros são realizadas utilizando as informações disponíveis até o instante presente t . Discussões sobre a

¹ Assume-se que no instante t a saída do sistema $y(t)$ é conhecida e deseja-se determinar o valor de $u(t)$, portanto desconhecido.

influência dos parâmetros de sintonia N_1 , N_y , N_u e λ no desempenho do controlador podem ser encontradas em (Clarke et al., 1987a; Clarke et al., 1987b).

A minimização do critério J é realizada supondo que o sinal de controle a partir do instante $t + N_u - 1$ é mantido constante, ou seja, entre $t + N_u$ e $t + N_y - 1$ o sinal de controle é igual àquele em $t + N_u - 1$. Isso significa que os incrementos do sinal de controle após o instante $t + N_u - 1$, isto é $\Delta u(t + N_u)$, $\Delta u(t + N_u + 1)$, \dots , $\Delta u(t + N_y - 1)$, são nulos. Dessa forma, o problema de otimização é definido como:

$$\begin{aligned} \min_{\Delta \mathbf{u}} \quad & J \\ \text{s. a} \quad & \Delta u(k) = 0 \quad k \in [t + N_u, t + N_y - 1] \end{aligned} \quad (\text{E.5})$$

onde $\Delta \mathbf{u} = [\Delta u(t) \cdots \Delta u(t + N_u - 1)]^T$. Esse problema é usualmente resolvido a partir da aplicação de uma identidade polinomial (Eq. diofantina) para separar, em cada instante de amostragem, as parcelas livre e forçada da previsão futura da saída do sistema (Åström e Wittenmark, 1995). No entanto, o mesmo resultado pode ser obtido de forma mais simples, como mostrado nas seções seguintes.

E.2.1 Previsão via Cálculo Iterativo do Modelo

Lembrando que dispõem-se das informações do sistema até o instante presente t , particularmente no que diz respeito às medidas da saída y , as melhores previsões possíveis das saídas futuras utilizando o modelo em (E.3) são dadas por:

$$\begin{aligned} \hat{y}(t + 1 | t) &= -\tilde{a}_1 y(t) - \tilde{a}_2 y(t - 1) - \tilde{a}_3 y(t - 2) - \cdots - \tilde{a}_{N_{\tilde{a}}} y(t - N_{\tilde{a}} + 1) \\ &\quad + B \Delta u(t) \\ \hat{y}(t + 2 | t) &= -\tilde{a}_1 \hat{y}(t + 1 | t) - \tilde{a}_2 y(t) - \tilde{a}_3 y(t - 1) - \cdots - \tilde{a}_{N_{\tilde{a}}} y(t - N_{\tilde{a}} + 2) \\ &\quad + B \Delta u(t + 1) \\ \hat{y}(t + 3 | t) &= -\tilde{a}_1 \hat{y}(t + 2 | t) - \tilde{a}_2 \hat{y}(t + 1 | t) - \tilde{a}_3 y(t) - \cdots - \tilde{a}_{N_{\tilde{a}}} y(t - N_{\tilde{a}} + 3) \\ &\quad + B \Delta u(t + 2) \\ &\quad \vdots \end{aligned} \quad (\text{E.6})$$

Das equações acima pode-se verificar que as previsões da saída após o instante t podem ser descritas em função dos incrementos presente e futuros do sinal de controle (desconhecidos) e de acontecimentos ocorridos anteriormente àquele instante. Em outras palavras, a previsão da saída pode ser decomposta em

$$\hat{y} = \hat{y}_l + \hat{y}_f \quad (\text{E.7})$$

onde \hat{y}_l é a previsão da saída em função da *resposta livre* do sistema (isto é, com incrementos presente e futuros de controle nulos) e \hat{y}_f é a previsão da saída em função da *resposta forçada* (isto é, considerando apenas os incrementos presente e futuros do sinal de controle, que devem ser calculados de forma a minimizar o critério de desempenho).

Das equações de previsão anteriores tem-se que

$$\begin{aligned}
\hat{y}(t+1|t) &= -\tilde{a}_1 y(t) - \tilde{a}_2 y(t-1) - \cdots - \tilde{a}_{N_{\tilde{a}}} y(t-N_{\tilde{a}}+1) + b_0 \Delta u(t) \\
&\quad + b_1 \Delta u(t-1) + \cdots + b_{N_b} \Delta u(t-N_b) \\
\hat{y}(t+2|t) &= -\tilde{a}_1 (\hat{y}_l(t+1|t) + \hat{y}_f(t+1|t)) - \tilde{a}_2 y(t) - \cdots - \tilde{a}_{N_{\tilde{a}}} y(t-N_{\tilde{a}}+2) \\
&\quad + b_0 \Delta u(t+1) + b_1 \Delta u(t) \\
&\quad + b_2 \Delta u(t-1) + \cdots + b_{N_b} \Delta u(t-N_b+1) \\
&\quad \vdots
\end{aligned} \tag{E.8}$$

Logo, a partir das equações acima e das definições de \hat{y}_l e \hat{y}_f tem-se:

$$\begin{aligned}
\hat{y}_f(t+1|t) &= b_0 \Delta u(t) \\
\hat{y}_f(t+2|t) &= -\tilde{a}_1 \hat{y}_f(t+1|t) + b_0 \Delta u(t+1) + b_1 \Delta u(t) \\
\hat{y}_f(t+3|t) &= -\tilde{a}_1 \hat{y}_f(t+2|t) - \tilde{a}_2 \hat{y}_f(t+1|t) + b_0 \Delta u(t+2) + b_1 \Delta u(t+1) \\
&\quad + b_2 \Delta u(t) \\
&\quad \vdots
\end{aligned} \tag{E.9}$$

e

$$\begin{aligned}
\hat{y}_l(t+1|t) &= -\tilde{a}_1 y(t) - \tilde{a}_2 y(t-1) - \cdots - \tilde{a}_{N_{\tilde{a}}} y(t-N_{\tilde{a}}+1) \\
&\quad + b_1 \Delta u(t-1) + \cdots + b_{N_b} \Delta u(t-N_b) \\
\hat{y}_l(t+2|t) &= -\tilde{a}_1 \hat{y}_l(t+1|t) - \tilde{a}_2 y(t) - \cdots - \tilde{a}_{N_{\tilde{a}}} y(t-N_{\tilde{a}}+2) \\
&\quad + b_2 \Delta u(t-1) + \cdots + b_{N_b} \Delta u(t-N_b+1) \\
\hat{y}_l(t+3|t) &= -\tilde{a}_1 \hat{y}_l(t+2|t) - \tilde{a}_2 \hat{y}_l(t+1|t) - \tilde{a}_3 y(t) - \cdots - \tilde{a}_{N_{\tilde{a}}} y(t-N_{\tilde{a}}+3) \\
&\quad + b_3 \Delta u(t-1) + \cdots + b_{N_b} \Delta u(t-N_b+2) \\
&\quad \vdots
\end{aligned} \tag{E.10}$$

Exemplo

Considere o seguinte sistema:

$$Ay(t) = Bu(t - 1) \quad (\text{E.11})$$

com $A = (1 + a_1q^{-1})$ e $B = (b_0 + b_1q^{-1})$, ou seja

$$y(t) = -a_1y(t - 1) + b_0u(t - 1) + b_1u(t - 2) \quad (\text{E.12})$$

Na forma incremental, descreve-se o sistema acima como:

$$y(t) = -\tilde{a}_1y(t - 1) - \tilde{a}_2y(t - 2) + b_0\Delta u(t - 1) + b_1\Delta u(t - 2) \quad (\text{E.13})$$

onde

$$\begin{aligned} \tilde{A} &= \Delta A = (1 - q^{-1})A = (1 - q^{-1})(1 + a_1q^{-1}) = \\ &= (1 + (a_1 - 1)q^{-1} - a_1q^{-2}) = (1 + \tilde{a}_1q^{-1} + \tilde{a}_2q^{-2}) \end{aligned} \quad (\text{E.14})$$

Logo, as previsões futuras do sinal de saída são dadas a partir de (E.6) por

$$\begin{aligned} \hat{y}(t + 1 | t) &= -\tilde{a}_1y(t) - \tilde{a}_2y(t - 1) + b_0\Delta u(t) + b_1\Delta u(t - 1) \\ \hat{y}(t + 2 | t) &= -\tilde{a}_1\hat{y}(t + 1 | t) - \tilde{a}_2y(t) + b_0\Delta u(t + 1) + b_1\Delta u(t) \\ \hat{y}(t + 3 | t) &= -\tilde{a}_1\hat{y}(t + 2 | t) - \tilde{a}_2\hat{y}(t + 1 | t) + b_0\Delta u(t + 2) + b_1\Delta u(t + 1) \\ &\vdots \end{aligned} \quad (\text{E.15})$$

e considerando a equação (E.7) tem-se

$$\begin{aligned} \hat{y}_l(t + 1 | t) &= -\tilde{a}_1y(t) - \tilde{a}_2y(t - 1) + b_1\Delta u(t - 1) \\ \hat{y}_l(t + 2 | t) &= -\tilde{a}_1\hat{y}_l(t + 1 | t) - \tilde{a}_2y(t) \\ \hat{y}_l(t + 3 | t) &= -\tilde{a}_1\hat{y}_l(t + 2 | t) - \tilde{a}_2\hat{y}_l(t + 1 | t) \\ &\vdots \end{aligned} \quad (\text{E.16})$$

e

$$\begin{aligned} \hat{y}_f(t + 1 | t) &= b_0\Delta u(t) \\ \hat{y}_f(t + 2 | t) &= -\tilde{a}_1\hat{y}_f(t + 1 | t) + b_0\Delta u(t + 1) + b_1\Delta u(t) \\ &= b_0\Delta u(t + 1) + (b_1 - \tilde{a}_1b_0)\Delta u(t) \\ \hat{y}_f(t + 3 | t) &= -\tilde{a}_1\hat{y}_f(t + 2 | t) - \tilde{a}_2\hat{y}_f(t + 1 | t) + b_0\Delta u(t + 2) + b_1\Delta u(t + 1) \\ &= -\tilde{a}_1(b_0\Delta u(t + 1) + (b_1 - \tilde{a}_1b_0)\Delta u(t)) - \tilde{a}_2b_0\Delta u(t) + b_0\Delta u(t + 2) \\ &\quad + b_1\Delta u(t + 1) \\ &\vdots \end{aligned} \quad (\text{E.17})$$

Definindo

$$h_0 = b_0 ; h_1 = b_1 - \tilde{a}_1 h_0 ; h_2 = b_2 - \tilde{a}_1 h_1 - \tilde{a}_2 h_0 = -\tilde{a}_1 h_1 - \tilde{a}_2 h_0 ; \dots \quad (\text{E.18})$$

tem-se

$$\begin{aligned} \hat{y}_f(t+1|t) &= h_0 \Delta u(t) \\ \hat{y}_f(t+2|t) &= h_0 \Delta u(t+1) + h_1 \Delta u(t) \\ \hat{y}_f(t+3|t) &= h_0 \Delta u(t+2) + h_1 \Delta u(t+1) + h_2 \Delta u(t) \\ &\vdots \\ \hat{y}_f(t+N_u|t) &= h_0 \Delta u(t+N_u-1) + h_1 \Delta u(t+N_u-2) + \dots + h_{N_u-1} \Delta u(t) \\ \hat{y}_f(t+N_u+1|t) &= h_0 \Delta u(t+N_u) + h_1 \Delta u(t+N_u-1) + \dots + h_{N_u} \Delta u(t) \\ &\vdots \\ \hat{y}_f(t+N_y|t) &= h_0 \Delta u(t+N_y-1) + h_1 \Delta u(t+N_y-2) + \\ &\quad + \dots + h_{N_y-N_u} \Delta u(t+N_u-1) + \dots + h_{N_y-1} \Delta u(t) \end{aligned} \quad (\text{E.19})$$

e no caso geral:

$$\begin{aligned} \hat{y}_f(t+j|t) &= \sum_{i=0}^{j-1} h_i \Delta u(t+j-i-1) \\ h_j &= b_j - \sum_{i=1}^{\min(j, N_{\tilde{a}})} \tilde{a}_i h_{j-i} ; h_0 = b_0 \end{aligned} \quad (\text{E.20})$$

É interessante notar que $\{h_0, h_1, h_2, \dots\}$ é a seqüência de resposta ao impulso do sistema incremental, que possui função de transferência pulsada no operador deslocamento q^{-1} dada por $\frac{B(q^{-1})}{\tilde{A}(q^{-1})}$.

Considerando a restrição imposta ao sinal de controle no problema de otimização estabelecido em (E.5), rescreve-se o conjunto de previsões forçadas em (E.19) como:

$$\hat{\mathbf{y}}_{\mathbf{f}} = H \Delta \mathbf{u} \quad (\text{E.21})$$

onde

$$\hat{\mathbf{y}}_{\mathbf{f}} = [\hat{y}_f(t+1|t) \cdots \hat{y}_f(t+N_y|t)]^T \quad (\text{E.22})$$

$$\Delta \mathbf{u} = [\Delta u(t) \cdots \Delta u(t + N_u - 1)]^T \quad (\text{E.23})$$

$$H = \begin{bmatrix} h_0 & 0 & \cdots & 0 \\ h_1 & h_0 & \cdots & 0 \\ \vdots & & \ddots & \\ h_{N_u-1} & h_{N_u-2} & \cdots & h_0 \\ h_{N_u} & h_{N_u-1} & \cdots & h_1 \\ & & \vdots & \\ h_{N_y-1} & h_{N_y-2} & \cdots & h_{N_y-N_u} \end{bmatrix} \quad (\text{E.24})$$

e definindo

$$\hat{\mathbf{y}}_1 = [\hat{y}_l(t+1|t) \cdots \hat{y}_l(t+N_y|t)]^T \quad (\text{E.25})$$

$$\hat{\mathbf{y}} = [\hat{y}(t+1|t) \cdots \hat{y}(t+N_y|t)]^T \quad (\text{E.26})$$

tem-se de (E.7) que

$$\hat{\mathbf{y}} = H \Delta \mathbf{u} + \hat{\mathbf{y}}_1 \quad (\text{E.27})$$

Pode-se ainda tomar apenas o subconjunto de previsões de interesse que estão envolvidas no critério (E.4), como segue:

$$\hat{\mathbf{y}}' = H' \Delta \mathbf{u} + \hat{\mathbf{y}}'_1 \quad (\text{E.28})$$

onde

$$\hat{\mathbf{y}}'_1 = [\hat{y}_l(t+N_1|t) \cdots \hat{y}_l(t+N_y|t)]^T \quad (\text{E.29})$$

$$\hat{\mathbf{y}}' = [\hat{y}(t+N_1|t) \cdots \hat{y}(t+N_y|t)]^T \quad (\text{E.30})$$

e H' é a submatriz constituída das últimas $N_y - N_1 + 1$ linhas de H .

E.2.2 Otimização do Critério de Desempenho

O critério (E.4) pode ser rescrito como:

$$J = (\hat{y}(t + N_1 | t) - w(t + N_1))^2 + \dots + (\hat{y}(t + N_y | t) - w(t + N_y))^2 + \lambda((\Delta u(t))^2 + \dots + (\Delta u(t + N_u - 1))^2) \quad (\text{E.31})$$

e definindo $\mathbf{w}' = [w(t + N_1) \dots w(t + N_y)]^T$ tem-se

$$J = (\hat{\mathbf{y}}' - \mathbf{w}')^T (\hat{\mathbf{y}}' - \mathbf{w}') + \lambda \Delta \mathbf{u}^T \Delta \mathbf{u} \quad (\text{E.32})$$

Portanto

$$J = (H' \Delta \mathbf{u} + \hat{\mathbf{y}}'_1 - \mathbf{w}')^T (H' \Delta \mathbf{u} + \hat{\mathbf{y}}'_1 - \mathbf{w}') + \lambda \Delta \mathbf{u}^T \Delta \mathbf{u} \quad (\text{E.33})$$

e utilizando a condição necessária para otimalidade tem-se

$$\frac{\partial J}{\partial \Delta \mathbf{u}} = 2H'^T H' \Delta \mathbf{u} + 2H'^T (\hat{\mathbf{y}}'_1 - \mathbf{w}') + 2\lambda \Delta \mathbf{u} = 0 \quad (\text{E.34})$$

que implica na lei de controle ótima para o GPC dada por:

$$\Delta \mathbf{u} = (H'^T H' + \lambda I)^{-1} H'^T (\mathbf{w}' - \hat{\mathbf{y}}'_1) \quad (\text{E.35})$$

onde $\Delta \mathbf{u}$ é definido em (E.23). Como comentado anteriormente, utiliza-se a estratégia *Receding Horizon*, de tal forma que em cada instante t calcula-se a lei de controle em (E.35) e aplica-se somente o primeiro elemento, ou seja,

$$u(t) = u(t - 1) + \Delta u(t) \quad (\text{E.36})$$

É importante observar que a obtenção da lei de controle posicional a partir da lei incremental através da equação acima insere automaticamente um integrador na malha, permitindo o cancelamento de erros em regime permanente para sinais de referência do tipo degrau.

E.2.3 Restrições de Entrada e Saída

A lei de controle em (E.35) foi obtida sem considerar restrições operacionais na entrada manipulada e na saída do sistema. Limitantes inferiores e superiores, no entanto, podem facilmente ser impostos tanto ao sinal de controle quanto à saída prevista do sistema. Nesse caso, a lei de controle $\Delta \mathbf{u}$ deve ser obtida em cada instante de amostragem através da minimização do critério quadrático em (E.33) sujeita ao conjunto de restrições impostas. Como as referidas restrições são dadas por desigualdades lineares em $\Delta \mathbf{u}$, a solução ótima global estrita para o problema pode ser obtida através de Programação Quadrática (Bazaraa et al., 1993). Previsões de dinâmicas relacionadas a perturbações mensuráveis agindo sobre o sistema e que sejam representadas pelo modelo podem também ser facilmente inseridas no cálculo da lei de controle (Clarke, 1994; Camacho e Bordons, 1999).

E.2.4 Modelos FBO

O cálculo iterativo (recursivo) da previsão livre do sistema através da equação a diferenças (E.10) conduz a dois tipos de problemas: A inserção de erros de previsão devido à presença de eventuais ruídos de medida nas amostras $y(t - N_{\bar{a}} + 1), \dots, y(t)$ e a realimentação desses erros ao longo do horizonte de previsão. Pode-se, no entanto, contornar esses problemas realizando a previsão livre através de uma representação em espaço-de-estados em lugar da equação a diferenças. Nesse caso não existe recursão da saída e a condição inicial para a previsão livre é representada pelo conjunto atual de estados, e não mais pelas medidas passadas dos sinais do sistema. A questão que se coloca é como estimar o conjunto de estados de forma adequada. Nesse contexto, representações de estados de particular interesse são os modelos FBO descritos no capítulo 5. Esses modelos não requerem um procedimento específico para a estimação dos estados, tal como um observador (Chen, 1998), uma vez que: *i*) Os estados não são supostos serem de fato estados do sistema. Logo, a equação dinâmica dos modelos FBO não representa uma aproximação da dinâmica de estados do sistema; *ii*) A equação dinâmica dos modelos FBO é sempre estável e, portanto, os estados convergem, a partir de uma condição inicial qualquer, para os valores corretos após um período transitório. Para efeito do cálculo da matriz dinâmica H em (E.24), um modelo equivalente na forma de função de transferência pulsada pode ser obtido diretamente a partir da representação de estados (Åström e Wittenmark, 1997; Chen, 1998).

A utilização da representação de estados FBO para a realização da previsão da resposta livre do sistema faz com que o erro entre o valor de referência desejado e o valor real de saída do sistema não seja inserido no controlador. A razão é a ausência da medida da saída no cálculo das previsões e, portanto, da lei de controle, que nesse caso opera em malha aberta. Como consequência, o sinal indisponível do erro de saída não pode ser integrado pelo controlador. Logo, erros de modelagem e/ou devido a perturbações não modeladas podem resultar em erros de regime, mesmo para sinais de referência do tipo degrau. Uma forma de solucionar esse problema é a inclusão de uma

compensação de erros no critério (E.4), como segue:

$$J = \sum_{j=N_1}^{N_y} (\hat{y}(t+j|t) + \hat{d}(t+j|t) - w(t+j))^2 + \sum_{j=1}^{N_u} \lambda (\Delta u(t+j-1))^2 \quad (\text{E.37})$$

onde os termos de compensação $\hat{d}(t+j|t)$ são dados pela diferença entre as saídas futuras do sistema $y(t+j)$ e as respectivas previsões do modelo $\hat{y}(t+j|t)$. Uma vez que não dispõem-se das medidas futuras da saída, pode-se aproximar esses termos como (Garcia et al., 1989; Nazzetta et al., 1998):

$$\hat{d}(t+j|t) \triangleq \hat{d}(t|t) = y(t) - \hat{y}(t|t-1) \quad (\text{E.38})$$

onde $\hat{y}(t|t-1)$ é a previsão da saída para o instante atual t realizada no instante anterior $t-1$. Note que, em regime, a aproximação acima é exata, permitindo assim o cancelamento de um eventual *offset* na malha.

Para o cálculo da lei de controle do GPC com base no critério (E.37) basta substituir o vetor de referência \mathbf{w}' em (E.35) por um novo vetor dado por $\mathbf{w}'' = [w(t+N_1) - \hat{d}(t|t) \ \cdots \ w(t+N_y) - \hat{d}(t|t)]^T$, isto é, um vetor de referência artificial.

E.2.5 Propriedades do Controle Preditivo

Algumas propriedades da estratégia de controle preditivo são apresentadas abaixo:

- Com o conhecimento da referência em instantes futuros, o controlador pode reagir de forma antecipativa.
- Restrições operacionais podem ser consideradas explicitamente na obtenção da lei de controle ótima.
- O controlador trata de forma simples a presença de atraso de transporte no sistema a ser controlado. Esse tratamento pode ser realizado de forma direta, através da sua consideração explícita no modelo, ou indireta, através da definição adequada dos horizontes de previsão N_1 e N_y . Nesse último caso, garantindo N_y maior do que o atraso de transporte e suas eventuais variações, não é necessário o conhecimento exato desse atraso. Pode-se também adotar um horizonte inicial $N_1 > 1$, reduzindo assim o esforço computacional na medida que desconsideram-se as previsões que não dependem, devido ao tempo morto, dos sinais de controle a serem determinados.

Bibliografia

- Alataris, K., Berger, T. W. e Marmarelis, V. Z. (2000). A novel network for nonlinear modeling of neural systems with arbitrary point-process inputs, *Neural Networks* **13**: 255–266.
- Anderson, B. e Moore, J. (1979). *Optimal Filtering*, Prentice Hall.
- Andrietta, S. R. (1994). *Modelagem, Simulação e Controle de Fermentação Alcoólica Contínua em Escala Industrial*, Tese de Doutorado, FEA/UNICAMP - Campinas/SP/Brasil.
- Andrietta, S. R. e Maugeri, F. (1994). Optimum design of a continuous fermentation unit of an industrial plant for alcohol production, *in* E. Galindo e O. T. Ramírez (eds), *Advances in Bioprocess Engineering*, Kluwer Academic Publishers, pp. 47–52.
- Arto, V., Hannu, P. e Halme, A. (2001). Modeling of chromatographic separation process with Wiener-MLP representation, *Journal of Process Control* **11**: 443–458.
- Åström, K. J. e Wittenmark, B. (1995). *Adaptive Control*, 2nd edn, Addison Wesley.
- Åström, K. J. e Wittenmark, B. (1997). *Computer-Controlled Systems*, 3rd edn, Prentice Hall.
- Babuška, R., Fantuzzi, C., Kaymak, U. e Verbruggen, H. B. (1996). Improved inference for Takagi-Sugeno models, *Proc. 5th IEEE Internat. Conference on Fuzzy Systems*, New Orleans/USA, pp. 701–706.
- Babuška, R. e Verbruggen, H. B. (1995). A new identification method for linguistic fuzzy models, *Proc. 4th IEEE Internat. Conference on Fuzzy Systems*, Yokohama/Japan, pp. 905–912.
- Babuška, R. e Verbruggen, H. B. (1996). An overview of fuzzy modeling for control, *Control Engineering Practice* **4**: 1593–1605.
- Back, A. D. e Tsoi, A. C. (1996). Nonlinear system identification using discrete Laguerre functions, *Journal of Systems Engineering* **6**: 194–207.

- Balestrino, A., Caiti, A. e Zanobini, G. (1999). Identification of Wiener-type nonlinear systems by Laguerre filters and neural networks, *Proc. 14th International Federation of Automatic Control (IFAC) World Congress*, Beijing/China, pp. 433–438.
- Bazaraa, M. S., Sherali, H. D. e Shetty, C. M. (1993). *Nonlinear Programming: Theory and Algorithms*, 2nd edn, John Wiley & Sons.
- Bellman, R. E. (1961). *Adaptive Control Processes*, Princeton University Press.
- Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithm*, Plenum Press.
- Bezdek, J. C., Keller, J. M., Krishnapuram, R., Kuncheva, L. I. e Pal, N. R. (1999). Will the *real* iris data please stand up?, *IEEE Trans. Fuzzy Systems* **7**: 368–369.
- Billings, S. A. (1980). Identification of nonlinear systems - a survey, *IEE Proc. Pt D* **127**(6): 272–285.
- Bissessur, Y. e Naguib, R. N. G. (1995). Buried plant detection: A Volterra series modelling approach using artificial neural networks, *Neural Networks* **9**: 1045–1060.
- Bourke, M. M. e Fisher, D. G. (2000a). Identification algorithms for fuzzy relational matrices, part 1: Non-optimizing algorithms, *Fuzzy Sets and Systems* **109**: 305–320.
- Bourke, M. M. e Fisher, D. G. (2000b). Identification algorithms for fuzzy relational matrices, part 2: Optimizing algorithms, *Fuzzy Sets and Systems* **109**: 321–341.
- Box, G. E. P. e Jenkins, G. M. (1970). *Time Series Analysis, Forecasting and Control*, Holden Day.
- Boyd, S. e Chua, L. O. (1985). Fading memory and the problem of approximating nonlinear operators with Volterra series, *IEEE Trans. on Circuits and Systems* **32**(11): 1150–1161.
- Branco, P. J. C. e Dente, J. A. (2000). A fuzzy relational identification algorithm and its application to predict the behaviour of a motor drive system, *Fuzzy Sets and Systems* **109**: 343–354.
- Broome, P. W. (1965). Discrete orthonormal sequences, *Journal of the Association for Computing Machinery* **12**(2): 151–168.
- Broomhead, D. S. e Lowe, D. (1988). Multivariate functional interpolation and adaptive networks, *Complex Systems* **2**: 321–355.

- Brown, M., Bossley, K. M., Mills, D. J. e Harris, C. J. (1995). High dimensional neuro-fuzzy systems: Overcoming the curse of dimensionality, *Proc. 4th IEEE Internat. Conference on Fuzzy Systems*, Yokohama/Japan, pp. 2139–2146.
- Buckley, J. J. e Hayashi, Y. (1994). Fuzzy neural networks: A survey, *Fuzzy Sets and Systems* **66**: 1–13.
- Camacho, E. F. e Bordons, C. (1999). *Model Predictive Control*, Springer-Verlag.
- Campello, R. J. G. B. (1997). *Uma nova metodologia para identificação adaptativa de modelos relacionais nebulosos*, Tese de Mestrado, DCA/FEEC/UNICAMP - Campinas/SP/Brasil.
- Campello, R. J. G. B. e Amaral, W. C. (1998a). Modelos relacionais nebulosos: Identificação e sintonia fina, *Anais XII Congresso Brasileiro de Automática*, Uberlândia/Brasil, pp. 2169–2174.
- Campello, R. J. G. B. e Amaral, W. C. (1998b). Refinement and identification of fuzzy relational models, *Proc. 7th IEEE Internat. Conference on Fuzzy Systems*, Anchorage/USA, pp. 651–656.
- Campello, R. J. G. B. e Amaral, W. C. (1999a). Equivalência entre modelos nebulosos e redes neurais, *Anais IV Simpósio Brasileiro de Automação Inteligente*, São Paulo/Brasil, pp. 208–212.
- Campello, R. J. G. B. e Amaral, W. C. (1999b). Extracting linguistic knowledge from fuzzy relational models, *Proc. 8th IFSA World Congress*, Taipei/Taiwan, pp. 1061–1065.
- Campello, R. J. G. B. e Amaral, W. C. (1999c). Otimização irrestrita de modelos nebulosos e neurais com estrutura em cascata, *Anais IV Simpósio Brasileiro de Automação Inteligente*, São Paulo/Brasil, pp. 449–454.
- Campello, R. J. G. B. e Amaral, W. C. (1999d). A relational approach for complex system identification, *Controle & Automação* **10**: 139–148.
- Campello, R. J. G. B. e Amaral, W. C. (2000a). Modelos relacionais nebulosos: Aspectos numéricos e linguísticos, *Anais XIII Congresso Brasileiro de Automática*, Florianópolis/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. (2002a). Desenvolvimento ótimo de modelos de Volterra em funções ortonormais de Laguerre, *Anais XIV Congresso Brasileiro de Automática*, Natal/Brasil, p. (aceito para publicação).
- Campello, R. J. G. B. e Amaral, W. C. (2002b). Hierarchical fuzzy relational models: Linguistic interpretation and universal approximation, *Proc. 11th IEEE Internat. Conference on Fuzzy Systems*, Honolulu/USA, pp. 162–167.
- Campello, R. J. G. B. e Amaral, W. C. (2002c). Takagi-Sugeno fuzzy models within orthonormal basis function framework and their application to process control, *Proc. 11th IEEE Internat. Conference on Fuzzy Systems*, Honolulu/USA, pp. 1399–1404.
- Campello, R. J. G. B. e Amaral, W. C. (nd). Towards true linguistic modeling through optimal numerical solutions, *submitted*.
- Campello, R. J. G. B., Amaral, W. C. e Favier, G. (2001). Optimal Laguerre series expansion of discrete Volterra models, *Proc. European Control Conference*, Porto/Portugal, pp. 372–377.
- Campello, R. J. G. B., Meleiro, L. A. C., Amaral, W. C. e Maciel Filho, R. (2001). Identification of a bioprocess using Laguerre function based models, *Proc. 6th World Congress on Chemical Engineering*, Melbourne/Australia, p. CD.
- Campello, R. J. G. B., Nazzetta, R. M. e Amaral, W. C. (1997). A new methodology for fuzzy model identification, *Proc. 7th IFSA World Congress*, Prague/Czech Republic, pp. 366–370.
- Campello, R. J. G. B., Nazzetta, R. M. e Amaral, W. C. (1998). A highly adaptive algorithm for fuzzy modelling of systems, *Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems* **6**: 35–50.
- Campello, R. J. G. B., Oliveira, G. H. C., Von Zuben, F. J. e Amaral, W. C. (1999). Redes neurais com base de funções ortonormais, *Anais IV Simpósio Brasileiro de Automação Inteligente*, São Paulo/Brasil, pp. 213–218.
- Campello, R. J. G. B., Von Zuben, F. J., Amaral, W. C., Meleiro, L. A. C. e Maciel Filho, R. (nd). Hierarchical fuzzy models within the framework of orthonormal basis functions and their application to bioprocess control, *submitted*.
- Cavalcante Júnior, F. L. (2000). *Controle preditivo utilizando um modelo nebuloso*, Tese de Mestrado, DCA/FEEC/UNICAMP - Campinas/SP/Brasil.
- Cavalcante Júnior, F. L. e Amaral, W. C. (2000). Controle preditivo a partir de um modelo nebuloso e aplicação ao controle de pH, *Anais XIII Congresso Brasileiro de Automática*, Florianópolis/Brasil, pp. 2180–2185.

- Chen, C. T. (1998). *Linear System Theory and Design*, 3rd edn, Oxford Univ. Press.
- Chen, W. e Wang, L.-X. (2000). A note on universal approximation by hierarchical fuzzy systems, *Information Sciences* **123**: 241–248.
- Cho, K. B. e Wang, B. H. (1996). Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction, *Fuzzy Sets and Systems* **83**: 325–339.
- Chon, K. H., Holstein-Rathlou, N. H., Marsh, D. J. e Marmarelis, V. Z. (1998). Comparative nonlinear modeling of renal autoregulation in rats: Volterra approach versus artificial neural networks, *IEEE Trans. Neural Networks* **9**: 430–435.
- Chung, F.-L. e Duan, J.-C. (2000). On multistage fuzzy neural network modeling, *IEEE Trans. Fuzzy Systems* **8**: 125–142.
- Chung, F.-L. e Lee, T. (1998). Analytical resolution and numerical identification of fuzzy relational systems, *IEEE Trans. Systems, Man and Cybernetics - Part B: Cybernetics* **28**: 919–924.
- Clarke, D. W. (ed.) (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.
- Combs, W. E. e Andrews, J. E. (1998). Combinatorial rule explosion eliminated by a fuzzy rule configuration, *IEEE Trans. Fuzzy Systems* **6**: 1–11.
- Czogala, E. e Pedrycz, W. (1981). On identification in fuzzy systems and its applications in control problems, *Fuzzy Sets and Systems* **6**: 73–83.
- Czogala, E. e Pedrycz, W. (1982). Control problems in fuzzy systems, *Fuzzy Sets and Systems* **7**: 257–273.
- Dechechi, E. C. (1998). *Controle Avançado Preditivo Adaptativo “DMC Multivariável Adaptativo”*, Tese de Doutorado, FEQ/UNICAMP - Campinas/SP/Brasil.
- Delgado, M. R. B. S. (2002). *Projeto Automático de Sistemas Nebulosos: Uma Abordagem Co-Evolutiva*, Tese de Doutorado, DCA/FEEC/UNICAMP - Campinas/SP/Brasil.

- Delgado, M. R., Von Zuben, F. J. e Gomide, F. A. C. (2000). Evolutionary design of Takagi-Sugeno fuzzy systems: A modular and hierarchical approach, *Proc. 9th IEEE Internat. Conference on Fuzzy Systems*, Houston/USA, pp. 447–452.
- Desoer, C. A. e Vidyasagar, M. (1975). *Feedback Systems: Input-Output Properties*, Academic Press.
- Di Nola, A., Pedrycz, W. e Sessa, S. (1985). When is a fuzzy relation decomposable in two fuzzy sets?, *Fuzzy Sets and Systems* **16**: 87–90.
- Di Nola, A., Pedrycz, W., Sessa, S. e Sanchez, E. (1991). Fuzzy relation equations theory as a basis of fuzzy modelling: An overview, *Fuzzy Sets and Systems* **40**: 415–429.
- Duan, J.-C. e Chung, F.-L. (2001). Cascade fuzzy neural network model based on syllogistic fuzzy reasoning, *IEEE Trans. Fuzzy Systems* **9**: 293–306.
- Dubois, D. e Prade, H. (1980). *Fuzzy Sets and Systems: Theory and Applications*, Academic Press.
- Dumont, G. A. e Fu, Y. (1993). Non-linear adaptive control via Laguerre expansion of Volterra kernels, *Int. J. Adaptive Control and Signal Processing* **7**: 367–382.
- Espinosa, J. e Vandewalle, J. (2000). Constructing fuzzy models with linguistic integrity from numerical data - AFRELI algorithm, *IEEE Trans. Fuzzy Systems* **8**: 591–600.
- Eykhoff, P. (1974). *System Identification: Parameter and State Estimation*, John Wiley & Sons.
- Figueiredo, M., Gomide, F. A. C. e Pedrycz, W. (1995). Fuzzy neurons and networks: Models and learning, *Proc. European Congress on Intelligent Techniques and Soft Computing - EUFIT'95*, Aachen/Germany, pp. 332–335.
- Filev, D. (1991). Fuzzy modeling of complex systems, *Int. J. Approximate Reasoning* **5**: 281–290.
- Filev, D. P. e Yager, R. R. (1997). Learning celibate fuzzy models, *Proc. 7th IFSA World Congress*, Prague/Czech Republic, pp. 422–427.
- Fischer, M., Nelles, O. e Isermann, R. (1998a). Adaptive predictive control of a heat exchanger based on a fuzzy model, *Control Engineering Practice* **6**: 259–269.
- Fischer, M., Nelles, O. e Isermann, R. (1998b). Predictive control based on local linear fuzzy models, *Int. J. Systems Science* **29**: 679–697.
- Fogel, D. (2000). *Evolutionary Computation: Principles and Practice for Signal Processing*, Society of Photo-Optical Instrumentation Engineers.

- Franklin, G. F., Powell, J. D. e Workman, M. L. (1990). *Digital Control of Dynamic Systems*, 2nd edn, Addison-Wesley.
- Fu, Y. e Dumont, G. A. (1993). An optimum time scale for discrete Laguerre network, *IEEE Transactions on Automatic Control* **38**(6): 934–938.
- Furuhashi, T., Matsushita, S., Tsutsui, H. e Uchikawa, Y. (1997). Knowledge extraction from hierarchical fuzzy model obtained by fuzzy neural networks and genetic algorithm, *Proc. IEEE Internat. Conference on Neural Networks*, Houston/USA, pp. 2374–2379.
- Garcia, C. E., Prett, D. M. e Morari, M. (1989). Model predictive control: Theory and practice - a survey, *Automatica* **25**(3): 335–348.
- Gomide, F. A. C. e Gudwin, R. R. (1994). Modelagem, controle, sistemas e lógica fuzzy, *Controle & Automação* **4**: 97–114.
- Gottwald, S. e Pedrycz, W. (1986). Solvability of fuzzy relational equations and manipulation of fuzzy data, *Fuzzy Sets and Systems* **18**: 45–65.
- Graham, A. (1981). *Kronecker Products and Matrix Calculus*, Ellis Horwood Ltd.
- Graham, B. P. e Newell, R. B. (1988). Fuzzy identification and control of a liquid level rig, *Fuzzy Sets and Systems* **26**: 255–273.
- Graham, B. P. e Newell, R. B. (1989). Fuzzy adaptive control of a first-order process, *Fuzzy Sets and Systems* **31**: 47–65.
- Guillaume, S. (2001). Designing fuzzy inference systems from data: An interpretability-oriented review, *IEEE Trans. Fuzzy Systems* **9**: 426–443.
- Gupta, M. M. e Rao, D. H. (1994). On the principles of fuzzy neural networks, *Fuzzy Sets and Systems* **61**: 1–18.
- Güven, M. K. e Passino, K. M. (2001). Avoiding exponential parameter growth in fuzzy systems, *IEEE Trans. Fuzzy Systems* **9**: 194–199.
- Harris, C. J., Moore, C. G. e Brown, M. (1993). *Intelligent Control: Aspects of Fuzzy Logic and Neural Nets*, World Scientific.
- Haykin, S. (1989). *Modern Filters*, Macmillan Publishing Company.
- Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*, 2nd edn, Prentice Hall.
- Hebb, D. O. (1949). *The Organization of Behaviour: A Neuropsychological Theory*, Wiley.

- Hellendoorn, H. e Driankov, D. (eds) (1997). *Fuzzy Model Identification: Selected Approaches*, Springer-Verlag.
- Higashi, M. e Klir, G. J. (1984a). Identification of fuzzy relation systems, *IEEE Trans. Systems, Man and Cybernetics* **SMC-14**: 349–355.
- Higashi, M. e Klir, G. J. (1984b). Resolution of finite fuzzy relation equations, *Fuzzy Sets and Systems* **13**: 65–82.
- Holve, R. (1997). Rule generation for hierarchical fuzzy systems, *Proc. Annual Conference of the North American Fuzzy Information Processing Society*, Syracuse/USA, pp. 444–449.
- Holve, R. (1998). Investigation of automatic rule generation for hierarchical fuzzy systems, *Proc. 7th IEEE Internat. Conference on Fuzzy Systems*, Anchorage/USA, pp. 973–978.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities, *Proc. National Academy of Sciences*, Vol. 79, USA, pp. 2554–2558.
- Hopfield, J. J. e Tank, T. W. (1985). Neural computation of decisions in optimization problems, *Biological Cybernetics* **52**: 141–152.
- Horikawa, S.-I., Furuhashi, T. e Uchikawa, Y. (1992). On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm, *IEEE Trans. Neural Networks* **3**: 801–806.
- Hunt, K. J., Haas, R. e Murray-Smith, R. (1996). Extending the functional equivalence of radial basis function networks and fuzzy inference systems, *IEEE Trans. Neural Networks* **7**: 776–781.
- Ishibuchi, H., Fujioka, R. e Tanaka, H. (1993). Neural networks that learn from fuzzy If-Then rules, *IEEE Trans. Fuzzy Systems* **1**: 85–97.
- Ishibuchi, H. e Nakashima, T. (2001). Effect of rule weights in fuzzy rule-based classification problems, *IEEE Trans. Fuzzy Systems* **9**: 506–515.
- Jamshidi, M. (1997). Fuzzy control of complex systems: Structures and implementation, *Proc. 7th IFSA World Congress*, Prague/Czech Republic, pp. 324–329.
- Jang, J.-S. R. (1993). ANFIS: Adaptive-network-based fuzzy inference system, *IEEE Trans. Systems, Man and Cybernetics* **23**: 665–685.
- Jin, Y. (2000). Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement, *IEEE Trans. Fuzzy Systems* **8**: 212–221.

- Johansen, T. A., Shorten, R. e Murray-Smith, R. (2000). On the interpretation and identification of dynamic Takagi-Sugeno fuzzy models, *IEEE Trans. Fuzzy Systems* **8**: 297–313.
- Kashyap, R. L. e Rao, A. R. (1976). *Dynamic Stochastic Models from Empirical Data*, Academic Press.
- Kelkar, B. e Postlethwaite, B. (1998). Enhancing the generality of fuzzy relational models for control, *Fuzzy Sets and Systems* **100**: 117–129.
- King, P. J. e Mamdani, E. H. (1977). The application of fuzzy control systems to industrial processes, *Automatica* **13**: 235–242.
- Kiszka, J. B., Kočańska, M. E. e Sliwińska, D. S. (1985). The influence of some fuzzy implication operators on the accuracy of a fuzzy model - part I, *Fuzzy Sets and Systems* **15**: 111–128.
- Kohonen, T. (1982). Self-organized formation of topologically correct features maps, *Biological Cybernetics* **43**: 59–69.
- Kohonen, T. (1997). *Self-Organizing Maps*, 2nd edn, Springer-Verlag.
- Kosko, B. (1992). *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall.
- Kosko, B. (1997). *Fuzzy Engineering*, Prentice Hall.
- Kreysig, E. (1978). *Introductory Functional Analysis with Applications*, John Wiley & Sons.
- Kreysig, E. (1993). *Advanced Engineering Mathematics*, 7th edn, John Wiley & Sons.
- Lee, C. C. (1990). Fuzzy logic in control systems: Fuzzy logic controller - Parts 1 and 2, *IEEE Trans. Systems, Man and Cybernetics* **20**: 404–435.
- Lee, Y. C., Hwang, C. e Shih, Y. P. (1994). A combined approach to fuzzy model identification, *IEEE Trans. Systems, Man and Cybernetics* **24**: 736–743.
- Lee, Y. W. (1960). *Statistical Theory of Communication*, Wiley.
- Lin, Y. e Cunningham III, G. A. (1995). A new approach to fuzzy-neural system modeling, *IEEE Trans. Fuzzy Systems* **3**: 190–198.
- Lindskog, P. (1996). *Methods, Algorithms and Tools for System Identification Based on Prior Knowledge*, PhD thesis, Linköping University - Sweden.
- Ljung, L. (1999). *System Identification: Theory for the user*, 2nd edn, Prentice Hall.

- Luenberger, D. G. (1997). *Optimization by Vector Space Methods*, John Wiley & Sons.
- Mäkilä, P. M. (1990). Approximation of stable systems by Laguerre filters, *Automatica* **26**(2): 333–345.
- Mamdani, E. H. (1974). Application of fuzzy algorithms for control of simple dynamic plant, *IEE Proceedings* **121**: 1585–1588.
- Mamdani, E. H. (1977). Application of fuzzy logic to approximate reasoning using linguistic synthesis, *IEEE Trans. Computers* **C-26**: 1182–1191.
- Maner, B. R., Doyle III, F. J., Ogunnaike, B. A. e Pearson, R. K. (1996). Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models, *Automatica* **32**(9): 1285–1301.
- Marmarelis, V. Z. e Zhao, X. (1997). Volterra models and three-layer perceptrons, *IEEE Trans. Neural Networks* **8**: 1421–1433.
- Masnadi-Shirazi, M. A. e Ahmed, N. (1991). Optimum Laguerre networks for a class of discrete-time systems, *IEEE Transactions on Signal Processing* **39**(9): 2104–2108.
- McCulloch, W. S. e Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activities, *Bulletin of Mathematical Biophysics* **5**: 115–133.
- Medsker, L. R. e Jain, L. C. (eds) (1999). *Recurrent Neural Networks: Design and Applications*, CRC Press.
- Meleiro, L. A. C. (2002). *Projeto e Aplicação de Controladores Baseados em Modelos Lineares, Neurais e Nebulosos*, Tese de Doutorado (a ser publicada), DPQ/FEQ/UNICAMP - Campinas/SP/Brasil.
- Meleiro, L. A. C. e Maciel Filho, R. (2000). A self-tuning adaptive control applied to an industrial large-scale ethanol production, *Computers and Chemical Engineering* **24**(2-7): 925–930.
- Meleiro, L. A. C., Maciel Filho, R., Campello, R. J. G. B. e Amaral, W. C. (2001). Hierarchical neural fuzzy models as a tool for process identification: A bioprocess application, in I. M. Mujtaba e M. A. Hussain (eds), *Application of Neural Networks and Other Learning Technologies in Process Engineering*, Imperial College Press.
- Meleiro, L. A. C., Maciel Filho, R., Campello, R. J. G. B. e Amaral, W. C. (nd). Application of hierarchical neural fuzzy models to modeling and control of a bioprocess, *submitted* .
- Mendel, J. M. (1995). Fuzzy logic systems for engineering: A tutorial, *Proceedings of the IEEE* **83**: 345–377.

- Mendel, J. M. e Liang, Q. (1999). Comments on “combinatorial rule explosion eliminated by a fuzzy rule configuration”, *IEEE Trans. Fuzzy Systems* **7**: 369–371.
- Mesarovic, M. D., Macko, D. e Takahara, Y. (1970). *Theory of Hierarchical, Multilevel, Systems*, Academic Press.
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Verlag.
- Minsky, M. L. e Papert, S. A. (1969). *Perceptrons: An Introduction to Computational Geometry*, MIT Press.
- Mitchell, M. (1998). *An Introduction to Genetic Algorithms*, MIT Press.
- Mizumoto, M. (1982). Comparison of fuzzy reasoning methods, *Fuzzy Sets and Systems* **8**: 253–283.
- Mizumoto, M. (1991). Min-Max-Gravity method versus Product-Sum-Gravity method for fuzzy controls, *Proc. 4th IFSA World Congress*, Brussels/Belgium, pp. 127–130.
- Mizumoto, M. (1996). Product-Sum-Gravity method = Fuzzy-Singleton-type reasoning method = Simplified fuzzy reasoning method, *Proc. 5th IEEE Internat. Conference on Fuzzy Systems*, New Orleans/USA, pp. 2098–2102.
- Narendra, K. S. e Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks, *IEEE Trans. Neural Networks* **1**: 4–27.
- Nazzetta, R. M., Oliveira, G. H. C., Campello, R. J. G. B. e Amaral, W. C. (1998). Controle preditivo de processos não lineares utilizando modelos dinâmicos nebulosos, *Anais XII Congresso Brasileiro de Automática*, Uberlândia/Brasil, pp. 163–168.
- Ninness, B. e Gustafsson, F. (1995). Orthonormal bases for system identification, *Proc. 3rd European Control Conference*, Vol. 1, Rome/Italy, pp. 13–18.
- Nozaki, K., Ishibuchi, H. e Tanaka, H. (1997). A simple but powerful heuristic method for generating fuzzy rules from numerical data, *Fuzzy Sets and Systems* **86**: 251–270.
- Oliveira, G. H. C. (1997). *Controle Preditivo para Processos com Incertezas Estruturadas baseado em Séries de Funções Ortonormais*, Tese de Doutorado, DCA/FEEC/UNICAMP - Campinas/SP/Brasil.
- Oliveira, G. H. C. e Amaral, W. C. (2000). Identificação e controle preditivo de processos não lineares utilizando séries de Volterra e bases de funções ortonormais, *Anais XIII Congresso Brasileiro de Automática*, Florianópolis/Brasil, pp. 2174–2179.

- Oliveira, G. H. C., Amaral, W. C., Favier, G. e Dumont, G. A. (2000). Constrained robust predictive controller for uncertain processes modeled by orthonormal series functions, *Automatica* **36**(4): 563–571.
- 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, G. H. C., Lopez, J. F., Favier, G. e Amaral, W. C. (1996). Review of predictive control methods using Laguerre functions, *Proc. IEEE/IMACS/CESA*, Lille/France, pp. 304–309.
- Oliveira, J. V. (1993a). Neuron inspired learning rules for fuzzy relational structures, *Fuzzy Sets and Systems* **57**: 41–53.
- Oliveira, J. V. (1993b). On optimal fuzzy systems I/O interfaces, *Proc. 2nd IEEE Internat. Conference on Fuzzy Systems*, San Francisco/USA, pp. 851–856.
- Oliveira, J. V. (1995). A design methodology for fuzzy system interfaces, *IEEE Trans. Fuzzy Systems* **3**: 404–414.
- Oliveira, J. V. (1999a). Semantic constraints for membership function optimization, *IEEE Trans. Systems, Man and Cybernetics - Part A: Systems and Humans* **29**: 128–138.
- Oliveira, J. V. (1999b). Towards neuro-linguistic modeling: Constraints for optimization of membership functions, *Fuzzy Sets and Systems* **106**: 357–380.
- Oliveira, J. V. e Lemos, J. M. (1995). Long-range predictive adaptive fuzzy relational control, *Fuzzy Sets and Systems* **70**: 337–357.
- 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, J. V. e Lemos, J. M. (1998). Improving adaptive fuzzy control performance by speeding up identification: Application to an electric furnace, *Journal of Intelligent and Fuzzy Systems* **6**: 297–314.
- Passino, K. M. e Yurkovich, S. (1997). *Fuzzy Control*, Addison Wesley.
- Pedrycz, W. (1981). An approach to the analysis of fuzzy systems, *Int. J. Control* **34**: 403–421.
- Pedrycz, W. (1983). Numerical and applicational aspects of fuzzy relational equations, *Fuzzy Sets and Systems* **11**: 1–18.

- Pedrycz, W. (1984a). An identification algorithm in fuzzy relational systems, *Fuzzy Sets and Systems* **13**: 153–167.
- Pedrycz, W. (1984b). Identification in fuzzy systems, *IEEE Trans. Systems, Man and Cybernetics* **SMC-14**: 361–366.
- Pedrycz, W. (1985a). Applications of fuzzy relational equations for methods of reasoning in presence of fuzzy data, *Fuzzy Sets and Systems* **16**: 163–175.
- Pedrycz, W. (1985b). On generalized fuzzy relational equations and their applications, *J. Mathematical Analysis and Applications* **107**: 520–536.
- Pedrycz, W. (1985c). Structured fuzzy models, *Int. J. Cybernetics and Systems* **16**: 103–117.
- Pedrycz, W. (1988). Approximate solutions of fuzzy relational equations, *Fuzzy Sets and Systems* **28**: 183–202.
- Pedrycz, W. (1990). Algorithms for solving fuzzy relational equations in a probabilistic setting, *Fuzzy Sets and Systems* **38**: 313–327.
- Pedrycz, W. (1991). Processing in relational structures: Fuzzy relational equations, *Fuzzy Sets and Systems* **40**: 77–106.
- Pedrycz, W. (1993). *Fuzzy Control and Fuzzy Systems*, 2nd edn, Research Studies Press/John Wiley & Sons.
- Pedrycz, W. (1994). Why triangular membership functions?, *Fuzzy Sets and Systems* **64**: 21–30.
- Pedrycz, W. (1995). *Fuzzy Sets Engineering*, CRC Press.
- Pedrycz, W. (1996a). Fuzzy multimodels, *IEEE Trans. Fuzzy Systems* **4**: 139–148.
- Pedrycz, W., Czogała, E. e Hirota, K. (1984). Some remarks on the identification problem in fuzzy systems, *Fuzzy Sets and Systems* **12**: 185–189.
- Pedrycz, W. (ed.) (1996b). *Fuzzy Modelling: Paradigms and Practice*, Kluwer Academic Publishers.
- Pedrycz, W. e Gomide, F. A. C. (1998). *An Introduction to Fuzzy sets. Analysis and Design*, MIT Press.
- Pedrycz, W., Hirota, K. e Sessa, S. (2001). A decomposition of fuzzy relations, *IEEE Trans. Systems, Man and Cybernetics - Part B: Cybernetics* **31**: 657–662.
- Pedrycz, W., Lam, P. C. F. e Rocha, A. F. (1995). Distributed fuzzy system modeling, *IEEE Trans. Systems, Man and Cybernetics* **25**: 769–780.

- Pedrycz, W. e Oliveira, J. V. (1996a). An algorithmic framework for development and optimization of fuzzy models, *Fuzzy Sets and Systems* **80**: 37–55.
- Pedrycz, W. e Oliveira, J. V. (1996b). Optimization of fuzzy models, *IEEE Trans. Systems, Man and Cybernetics* **26**: 627–636.
- Pedrycz, W. e Oliveira, J. V. (1996c). Some design options for optimal fuzzy model I/O interfaces, *Proc. 5th IEEE Internat. Conference on Fuzzy Systems*, New Orleans/USA, pp. 1481–1487.
- Pedrycz, W. e Vasilakos, A. V. (1999). Linguistic models and linguistic modeling, *IEEE Trans. Systems, Man and Cybernetics - Part B: Cybernetics* **29**: 745–757.
- Pinheiro, C. A. M. (2000). *Análise e Projeto de Sistemas de Controle Fuzzy: Uma Abordagem no Domínio da Frequência*, Tese de Doutorado, DCA/FEEC/UNICAMP - Campinas/SP/Brasil.
- Raju, G. U., Zhou, J. e Kisner, R. A. (1991). Hierarchical fuzzy control, *Int. J. Control* pp. 1201–1216.
- Rosenblatt, F. (1959). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books.
- Roubos, H. e Setnes, M. (2001). Compact and transparent fuzzy models and classifiers through iterative complexity reduction, *IEEE Trans. Fuzzy Systems* **9**: 516–524.
- Rovatti, R. e Guerrieri, R. (1996). Fuzzy sets of rules for system identification, *IEEE Trans. Fuzzy Systems* **4**: 89–102.
- Rudin, W. (1964). *Principles of Mathematical Analysis*, 2nd edn, McGraw-Hill.
- Rumelhart, D. E., Hinton, G. E. e Williams, R. J. (1986). *Learning Internal Representations by Error Propagation*, Chap. 8, (D. E. Rumelhart and J. L. McClelland, Eds), MIT Press.
- Sanchez, E. (1976). Resolution of composite fuzzy relation equations, *Information and Control* **30**: 38–48.
- Sentoni, G., Agamennoni, O., Desages, A. e Romagnoli, J. (1996). Approximate models for nonlinear process control, *AIChE Journal* **42**: 2240–2250.
- Sentoni, G. B., Biegler, L. T., Guiver, J. B. e Zhao, H. (1998). State-Space nonlinear process modeling: Identification and universality, *AIChE Journal* **44**: 2229–2239.
- Setnes, M. (2000). Supervised fuzzy clustering for rule extraction, *IEEE Trans. Fuzzy Systems* **8**: 416–424.

- Setnes, M. e Babuška, R. (2001). Rule base reduction: Some comments on the use of orthogonal transforms, *IEEE Trans. Systems, Man and Cybernetics - Part C: Applications and Reviews* **31**: 199–206.
- Setnes, M., Babuška, R., Kaymak, U. e van Nauta Lemke, H. R. (1998). Similarity measures in fuzzy rule base simplification, *IEEE Trans. Systems, Man and Cybernetics - Part B: Cybernetics* **28**: 376–386.
- Setnes, M., Babuška, R. e Verbruggen, H. B. (1998). Rule-Based modeling: Precision and transparency, *IEEE Trans. Systems, Man and Cybernetics - Part C: Applications and Reviews* **28**: 165–169.
- Shimojima, K., Fukuda, T. e Hasegawa, Y. (1995). Self-tuning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm, *Fuzzy Sets and Systems* **71**: 295–309.
- Sing, C. H. e Postlethwaite, B. (1997). pH control: Handling nonlinearity and deadtime with fuzzy relational model-based control, *IEE Proceedings - Control Theory Appl.* **144**: 263–268.
- Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.-Y., Hjalmarsson, H. e Juditsky, A. (1995). Nonlinear black-box modeling in system identification: A unified overview, *Automatica* **31**: 1691–1724.
- Söderström, T. e Stoica, P. (1989). *System Identification*, Prentice Hall.
- Soeterboek, R. (1992). *Predictive Control: A Unified Approach*, Prentice Hall.
- Stamou, G. B. e Tzafestas, S. G. (1999). Fuzzy relation equations and fuzzy inference systems: An inside approach, *IEEE Trans. Systems, Man and Cybernetics - Part B: Cybernetics* **29**: 694–715.
- Sudkamp, T. e Hammell II, R. J. (1994). Interpolation, completion, and learning fuzzy rules, *IEEE Trans. Systems, Man and Cybernetics* **24**: 332–342.
- Sugeno, M. e Kang, G. T. (1986). Fuzzy modelling and control of multilayer incinerator, *Fuzzy Sets and Systems* **18**: 329–346.
- Sugeno, M. e Kang, G. T. (1988). Structure identification of fuzzy model, *Fuzzy Sets and Systems* **28**: 15–33.
- Sugeno, M. e Tanaka, K. (1991). Successive identification of a fuzzy model and its applications to prediction of a complex system, *Fuzzy Sets and Systems* **42**: 315–334.
- Sugeno, M. e Yasukawa, T. (1993). A fuzzy-logic-based approach to qualitative modeling, *IEEE Trans. Fuzzy Systems* **1**: 7–31.

- Tachibana, K. e Furuhashi, T. (1998). Uneven allocation of membership functions for hierarchical fuzzy modeling using genetic algorithm, *Proc. 7th IEEE Internat. Conference on Fuzzy Systems*, Anchorage/USA, pp. 746–751.
- 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.
- Tanscheit, R. (1992). Controle nebuloso, *Anais IX Congresso Brasileiro de Automática - Mini-Cursos*, Vitória-ES/Brasil, pp. 82–95.
- Taub, H. (1984). *Circuitos Digitais e Microprocessadores*, McGraw-Hill do Brasil.
- Teixeira, M. C. M., Pietrobon, H. C. e Assunção, E. (2000). Novos resultados sobre a estabilidade e controle de sistemas não-lineares utilizando modelos fuzzy e LMI, *Controle & Automação* **11**: 37–48.
- Teixeira, M. C. M. e Žak, S. H. (1999). Stabilizing controller design for uncertain nonlinear systems using fuzzy models, *IEEE Trans. Fuzzy Systems* **7**: 133–142.
- Tong, R. M. (1976). Analysis of fuzzy control algorithms using the relation matrix, *Int. J. Man-Machine Studies* **8**: 679–686.
- Tong, R. M. (1978). Synthesis of fuzzy models for industrial processes-some recent results, *Int. J. General Systems* **4**: 143–162.
- Uehara, K. e Fujise, M. (1993). Multistage fuzzy inference formulated as linguistic-truth-value propagation and its learning algorithm based on back-propagating error information, *IEEE Trans. Fuzzy Systems* **1**: 205–221.
- Van den Hof, P. M. J., Heuberger, P. S. C. e Bokor, J. (1995). System identification with generalized orthonormal basis functions, *Automatica* **31**(12): 1821–1834.
- Von Zuben, F. J. (1996). *Modelos Paramétricos e Não-Paramétricos de Redes Neurais Artificiais e Aplicações*, Tese de Doutorado, DCA/FEEC/UNICAMP - Campinas/SP/Brasil.
- Von Zuben, F. J. (2001). Computação evolutiva: Uma abordagem pragmática. Tutorial de Apoio ao Curso IA707 - Computação Evolutiva, DCA/FEEC/Unicamp.
*<http://www.dca.fee.unicamp.br/vonzuben/courses/ia707.html>
- Vázquez, M. . A. e Agamennoni, O. E. (2001). Approximate models for nonlinear dynamical systems and their generalization properties, *Mathematical and Computer Modelling* **33**: 965–986.
- Wahlberg, B. (1991a). Identification of resonant systems using Kautz filters, *Proc. 30th Conference on Decision and Control*, Brighton/England, pp. 2005–2010.

- Wahlberg, B. (1991b). System identification using Laguerre models, *IEEE Trans. on Automatic Control* **36**(5): 551–562.
- Wahlberg, B. (1994). Laguerre and Kautz models, *Proc. IFAC Symposium on System Identification*, Copenhagen/Denmark, pp. 1–11.
- Wahlberg, B. e Ljung, L. (1992). Hard frequency-domain model error bounds from least-squares like identification techniques, *IEEE Trans. on Automatic Control* **37**(7): 900–912.
- Wahlberg, B. e Mäkilä, P. M. (1996). Approximation of stable linear dynamical systems using Laguerre and Kautz functions, *Automatica* **32**(5): 693–708.
- Wan, F. e Wang, L.-X. (2000). Design of economical fuzzy systems using least fuzzy rules, *Proc. 9th IEEE Internat. Conference on Fuzzy Systems*, Houston/USA, pp. 1052–1055.
- 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 Trans. Fuzzy Systems* **7**: 617–624.
- Wang, L.-X. e Mendel, J. M. (1992). Fuzzy basis functions, universal approximation and orthogonal least squares learning, *IEEE Trans. Neural Networks* **3**: 807–814.
- Wellstead, P. E. e Zarrop, M. B. (1991). *Self-Tuning Systems: Control and Signal Processing*, John Wiley & Sons.
- Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, PhD thesis, Harvard University- Cambridge/MA/USA.
- Widrow, B. e Hoff, M. E. (1960). Adaptive switching circuits, *IRE WESCON Convention Record* pp. 96–104.
- Willaeys, D. e Malvache, N. (1981). The use of fuzzy sets for the treatment of fuzzy information by computer, *Fuzzy Sets and Systems* **5**: 323–327.
- Wong, C.-C. e Chen, C.-C. (2000). A GA-based method for constructing fuzzy systems directly from numerical data, *IEEE Trans. Systems, Man and Cybernetics - Part B: Cybernetics* **30**: 904–911.
- Wong, C. H., Shah, S. L. e Fisher, D. G. (2000). Fuzzy relational predictive identification, *Fuzzy Sets and Systems* **113**: 417–426.

- Wray, J. e Green, G. G. R. (1994). Calculation of the Volterra kernels of non-linear dynamic systems using an artificial neural network, *Biological Cybernetics* **71**: 187–195.
- Wu, T.-P. e Chen, S.-M. (1999). A new method for constructing membership functions and fuzzy rules from training examples, *IEEE Trans. Systems, Man and Cybernetics - Part B: Cybernetics* **29**: 25–40.
- Xu, C. W. (1989). Fuzzy systems identification, *IEE Proceedings* **136 - Pt. D**: 146–150.
- Xu, C. W. e Lu, Y. Z. (1987). Fuzzy model identification and self-learning for dynamic systems, *IEEE Trans. Systems, Man and Cybernetics* **SMC-17**: 683–689.
- Yager, R. R. (1998). On the construction of hierarchical fuzzy systems models, *IEEE Trans. Systems, Man and Cybernetics - Part C: Applications and Reviews* **28**: 55–66.
- Yager, R. R. e Filev, D. P. (1994). *Essentials of Fuzzy Modeling and Control*, John Wiley & Sons.
- Yager, R. R., Ovchinnikov, S., Tong, R. M. e Nguyen, H. T. (eds) (1987). *Fuzzy Sets and Applications: Selected Papers by L. A. Zadeh*, John Wiley & Sons.
- Yen, J. e Wang, L. (1999). Simplifying fuzzy rule-based models using orthogonal transformation methods, *IEEE Trans. Systems, Man and Cybernetics - Part B: Cybernetics* **29**: 13–24.
- Yen, J., Wang, L. e Gillespie, C. W. (1998). Improving the interpretability of TSK fuzzy models by combining global learning and local learning, *IEEE Trans. Fuzzy Systems* **6**: 530–537.
- Yi, S. Y. e Chung, M. J. (1993). Identification of fuzzy relational model and its application to control, *Fuzzy Sets and Systems* **59**: 25–33.
- Zadeh, L. A. (1965). Fuzzy sets, *Information and Control* **8**: 338–353.
- Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. Systems, Man and Cybernetics* **SMC-3**: 28–44.
- Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning - Parts 1 and 2, *Information Sciences* **8**: 199–249,301–357.
- Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems* **1**: 3–28.

-
- Zadeh, L. A. (1985). Syllogistic reasoning in fuzzy logic and its application to usability and reasoning with dispositions, *IEEE Trans. Systems, Man and Cybernetics SMC-15*: 754–763.
- Zeng, X.-J. e Singh, M. G. (1994). Approximation theory of fuzzy systems - SISO case, *IEEE Trans. Fuzzy Systems* **2**: 162–176.
- Zeng, X.-J. e Singh, M. G. (1995). Approximation theory of fuzzy systems - MIMO case, *IEEE Trans. Fuzzy Systems* **3**: 219–235.
- Zervos, C. C. e Dumont, G. A. (1988). Deterministic adaptive control based on Laguerre series representation, *Int. J. Control* **48**(6): 2333–2359.