



**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CURITIBA**

GERÊNCIA DE PESQUISA E PÓS-GRADUAÇÃO

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA E INFORMÁTICA INDUSTRIAL - CPGEI**

ELAINE YASSUE NAGAI

**IDENTIFICAÇÃO AUTOMÁTICA DE MODELOS
FUZZY INFERENCIAIS**

TESE DE DOUTORADO

**CURITIBA
DEZEMBRO-2006.**

TESE DE DOUTORADO
apresentada à UTFPR
para a obtenção do título de

DOUTOR EM CIÊNCIAS

por

ELAINE YASSUE NAGAI

**IDENTIFICAÇÃO AUTOMÁTICA DE MODELOS *FUZZY*
INFERENCIAIS**

Banca Examinadora:

Presidente e Orientadora:

Prof^ª. Dra. LÚCIA VALÉRIA RAMOS DE ARRUDA

UTFPR

Examinadores:

Prof^ª. Dra. MARLEY MARIA BERNARDES REBUZZI VELLASCO

PUC-RIO

Prof. Dr. MAURÍCIO FERNANDES FIGUEIREDO

UFSCAR

Prof. Dr. CELSO ANTÔNIO ALVES KAESTNER

PUC-PR

Prof. Dr. FLAVIO NEVES JUNIOR

UTFPR

Prof^ª. Dra. MYRIAM REGATTIERI DE BIASE DA SILVA DELGADO

UTFPR

Curitiba, 11 de Dezembro de 2006.

ELAINE YASSUE NAGAI

IDENTIFICAÇÃO AUTOMÁTICA DE MODELOS
FUZZY INFERENCIAIS

Orientadora:

Prof^ª. Dra. Lúcia Valéria Ramos de Arruda

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI) da Universidade Tecnológica Federal do Paraná (UTFPR), na área de concentração de Informática Industrial, como requisito parcial à obtenção do título de Doutor em Ciências.

Curitiba
2006

Ficha catalográfica elaborada pela Biblioteca da UTFPR – Campus Curitiba

N147i Nagai, Elaine Yassue
“Identificação automática de modelos fuzzy inferenciais” / Elaine Yassue
Nagai. Curitiba. UTFPR, 2006
XV, 143 f. il. ; 30 cm

Orientadora: Profª. Drª. Lúcia Valéria Ramos de Arruda
Tese (Doutorado) – Universidade Tecnológica Federal do Paraná. Curso de
Pós-Graduação em Engenharia Elétrica e Informática Industrial. Curitiba,
2006
Bibliografia f. 133- 143

1. Sistemas difusos. 2. Materiais. 3. Modelos e simulação por computador.
I. Arruda, Lúcia Valéria Ramos de, orient. II. Universidade Tecnológica Federal do Paraná. Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial. III. Título

CDD: 511.313

Agradecimentos

Agradeço, inicialmente, à minha família, por todo afeto, apoio e confiança.

À professora Valéria, pela orientação e atenção que recebi durante todo o processo de desenvolvimento desta tese de doutorado.

Ao professor Flávio, pelo suporte e incentivo oferecido.

Um agradecimento especial à Agência Nacional do Petróleo - ANP - e à Financiadora de Estudos e Projetos - FINEP - que através do Programa de Recursos Humanos para o Setor de Petróleo e Gás - PRH-ANP/MCT (PRH10-UTFPR-PR) disponibilizou, recursos e equipamentos que possibilitaram o desenvolvimento desta tese de doutorado.

Por fim, gostaria de agradecer a todas as pessoas que, ao longo desses anos, contribuíram direta e indiretamente para a minha formação.

Este trabalho contou com o apoio financeiro da Agência Nacional do Petróleo - ANP - e da Financiadora de Estudos e Projetos - FINEP - por meio do Programa de Recursos Humanos da ANP para o Setor de Petróleo e Gás - PRH-ANP/MCT (PRH10-UTFPR).

Sumário

Lista de Figuras	ix
Lista de Tabelas	xi
Resumo	xiii
Abstract	xv
1 Introdução	1
1.1 Objetivo	3
1.2 Estrutura do Trabalho	4
2 Modelos Inferenciais: Uma Visão Geral	7
2.1 Identificação de Sistemas	8
2.2 Aquisição dos Dados	10
2.3 Pré-processamento dos Dados	11
2.4 Seleção das Variáveis de Entrada	12
2.5 Métodos Estatísticos de Inferência	13
2.5.1 Regressão por Componentes Principais - <i>PCR</i>	14
2.5.2 Regressão por Mínimos Quadrados Parciais - <i>PLS</i>	16
2.6 Filtro de Kalman	17
2.6.1 Filtro de Kalman Discreto	18
2.7 Inteligência Computacional	21
2.7.1 Redes Neurais Multicamadas	21
2.7.2 Mapa Auto-Organizável de Kohonen	26
2.7.3 Sistemas <i>Fuzzy</i>	30
2.7.4 Algoritmo <i>Fuzzy C-Means</i> - <i>FCM</i>	38
2.8 Identificação de Modelos Fuzzy	40
2.9 Validação do Modelo	42
2.10 Metodologia Proposta para a Identificação de Modelos <i>Fuzzy</i>	43
3 Estado da Arte	47
3.1 Métodos Estatísticos	47
3.2 Métodos Baseados em Filtro de Kalman	50
3.3 Métodos Baseados em Inteligência Computacional	52
3.4 Abordagens Híbridas	57
3.5 Comentários Finais	59

4	Metodologia Proposta	61
4.1	Descrição do Problema	61
4.2	Seleção das Variáveis de Entrada	62
4.3	Seleção da Ordem do Modelo	66
4.4	Geração da Base de Regras Inicial	70
4.5	Simplificação da Base de Regras Inicial	74
4.6	Tratamento de Regras Inconsistentes	77
4.7	Metodologia de Avaliação dos Resultados	77
5	Coluna de Destilação Binária	79
5.1	Descrição da Coluna de Destilação Binária	79
5.2	Inferência da Composição do Produto de Topo da Coluna Binária	84
5.2.1	Estudo Comparativo para a Inferência do Produto de Topo	90
5.3	Inferência da Composição do Produto de Fundo	93
5.3.1	Estudo Comparativo para a Inferência do Produto de Fundo	99
6	Coluna Fracionadora de Óleo Cru	103
6.1	Descrição da Coluna Fracionadora de Óleo Cru	103
6.2	Caso 01: Simulação com o Conjunto CVS01	109
6.3	Caso 02: Simulação com o conjunto CVS02	116
7	Considerações Finais	123
7.1	Contribuições	129
7.2	Trabalhos Futuros	129
8	Publicações	131
	Referências Bibliográficas	133

Lista de Figuras

2.1	Fluxograma do procedimento de identificação de sistemas.	9
2.2	Filtro de Kalman discreto	18
2.3	Modelo de neurônio proposto por Rosenblatt (1962).	22
2.4	Exemplo de uma rede neural MLP.	23
2.5	Mapa auto-organizável unidimensional.	26
2.6	Algoritmo de aprendizagem do <i>SOM</i>	29
2.7	Diagrama de um sistema fuzzy básico.	31
2.8	Regra se-então.	32
2.9	Modelo <i>fuzzy</i> lingüístico.	33
2.10	Exemplo de regra <i>fuzzy Takagi-Sugeno</i>	35
2.11	Exemplo de regra Singleton.	36
2.12	Algoritmo FCM.	40
2.13	Fluxograma da metodologia proposta para a identificação de modelos fuzzy inferenciais.	44
4.1	Exemplo de um mapa de Kohonen unidimensional.	64
4.2	Procedimento de seleção de variáveis de entrada.	65
4.3	O algoritmo de seleção de variáveis.	66
4.4	Conjunto de regressores.	67
4.5	O algoritmo de seleção da ordem do modelo <i>fuzzy</i>	69
4.6	Projeção dos <i>clusters</i>	70
4.7	Base de regras resultante da projeção.	70
4.8	Função de pertinência do tipo gbell.	72
4.9	O algoritmo da geração da base de regras inicial.	73
4.10	Exemplo de substituição de funções de pertinência similares.	75
4.11	Simplificação da base de regras inicial.	76
4.12	O algoritmo da geração da base de regras inicial.	76
5.1	Coluna de destilação binária.	80
5.2	Variáveis de entrada do modelo da coluna binária (Parte I) $\{ u_1, u_2, u_3, u_4, u_5, u_6 \}$	82
5.3	Variáveis de entrada do modelo da coluna binária (Parte II) $\{ u_7, u_8, u_9, u_{10}, u_{11}, u_{12} \}$	83
5.4	Variáveis de saída da coluna binária: composição do produto de topo (y_1) e de fundo (y_2).	83
5.5	Cálculo da correlação para a inferência da composição do produto de topo (coluna binária).	84

5.6	Seleção de variáveis para a inferência da composição do produto de topo (coluna binária).	86
5.7	Seleção do atraso para a inferência da composição do produto de topo (coluna binária).	87
5.8	Funções de pertinência das variáveis do modelo fuzzy para a inferência da composição do produto de topo y_1 (coluna binária).	88
5.9	Inferência da composição do produto de topo da coluna binária.	90
5.10	Comparações de modelos para a inferência da composição do produto de topo.	92
5.11	Cálculo da correlação para a inferência da composição do produto de fundo (coluna binária).	93
5.12	Seleção de variáveis para a inferência da composição do produto de fundo (coluna binária).	95
5.13	Seleção do atraso para a inferência da composição do produto de fundo (coluna binária).	96
5.14	Funções de pertinência das variáveis do modelo fuzzy para a inferência da composição do produto de fundo (coluna binária).	97
5.15	Inferência da composição do produto de base da coluna binária.	99
5.16	Comparações de modelos para a inferência da composição do produto de topo.	101
6.1	Modelo da coluna fracionadora de óleo cru.	104
6.2	Incertezas nos ganhos ϵ_i , $i = 1, \dots, 5$	107
6.3	Variáveis de entrada do modelo da coluna fracionadora $\{ u_1, u_2, u_3, d_1, d_2 \}$	107
6.4	Variáveis de saída do modelo da fracionadora $\{ y_1, y_2, y_3, y_4, y_5, y_6, y_7 \}$	108
6.5	Cálculo da correlação para o Caso 01 (coluna fracionadora).	109
6.6	Seleção de variáveis para o Caso 01 (coluna fracionadora).	111
6.7	Seleção do atraso para o Caso 01 (coluna fracionadora).	111
6.8	Funções de pertinência das variáveis do modelo fuzzy inferencial para o Caso 01 (coluna fracionadora).	113
6.9	Inferência da composição do produto de topo para o Caso 01 (coluna fracionadora).	114
6.10	Erro absoluto entre a saída simulada e a saída inferida.	115
6.11	Comparação das saídas com o erro absoluto.	115
6.12	Seleção do atraso para o Caso 02 (coluna fracionadora).	117
6.13	Funções de pertinência das variáveis do modelo fuzzy inferencial para o Caso 02 (coluna fracionadora).	118
6.14	Comparação entre a saída do modelo fuzzy inferencial e a saída desejada para o Caso 02 (coluna fracionadora).	120
6.15	Erro absoluto entre a saída simulada e a saída inferida.	120
6.16	Comparação das saídas com o erro absoluto.	120

Lista de Tabelas

5.1	Descrição das variáveis secundárias da coluna binária.	82
5.2	Classificação das variáveis de acordo com o valor absoluto da correlação para a inferência do produto de topo (coluna binária).	84
5.3	Especificação dos subconjuntos de variáveis secundárias para a inferência da composição do produto de topo (coluna binária).	85
5.4	Variáveis selecionadas para a inferência da composição do produto de topo (coluna binária).	89
5.5	Parâmetros do modelo fuzzy para a inferência da composição do produto de topo (coluna binária).	89
5.6	Índices de desempenho do modelo inferencial para estimar a composição do produto de topo (coluna binária).	90
5.7	Matriz de escores \mathbf{T} (produto de topo).	91
5.8	Comparação do desempenho dos modelos PCR6, PCR12, neural e fuzzy.	92
5.9	Classificação das variáveis de acordo com o valor absoluto da correlação para a inferência da composição do produto de fundo (coluna binária).	94
5.10	Especificação dos subconjuntos de variáveis secundárias para a inferência da composição do produto de fundo (coluna binária).	94
5.11	Variáveis selecionadas para a inferência da composição do produto de fundo (coluna binária).	96
5.12	Parâmetros do modelo fuzzy para a inferência da composição do produto de fundo (coluna binária).	98
5.13	Índices de desempenho do modelo inferencial para estimar a composição do produto de fundo (coluna binária).	98
5.14	Matriz de escores \mathbf{T} (produto de fundo).	100
5.15	Comparação do desempenho dos modelos PCR7, PCR10, neural e fuzzy.	101
6.1	Descrição das variáveis da coluna fracionadora.	105
6.2	Modelo discreto da coluna fracionadora.	106
6.3	Descrição das incertezas nos ganhos em regime da coluna fracionadora.	106
6.4	Descrição das variáveis secundárias que compõem o conjunto CVS01 (coluna fracionadora).	109
6.5	Classificação das variáveis de acordo com o valor absoluto da correlação para o Caso 01 (coluna fracionadora).	110
6.6	Especificação dos subconjuntos de variáveis secundárias para o Caso 01 (coluna fracionadora).	110
6.7	Variáveis selecionadas para o Caso 01 (coluna fracionadora).	112

6.8	Parâmetros do modelo fuzzy inferencial obtido para o Caso 01 (coluna fracionadora).	114
6.9	Índices de desempenho do modelo inferencial obtido para o Caso 01 (coluna fracionadora).	115
6.10	Variáveis selecionadas para o Caso 02 (coluna fracionadora).	119
6.11	Parâmetros do modelo fuzzy inferencial obtido para o Caso 02 (coluna fracionadora).	119
6.12	Índices de desempenho do modelo inferencial obtido para o Caso 01 (coluna fracionadora).	119

Resumo

Em muitos processos químicos existe a necessidade do conhecimento de medidas de variáveis cujos valores não são disponibilizados *on-line*, muitas vezes devido ao alto-custo da instalação de sensores para medir estas variáveis, ou porque é necessário realizar testes laboratoriais para a obtenção destas medidas, que em geral são medidas de composição.

Uma solução é o emprego de sensores baseados em *software* (*soft sensors*), também conhecidos como modelos inferenciais, para gerar estimativas destas variáveis. A grande motivação para o uso de modelos inferenciais é que estes podem substituir testes laboratoriais na obtenção de medidas de variáveis não disponíveis *on-line*. Esta disponibilidade *on-line* pode reduzir o atraso nas ações de controle, resultando em um sistema de controle mais efetivo, que possibilita o aumento da qualidade do produto final com um menor custo.

Considerando-se que, em geral, processos químicos possuem alta complexidade, alta dimensionalidade, não-linearidades e comportamento dinâmico, torna-se difícil a modelagem desses processos através de abordagens baseadas em técnicas clássicas de identificação. Na tentativa de superar tais dificuldades, este trabalho propõe uma metodologia para a construção de modelos inferenciais, baseados na teoria dos conjuntos *fuzzy*, a serem utilizados na construção de modelos inferenciais para processos petroquímicos. A construção do modelo *fuzzy* inferencial é realizado através dados amostrados do processo.

Esta metodologia é composta por quatro etapas. A primeira etapa corresponde à identificação da estrutura do modelo *fuzzy*. Inicialmente as variáveis de entrada do modelo são selecionadas, dentre um conjunto de variáveis secundárias disponíveis, aplicando-se mapas de Kohonen. Então, o conjunto de regressores (amostras atrasadas das variáveis de entrada e saída utilizadas na identificação do modelo do processo) é determinado com o auxílio dos quocientes de Lipschitz. Após esta fase inicial, um algoritmo de agrupamento *fuzzy* é aplicado ao espaço de entrada e saída para encontrar uma base de regras inicial do modelo *fuzzy* inferencial. Na terceira etapa o modelo inicial, gerado na fase anterior, passa por um processo de simplificação, onde as funções de pertinência similares são agrupadas com o objetivo de diminuir a complexidade e a carga computacional do modelo final, tornando-o mais simples e conciso. Finalizando a metodologia, a existência de regras inconsistentes é verificada e tratada. O modelo *fuzzy* obtido passa, então, por uma fase de validação, onde é verificado o seu desempenho na geração das estimativas. A aplicabilidade da metodologia proposta é verificada através da construção de modelos *fuzzy* inferenciais para processos petroquímicos simulados.

Palavras-chave: agrupamento *fuzzy*, coluna de destilação, estimadores inferenciais, mapas de Kohonen, modelagem *fuzzy*, *soft sensing*.

Abstract

In many chemical processes it is essential to know variable measurements that are not online available, sometimes due to the high cost of the hardware sensors installation, used to measure the variable values, or because it is necessary laboratorial tests for obtaining these measurements, that in general are product composition measurements.

An option is to use sensors based in software (or soft sensors), also known as inferential models, to generate the variable estimates. The main motivation for inferential models usage is that they can substitute laboratorial tests in the obtaining of the variables measurement that are not online available. This online availability can reduce the control action delays, resulting in a more effective control system, that can increase the final product quality with a smaller cost.

Generally, chemical processes present high complexity, high dimensionality, nonlinearities and dynamic behavior. These characteristics become the modelling a hard task when classical techniques of identification are considered. Trying to overcome difficulties, this work proposes a construction methodology of inferential models based on fuzzy sets theory, that can be used to build inferential models for petrochemical processes. The inferential fuzzy model is constructed by means of a data set sampled from the process.

This methodology is composed by four stages. The first stage corresponds to the identification of the fuzzy model structure. Initially, the model input variables are selected, among a group of available secondary variables, by applying Kohonen maps. Then, the regressors set (the lagged samples of the input and output variables used in the process model identification) is stipulated by means of the Lipschitz quotients. After this initial phase, a fuzzy clustering algorithm is applied to find an initial rule base. In the third stage, the initial model, generated in the previous phase, is simplified by merging the similar membership functions, as a way to reduce the complexity and computational effort of the final model, that becomes simpler and concise. Concluding the methodology, the inconsistent rules existence is examined and treated. At last, in the validation phase, the performance of the obtained fuzzy model in the generation of the estimates is verified. The applicability of the proposed methodology is investigated through the construction of inferential fuzzy models for simulated petrochemical processes.

Keywords: fuzzy clustering, distillation columns, inferential estimators, Kohonen maps, fuzzy modelling, soft sensing.

Capítulo 1

Introdução

Métodos de controle *on-line* e otimização dependem diretamente da qualidade das medições das variáveis do processo em questão. Porém, em muitos casos, a medição infreqüente de algumas variáveis do processo impede a detecção antecipada de distúrbios, impossibilitando uma ação imediata, ou mesmo antecipada, de controle. Muitas vezes, esses efeitos adversos não podem ser aceitavelmente recuperados, mesmo com o uso de algoritmos existentes de controle avançado, e podem levar a um desempenho insatisfatório (Marlin, 1995). Esta indisponibilidade pode ser atribuída a diversos fatores, dentre os quais o alto custo da instalação de sensores adicionais, que pode não ser economicamente interessante.

Em tais casos, uma estimativa das variáveis principais (e.g. variáveis de medidas de composição) pode ser obtida através de variáveis secundárias, ou seja, através de um *modelo inferencial* (Bhartiya & Whiteley, 2001), também conhecido como estimador, *soft sensor* (*software sensor* - sensor baseado em *software*) ou ainda sensor virtual. O controle realizado através de informações provenientes de um modelo inferencial é referenciado na literatura como *controle inferencial*.

Na indústria petroquímica, modelos inferenciais podem ser utilizados para estimar a composição de produtos a partir de medidas de temperatura e de outras variáveis secundárias disponíveis *on-line*. Porém, considerando-se que, na maioria dos casos, esses processos possuem alta complexidade e dimensionalidade, além de apresentarem um comportamento não-linear e/ou variante no tempo, a modelagem destes processos torna-se difícil quando são empregadas abordagens teóricas baseadas em filtro de Kalman ou outras técnicas clássicas de estimação (Zambrogna et al., 2005; Tham et al., 1991).

De modo a lidar com a complexidade de processos, abordagens alternativas ba-

seadas em técnicas de inteligência computacional têm sido recentemente propostas (Qin, 1996; Rallo et al., 2002; Fabro, 2003), nas quais foram relatadas aplicações de redes neurais artificiais (*ANN - Artificial Neural Networks*) (Fausett, 1994) e de mínimos quadrados parciais (*PLS - Partial Least Squares*) (Wold, 1966) no desenvolvimento de modelos inferenciais para diferentes processos.

Sistemas *fuzzy* (Zadeh, 1965; Zadeh, 1973) têm sido utilizados de forma bem sucedida em um crescente número de áreas de aplicação, principalmente devido ao fato de fazerem uso de regras lingüísticas como forma de representação do conhecimento. Esses sistemas baseados em regras mostram-se apropriados para descrever processos quando a alta complexidade do sistema e/ou a indisponibilidade de conhecimento detalhado de seu comportamento impedem que uma modelagem matemática adequada seja realizada através de técnicas clássicas. Nestes casos, é possível gerar um modelo *fuzzy* do processo considerado.

Em geral, o modelo *fuzzy* é gerado através de conhecimento especialista, entretanto, quando consideram-se processos complexos, compostos por muitas variáveis, torna-se inviável a definição do modelo *fuzzy* utilizando-se somente de conhecimento especialista. Nestes casos, um modo automatizado para identificar modelos *fuzzy* torna-se mais apropriado.

Em aplicações práticas, as maiores dificuldades na identificação de modelos *fuzzy* são: a determinação da estrutura do modelo e a obtenção da base de regras que descreva o comportamento do sistema de forma apropriada. Diversos elementos devem ser especificados na realização destas tarefas, tais como: o conjunto de variáveis de entrada e saída, número de regras, número e formato das funções de pertinência, interpretação operacional de regras e operadores de inferência. Quanto mais complexa a aplicação envolvida, maiores serão os esforços necessários para se definir os elementos adequadamente, ou seja, aqueles que correspondam a um modelo *fuzzy* que satisfaça aos índices de desempenho desejados. Na literatura atual, as metodologias existentes para a identificação de modelos *fuzzy* não são muitas, principalmente quando a única informação disponível provém de um conjunto de dados amostrados (Wang & Mendel, 1991; Babůska et al., 2000; Ohtake et al., 2001; Roubos & Setnes, 2001; Gao et al., 2001; Delgado et al., 2004).

Uma abordagem bastante utilizada é a de tentativa e erro. Dois fatores associados aos sistemas *fuzzy* têm contribuído para essa prática. Primeiramente, as regras *fuzzy* podem ser geradas por especialistas na forma de regras lingüísticas de modo fácil e direto; segundo, o desempenho de um sistema *fuzzy* não sofre degradação

crítica devido à identificação não-ótima de seus parâmetros (Hwang et al., 2000). Esses fatores tornam o método de tentativa e erro muito prático em determinadas circunstâncias. Porém, essa abordagem não é apropriada ou até mesmo possível nos casos em que não existe a disponibilidade de conhecimento lingüístico ou quando o conhecimento deve ser extraído de dados amostrados do sistema (Figueiredo & Gomide, 1999; Nie, 1995; Pedrycz, 1993).

Nestes casos, uma abordagem simples é usar algoritmos de agrupamento como, por exemplo, os algoritmos de agrupamento Gustafson-Kessel (Gustafson & Kessel, 1979) e os algoritmos de agrupamento *Fuzzy C-Means (FCM)* (Bezdek, 1987), ou métodos similares para particionar o espaço padrão em vários subespaços (chamados de grupos ou *clusters*) com ou sem sobreposição entre eles, e então mapear o centro de cada um dos *cluster* especificados em uma regra de acordo com as definições das variáveis *fuzzy* (Abe & Lan, 1995; Setnes, 2000).

Os algoritmos de agrupamento são, na maioria das aplicações, utilizados para identificar modelos *fuzzy* do tipo Takagi-Sugeno. Neste tipo de modelo *fuzzy*, o conseqüente da regra *fuzzy* é uma função, em geral, uma função linear das variáveis da parte antecedente da regra. A grande desvantagem desse tipo de modelo *fuzzy* é a perda da interpretabilidade das regras, com o uso de funções no conseqüente ao invés de proposições *fuzzy*, como é o caso dos modelos *fuzzy* lingüísticos. O modelo lingüístico possibilita a compreensão do processo modelado através de suas regras lingüísticas.

Ao contrário dos modelos *fuzzy* Takagi-Sugeno, o número de abordagens para a identificação de modelos *fuzzy* lingüísticos é bastante restrito. Neste contexto, este trabalho propõe uma metodologia para a identificação de modelos *fuzzy* lingüísticos, com o intuito de utilizá-los como modelos inferenciais.

1.1 Objetivo

O principal objetivo deste trabalho é propor e desenvolver uma metodologia de identificação de modelos *fuzzy* lingüísticos inferenciais. Esta metodologia é composta por quatro etapas. A primeira etapa corresponde à identificação da estrutura do modelo *fuzzy*. Inicialmente as variáveis de entrada do modelo são selecionadas, dentre um conjunto de variáveis secundárias disponíveis, aplicando-se mapas de Kohonen. Então o conjunto de regressores (amostras atrasadas das variáveis de entrada e saída utilizadas na identificação do modelo do processo) é determinado com o auxílio dos

quocientes de Lipschitz. Após esta fase inicial, um algoritmo de agrupamento *fuzzy* é aplicado ao espaço de entrada e saída para encontrar uma base de regras inicial do modelo *fuzzy* inferencial. Na terceira etapa o modelo inicial, gerado na fase anterior, passa por um processo de simplificação, onde as funções de pertinência similares são agrupadas com o objetivo de diminuir a complexidade e a carga computacional do modelo final, tornando-o mais simples e conciso. Finalizando a metodologia, a existência de regras inconsistentes é verificada e tratada. O modelo *fuzzy* obtido passa, então, por uma fase de validação, onde é verificado o seu desempenho na geração das estimativas.

1.2 Estrutura do Trabalho

Este trabalho está organizado como segue. O capítulo 2 apresenta os principais tópicos que devem ser considerados na construção de modelos inferenciais, assim como uma explanação sobre algumas técnicas empregadas para o desenvolvimento desses modelos.

O estado da arte referente ao desenvolvimento de modelos inferenciais é apresentado no capítulo 3, onde as diferentes propostas foram agrupadas de acordo com o método de inferência empregado.

O capítulo 4 descreve a metodologia proposta neste trabalho para o desenvolvimento de modelos inferenciais.

A partir do capítulo 5 são apresentados os resultados obtidos com a aplicação da metodologia proposta na construção de *soft sensors* para inferir medidas de composição de produtos em dois processos simulados da indústria petroquímica. O capítulo 5 apresenta o primeiro processo, uma coluna de destilação binária, simulada através do simulador comercial HYSYS[®] (HYSYS, 1989). Para este processo, foram construídos dois *soft sensors*, um para inferir a composição do produto de topo da coluna e outro para a inferência do produto de fundo. Para fins de comparação, três modelos inferenciais (adicionais) foram implementados: dois modelos baseados em análise de componentes principais (*PCA - Principal Components Analysis*) (Pearson, 1901; Hotelling, 1933) e um terceiro modelo baseado em redes neurais, identificados com os mesmos dados utilizados para a identificação do *soft sensor* (para o produto de topo e de fundo).

O capítulo 6 apresenta os resultados para o segundo processo, uma coluna fracionadora de óleo cru, conhecida como processo *Shell*, que é um *benchmark* da área de

controle de processos proposto por Prett & Morari (1987), utilizado principalmente para testar e comparar novas estratégias de controle (Prett & García, 1988; Ansari & Tadó, 2000; Tyagunov, 2004).

O capítulo 7 apresenta as considerações finais decorrentes do desenvolvimento deste trabalho e os pontos que ainda podem ser desenvolvidos em trabalhos futuros.

Finalizando, o capítulo 8 lista as publicações geradas a partir do desenvolvimento deste trabalho.

Capítulo 2

Modelos Inferenciais: Uma Visão Geral

Um dos principais objetivos das indústrias petroquímicas é operar a planta de forma que o produto final atenda aos padrões de qualidade especificados sob o menor custo possível. Portanto, é necessária a implementação de estratégias de monitoramento e controle capazes de realizar diversas operações, tais como otimização da produção, monitoramento, previsão e detecção de falhas. Porém, a dificuldade no desenvolvimento destas estratégias é diretamente proporcional à complexidade da planta em questão. Além do mais, os índices de qualidade do produto estão relacionados ao uso final do produto e, em geral, não podem ser medidos em tempo real devido ao alto custo dos equipamentos de medição *on-line*, cujas técnicas de medição costumam ser lentas e complicadas, ou simplesmente devido à incapacidade para medir os índices de qualidade até que o produto final seja formulado e utilizado (Ogunnaike, 1995).

Na tentativa de contornar este problema, várias pesquisas são realizadas em torno do desenvolvimento de técnicas alternativas capazes de reduzir o custo de sensores *on-line*. Estas técnicas alternativas são, em geral, baseadas em algoritmos que emulam o comportamento de sensores físicos, disponibilizando em tempo real estimativas das variáveis primárias (e.g. variáveis de qualidade ou de composição não disponíveis *on-line*) a partir de variáveis secundárias (e.g. temperatura, pressão). Estes algoritmos são chamados de *modelos inferenciais* ou *soft sensors* (sendo também referenciada na literatura como estimação inferencial o processo de inferir as variáveis primárias a partir das variáveis secundárias) (Tham et al., 1991).

Reatores de polimerização, por exemplo, pertencem a uma classe de processos na qual muitas de suas variáveis essenciais, relacionadas à qualidade do produto, não po-

dem ser medidas, ou podem ser medidas a uma baixa taxa de amostragem e com um tempo de atraso significativo. Tais processos podem ser extremamente beneficiados com o uso da estimação inferencial. A carência de medidas facilmente disponíveis, freqüentes e *on-line*, através das quais as propriedades do polímero possam ser inferidas, há muito motiva um considerável esforço de pesquisa na tentativa de solucionar este problema. Uma das linhas de pesquisa mais consolidadas é o desenvolvimento de estimadores de estado que são capazes de estimar propriedades não-medidas do polímero através de medidas prontamente disponíveis (MacGregor, 1986).

Modelos inferenciais podem ser utilizados de forma isolada, como uma alternativa de baixo custo para os sensores usuais, ou ainda podem ser empregados em conjunto com sensores físicos gerando, assim, técnicas “baseadas em modelo” para o desenvolvimento de esquemas de detecção de falhas e inspeção do funcionamento desses sensores físicos. Além disso, esses modelos podem substituir temporariamente os sensores físicos enquanto estes são reparados, a fim de manter as malhas de controle em funcionamento de forma adequada (Rizzo & Xibilia, 2002).

Nas próximas seções deste capítulo serão discutidos alguns dos tópicos relacionados ao desenvolvimento de modelos inferenciais. Este procedimento de desenvolvimento é semelhante ao procedimento de identificação, onde a seqüência de passos é basicamente a mesma, como pode ser observado através das próximas seções.

2.1 Identificação de Sistemas

O fluxograma contido na Figura 2.1¹ ilustra a seqüência básica de passos a serem seguidos durante o processo de identificação de sistemas. Em termos gerais, um experimento de identificação é realizado pela excitação do sistema, utilizando-se algum sinal de entrada, tal como uma função degrau, um ruído branco, uma seqüência binária pseudo-aleatória (*PRBS - Pseudo Random Binary Sequence*), uma seqüência aleatória com instantes não-periódicos de transição (*LCRI - Level Change at Random Instances*), entre outros, e observando-se as suas entradas e saídas durante um intervalo de tempo. A escolha deste sinal deve assegurar que todos os modos dinâmicos do sistema sejam excitados a fim de garantir uma boa identificação.

Processos grandes e complexos contêm, em geral, um determinado número de variáveis controladas e um grande número de variáveis medidas. Se uma dada variável y não pode ser medida (disponibilizada *on-line*), ela pode ser inferida utilizando-se

¹Figura obtida de Soderström & Stoica (1989).

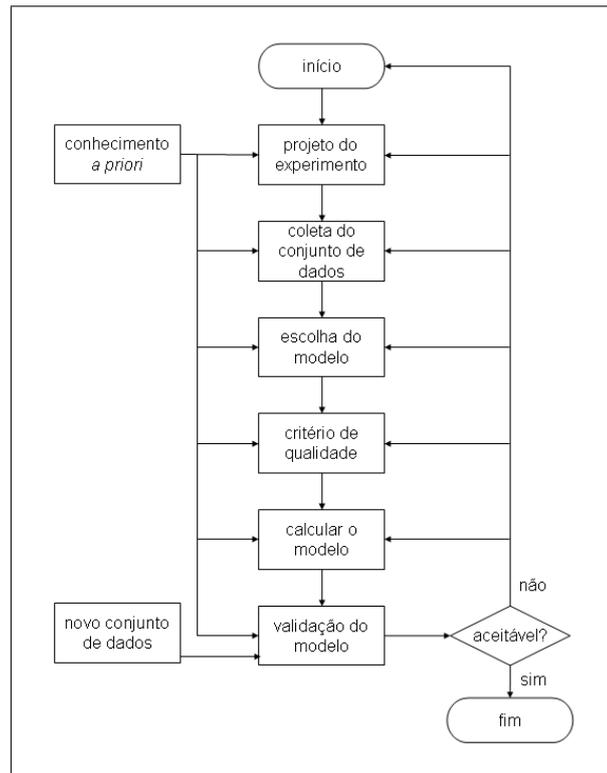


Figura 2.1: Fluxograma do procedimento de identificação de sistemas.

um subconjunto adequado com n variáveis, $\{x_i, i = 1, \dots, n\}$, selecionadas a partir do conjunto de p variáveis secundárias ($n \leq p$) (Bhartiya & Whiteley, 2001).

Os valores dessas variáveis são, geralmente, armazenados em arquivos para um subsequente processamento de informações. Tenta-se, então, adaptar um modelo para o sistema em questão, de acordo com as seqüências de entrada e saída amostradas do processo (registradas). O primeiro passo é determinar uma forma apropriada para o modelo (p.ex. representação através de modelo FIR - *Finite Impulse Response* (Ljung, 2000)). O segundo passo é estimar os parâmetros desconhecidos deste modelo (tais como os coeficientes dos polinômios que compõem o modelo), fazendo uso de algum método de identificação.

Na prática, a estimação da estrutura e dos parâmetros é geralmente realizada de forma iterativa, ou seja, uma estrutura provisória é escolhida e os parâmetros correspondentes são estimados. O modelo obtido é, então, validado de forma a verificar se este modelo provisório é uma representação apropriada para o sistema. Se este modelo provisório não corresponder aos critérios de desempenho desejados, uma estrutura mais complexa para o modelo deve ser considerada, e todos os passos subsequentes devem ser executados até que um modelo válido seja alcançado (Soderström & Stoica, 1989).

Tratando-se de modelos inferenciais, a seqüência de passos é basicamente a mesma, com algumas diferenças, como o passo para a seleção das variáveis de entrada do modelo, caso seja necessário, e também a seleção da metodologia da inferência, onde o tipo do modelo que será utilizado durante a inferência é estipulado, p.ex. filtro de Kalman, modelos estatísticos ou modelos baseados em técnicas de inteligência computacional, através do qual as estimativas serão geradas. Portanto, o desenvolvimento de um modelo inferencial envolve os seguintes tópicos, que serão abordados nas seções subseqüentes:

- Aquisição de dados;
- Pré-processamento dos dados;
- Seleção das variáveis de entrada do modelo;
- Escolha da metodologia de inferência;
- Desenvolvimento do modelo;
- Validação do modelo.

2.2 Aquisição dos Dados

Na maioria dos processos industriais, o monitoramento do estado do processo é realizado através de sensores físicos das variáveis do processo, tais como temperatura, pressão, taxas de vazão. Tais sensores fornecem informações que possibilitam o controle do processo, e também permitem que o histórico de cada variável monitorada seja armazenado. Algumas variáveis, especialmente as variáveis de qualidade, não possuem sensores *on-line* devido ao alto custo ou até mesmo à impossibilidade de implantação do sensor. Desta forma, estas variáveis são, em geral, obtidas *off-line* com um determinado intervalo de tempo. Com o emprego de testes laboratoriais, o atraso (*time-delay*) pode chegar a dez horas (Qin, 1996). Um modo alternativo de se obter as variáveis de qualidade *on-line* é construir um modelo inferencial para inferir estas variáveis de qualidade através das variáveis secundárias (variáveis disponíveis *on-line*) do processo. Para isto, parte-se do pressuposto que as variáveis de qualidade possuem algum relacionamento funcional com as variáveis secundárias do processo, sendo que este relacionamento pode ser linear ou não-linear. Esta é uma das características do processo que definirá a melhor metodologia a ser empregada para a inferência dessas variáveis de qualidade.

2.3 Pré-processamento dos Dados

Os dados amostrados do processo podem conter observações inconsistentes com as características estatísticas do restante dos dados. Essas observações aberrantes (*outliers*) podem ter uma influência indesejada nas estimativas do modelo. Desta forma, o primeiro passo no pré-processamento é identificar e tratar os *outliers*. Esses *outliers* podem ser resultantes de falhas nos sensores, erro de leitura nos testes de laboratório, entre outros fatores adversos. Alguns *outliers* podem ser dados normais, que representam informações importantes, sendo necessário, nestes casos, o uso de conhecimento adicional sobre o processo para diferenciá-los dos outros *outliers* (Qin, 1996).

Alguns *outliers* são tão óbvios que podem ser identificados apenas analisando-se as leis físicas e através de conhecimento *a priori*. Por exemplo, é comum estabelecer limites mínimo e máximo para as variáveis do processo, baseando-se em experiência de um operador do processo. Desta forma, *outliers* podem ser identificados verificando-se a existência de valores que ultrapassem os limites impostos para cada variável. Neste caso, esses *outliers* podem ser substituídos pelo valor máximo ou mínimo, ou pelo valor da média da variável em questão. Por outro lado, muitos outros *outliers*, apesar de estarem dentro dos limites da variável, acarretam consideráveis erros de modelagem. Um modo de identificá-los é verificar a existência de correlação entre as variáveis do processo, e se todos os dados coletados respeitam esta correlação. Se alguma amostra não respeitar a correlação, então ela é considerada um *outlier*. Esta verificação pode ser realizada através de métodos estatísticos, tais como *PCA* e *PLS*.

Dados faltantes (*missing values*) são muito comuns em base de dados históricos do processo e também de dados de testes de laboratório (Qin, 1996). Existem algumas técnicas para lidar com esses dados. O modo mais simples é remover as amostras que possuem dados faltantes. Porém, se são poucos os dados disponíveis, esta técnica não é a mais interessante. Neste caso, é possível aplicar uma interpolação para repor os dados faltantes. Dayal et al. (1992) aplicam uma interpolação linear quando existem menos do que três pontos faltantes consecutivos. Entretanto, quando existem mais do que três pontos consecutivos, a aplicação de interpolação não é confiável, e técnicas mais avançadas devem ser empregadas (Qin, 1996).

Caso os dados faltantes sejam distribuídos de maneira uniforme no conjunto de dados, métodos estatísticos, tais como regressão por componentes principais *PCR* - *Principal Component Regression* (Geladi & Kowalski, 1986) e *PLS*, podem ser

utilizados, pois estes são capazes de trabalhar com conjuntos de dados que possuem dados faltantes (Geladi & Kowalski, 1986). Estes métodos realizam cálculos dos componentes principais ou dos fatores principais e então constroem modelos de regressão baseado nesses componentes principais.

Também durante esta fase é realizada a normalização dos dados, de modo que eles apresentem média zero e desvio padrão unitário. Os dados também podem ser transformados para conter os seus valores dentro de um intervalo pré-estabelecido.

2.4 Seleção das Variáveis de Entrada

Uma base de dados industrial fornece todas as variáveis que podem ser registradas através de sensores físicos. Porém, na maioria dos casos, nem todas essas variáveis são relevantes para a inferência das variáveis principais (Qin, 1996). Muitas vezes, o uso dessas variáveis irrelevantes durante o processo de inferência pode acarretar em erros *mismatch*² substanciais, pois elas, além de não contribuírem, comportam-se como se fossem um ruído. Portanto, é fundamental que sejam selecionadas as variáveis de entrada, dentre as variáveis secundárias disponíveis, que sejam realmente relevantes para a inferência das variáveis principais. Além do conhecimento *a priori*, algumas abordagens podem ser utilizadas para realizar a seleção dessas variáveis de entrada do modelo.

Basicamente, cada variável do processo utilizada no modelo contribui com um determinado valor na inferência e com uma variância associada. Quando a contribuição da variável é maior do que a variância associada, então esta variável é útil para o modelo, caso contrário, ela torna-se prejudicial ao modelo (Hocking, 1976). Portanto, a seleção pode ser realizada verificando-se a contribuição ou relevância de cada variável em relação à variável a ser inferida. Neste caso, dois esquemas são normalmente utilizados: a seleção para frente, onde inicia-se com um pequeno número de variáveis e acrescentam-se mais variáveis, se estas contribuírem para a inferência; e a eliminação para trás, onde inicia-se com todas as variáveis e eliminam-se variáveis que não são relevantes para a inferência.

Uma abordagem de análise de sensibilidade para a seleção de variáveis, em um esquema baseado em redes neurais e *PLS*, é proposta em Qin & McAvoy (1992). Nesta abordagem, uma análise de sensibilidade das saídas em relação às entradas do modelo é realizada sobre a região de operação de onde esses dados foram coletados.

²Erro entre o modelo identificado e o processo real.

Se a sensibilidade de uma variável de saída com respeito a uma variável de entrada é muito pequena, então a variável de entrada é removida do modelo, pois a sua contribuição é pequena no processo de inferência.

A análise de sensibilidade fornece um meio para identificar as variáveis menos sensíveis. A sensibilidade é uma medida da importância da variável de entrada na explicação da variável de saída (Qin, 1996). Para determinar se uma variável realmente contribui para a variável de saída, o método de validação cruzada (*cross-validation*) e o erro de predição obtido com os dados de validação são calculados antes e após a remoção da variável. Se o erro não aumenta após a remoção da variável, então esta será removida do modelo, caso contrário, esta variável será utilizada na construção do modelo final. Com este método é possível alcançar uma maior precisão ao remover as variáveis desnecessárias, mantendo somente aquelas que são realmente relevantes ao modelo. Em Qin & McAvoy (1992) esta análise é utilizada para dimensionar a camada de entrada de uma rede neural que funciona como um modelo inferencial.

Existe, geralmente, alta correlação ou redundância entre as variáveis do processo. Algumas correlações são devido ao fato de que um grupo de variáveis afetam umas às outras, ou podem ser devido também à variabilidade insuficiente durante a operação normal do processo. Em muitas situações, porém, não é desejável remover variáveis que são correlacionadas, pois essa correlação oferece a redundância necessária para a reposição de valores faltantes e para reduzir os erros grosseiros (Kramer, 1992). Entretanto, variáveis correlacionadas implicam no problema de colinearidade. Em regressão estatística linear, tais como a abordagem dos mínimos quadrados ordinários, a colinearidade pode acarretar em uma solução mal-condicionada, onde o modelo resultante apresenta uma grande variância de predição. Neste caso, outras abordagens de regressão, tais como *PCR* e *PLS* são mais usualmente empregadas (Qin, 1996).

Assim, a seleção de variáveis que compõem o modelo inferencial deve levar em consideração o compromisso entre obter um conjunto de variáveis de entrada suficientemente pequeno para reduzir o ruído e grande o bastante para recuperar os possíveis dados faltantes.

2.5 Métodos Estatísticos de Inferência

Em muitos processos industriais as variáveis que se deseja controlar são, em geral, medidas de forma infreqüente (e.g. *off-line* em um laboratório de controle de quali-

dade). Nessas situações, o uso de técnicas avançadas de controle ou otimização requer que as medidas de qualidades sejam inferidas através da análise de correlações. Para processos “bem compreendidos”, a estrutura da correlação, assim como a escolha das variáveis de entradas são conhecidas *a priori*. Porém, a maioria dos processos industriais são tão complexos que a forma apropriada da correlação e a escolha das variáveis de entradas não é muito óbvia (Bhartiya & Whiteley, 2001). Nesses casos, o emprego de métodos estatísticos apresenta-se como uma boa alternativa para resolver o problema de correlação.

Recentemente, métodos estatísticos multivariáveis, baseados em projeção linear, tais como *PCA* e *PLS*, têm atraído grande interesse como métodos robustos para a construção de modelos, particularmente quando os dados apresentam alta dimensionalidade e colinearidades. Especialmente o *PLS* e suas variações têm sido aplicados em muitos problemas práticos de regressão em Engenharia Química, tais como estimação de composição de destilação (Mejdell & Skogestad, 1991a; Kresta et al., 1994) e estimação de variáveis de qualidade de polímeros (Skagerberg et al., 1992).

Embora estes métodos de projeção linear possam manipular alta dimensionalidade e colinearidade, eles são capazes de extrair somente informações lineares a partir dos dados. Considerando-se que muitos processos químicos são não-lineares, seria desejável um método robusto que pudesse modelar qualquer não-linearidade (Park & Han, 2000). Atualmente, existem alguns métodos não-lineares disponíveis, baseados em redes neurais e *PLS* não-linear (Malthouse et al., 1997; Frank & Lanteri, 1988; Wold et al., 1989; Qin & McAvoy, 1992).

Esta seção apresenta os dois principais métodos estatísticos de inferência, *PCR* e *PLS*.

2.5.1 Regressão por Componentes Principais - *PCR*

A Regressão nos Componentes Principais, ou *PCR* (*Principal Component Regression*), utiliza os resultados da Análise dos Componentes Principais (*Principal Components Analysis - PCA*) (Pearson, 1901; Hotelling, 1933) para realizar a regressão.

A *PCA* foi desenvolvida primeiramente por Pearson (1901) e posteriormente reformulada por Hotelling (1933). Esta análise tem a finalidade de substituir um conjunto de variáveis correlacionadas por um conjunto de novas variáveis não-correlacionadas, sendo essas as combinações lineares das variáveis originais e ordenadas de forma decrescente por suas variâncias, ou seja, a *PCA* consiste em reescrever

as coordenadas de um conjunto de dados em um outro sistema de eixos que seja mais conveniente para a análise desses dados. Estas novas coordenadas, chamadas de componentes principais, são o resultado da combinação linear das variáveis originais e são representadas sobre eixos ortogonais, sendo obtidas em ordem decrescente de variância. Portanto, a primeira componente principal detém mais informação sobre os dados do que a segunda componente principal que não detém informações contabilizadas anteriormente (na primeira componente principal) e assim sucessivamente. Em função da ortogonalidade dos eixos, as componentes principais não são correlacionadas (Mardia et al., 1980).

Para seguir a notação de *PCA* e *PLS*, os dados de entrada e saída são arranjados em duas matrizes, \mathbf{X} e \mathbf{Y} , respectivamente. A idéia básica do *PCA* é modelar o bloco de dados \mathbf{X} através de suas componentes ortogonais (Kaspar & Ray, 1992), ou seja, tal método modela o conjunto de dados \mathbf{X} através dos autovalores e autovetores da matriz de covariância $\mathbf{X}^T\mathbf{X}$.

A *PCR* utiliza a projeção da matriz de variáveis independentes \mathbf{X} na matriz de pesos \mathbf{P} . As colunas de \mathbf{P} são os autovetores de $\mathbf{X}^T\mathbf{X}$ colocados em ordem decrescente dos valores em módulo dos autovalores correspondentes (Geladi & Kowalski, 1986). Esta projeção é dada pela equação (2.1):

$$\mathbf{T} = \mathbf{XP}. \quad (2.1)$$

A matriz dos escores, \mathbf{T} , é uma representação da matriz \mathbf{X} , através das componentes principais de \mathbf{X} , onde a primeira coluna da matriz \mathbf{T} corresponde aos valores da primeira componente principal, e assim por diante. Portanto, a redução da dimensionalidade da matriz \mathbf{X} ocorre ao desconsiderar-se as colunas de \mathbf{T} cujos valores não são significativos. Isto significa excluir as dimensões que possuem autovalores pequenos (Besley et al., 1980).

A partir da matriz \mathbf{T} , é possível obter uma estimativa, $\hat{\mathbf{Y}}$, da matriz \mathbf{Y} , através da *PCR*, como na equação (2.2):

$$\hat{\mathbf{Y}} = \mathbf{TB} + \mathbf{E}, \quad (2.2)$$

cuja solução é dada pela equação (2.3):

$$\hat{\mathbf{B}} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{Y}, \quad (2.3)$$

onde $\hat{\mathbf{B}}$ é uma estimativa da matriz \mathbf{B} , \mathbf{Y} é a matriz de variáveis dependentes e \mathbf{E} é a matriz de resíduos.

A *PCR* resolve o problema de colinearidade, garantindo uma matriz inversível para o cálculo de $\hat{\mathbf{B}}$, com a habilidade de eliminar os componentes principais cujos autovalores têm menor módulo. Porém, trata-se de um método com dois passos e, deste modo, corre-se o risco de eliminar informações importantes ao descartar algum componente principal, assim como podem ser mantidos componentes principais com ruído utilizados na regressão (Geladi & Kowalski, 1986).

Visto que os componentes principais são ortogonais, eles podem ser utilizados para analisar as características dos dados e para monitorar as modificações no processo. Uma limitação é que o *PCA* foca-se somente sobre a variância das entradas, ignorando as possíveis correlações entre as entradas e as saídas. Uma componente que é não-principal na análise *PCA* pode ser significativa na explicação da saída. Esta situação pode acontecer quando algumas variáveis de entrada carregam muita variância mas têm uma pequena contribuição para a saída, e outras variáveis carregam menos variância mas apresentam uma significativa contribuição para a saída (Qin, 1996).

2.5.2 Regressão por Mínimos Quadrados Parciais - *PLS*

O método de regressão por Mínimos Quadrados Parciais, ou *Partial Least Squares - PLS*, foi inicialmente apresentado por Wold (1966). Este método tem como objetivo resolver problemas com alta dimensionalidade e variáveis altamente correlacionadas via regressão a partir de um pequeno número de variáveis ortogonais, chamadas variáveis latentes (Geladi & Kowalski, 1986). Essas variáveis latentes são uma combinação linear das variáveis originais medidas (variáveis independentes \mathbf{X}), e o modelo inferencial é obtido a partir dessas variáveis latentes (\mathbf{T}). Desta forma, a matriz \mathbf{X} pode ser descrita como na equação (2.4) (Kaspar & Ray, 1992):

$$\mathbf{X} = \mathbf{TP}^T + \mathbf{E}, \quad (2.4)$$

onde \mathbf{E} é a matriz de resíduo de \mathbf{X} ; e \mathbf{P} é a matriz do pesos de \mathbf{X} . A representação reduzida (através de variáveis latentes) também pode ser feita para as variáveis dependentes (matriz \mathbf{Y}). Da mesma forma, a matriz \mathbf{Y} pode ser escrita como na equação (2.5) (Kaspar & Ray, 1992):

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{F}, \quad (2.5)$$

onde \mathbf{U} é a matriz de variáveis latentes de \mathbf{Y} ; \mathbf{F} é a matriz de resíduo de \mathbf{Y} ; e \mathbf{Q} é a matriz dos pesos de \mathbf{Y} . As equações (2.4) e (2.5) são chamadas relações externas (*outer relations*) (Geladi & Kowalski, 1986). Para realizar a regressão entre \mathbf{X} e \mathbf{Y} é necessário, ainda, determinar a relação interna (*inner relation*) entre \mathbf{T} e \mathbf{Q} , representada pela matriz \mathbf{B} . Segundo Kaspar & Ray (1992), as matrizes \mathbf{T} e \mathbf{Q} são não singulares e a matriz \mathbf{B} é do tipo diagonal. Desta forma, o valor de \mathbf{Y} é calculado pela equação (2.6):

$$\hat{\mathbf{Y}} = \mathbf{T}\mathbf{B}\mathbf{Q}^T + \mathbf{F}. \quad (2.6)$$

O modelo *PLS* da equação (2.6) pode ser reescrito como um modelo de regressão linear em termos das variáveis originais \mathbf{X} (Kresta et al., 1994):

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\alpha}, \quad (2.7)$$

onde:

$$\hat{\alpha} = \sum_{i=1}^{\mathcal{A}} \left(\prod_{j=1}^{i-1} (\mathbf{I} - w_j p_j^T) \right) w_i q_i^T, \quad (2.8)$$

onde \mathcal{A} é o número de variáveis latentes; \mathbf{I} é a matriz identidade; $w_j \in \mathbf{W}$; \mathbf{W} matriz de pesos; e $p_j \in \mathbf{P}$. As matrizes \mathbf{P} , \mathbf{U} , \mathbf{B} , \mathbf{T} , \mathbf{W} e \mathbf{Q} podem ser calculadas através do método *NIPALS* (*Nonlinear Iterative Partial Least Squares*) (Geladi & Kowalski, 1986; Kresta et al., 1994).

2.6 Filtro de Kalman

O filtro de Kalman (Kalman, 1960) é um estimador para o chamado “problema linear-quadrático-gaussiano”, que é um problema da estimação dos estados instantâneos de um sistema linear dinâmico, perturbado por ruído gaussiano branco, usando-se medições linearmente relacionadas aos estados e também corrompidas por ruído branco (Grewal & Andrews, 1993). De modo simplificado, o filtro de Kalman é um conjunto de equações matemáticas que provê uma solução computacional eficiente para o

método dos mínimos quadrados (Welch & Bishop, 1998). Este algoritmo permite a estimação dos estados passados, presentes e futuros de um sistema, não sendo necessário para isto um conhecimento preciso sobre o sistema (Welch & Bishop, 1998). A próxima seção apresenta o algoritmo mais simples deste método de estimação, o filtro de Kalman discreto.

2.6.1 Filtro de Kalman Discreto

O filtro de Kalman é um algoritmo linear, recursivo, não-polarizado e eficiente para a estimação dos estados desconhecidos de sistemas dinâmicos a partir de medições discretas corrompidas por ruído (Welch & Bishop, 1998). Estados, neste caso, referem-se a qualquer quantidade de interesse envolvida no processo dinâmico. Esta seção apresenta, brevemente, o algoritmo do filtro de Kalman discreto (*DKF - Discrete Kalman Filter*) (Figura 2.2³).

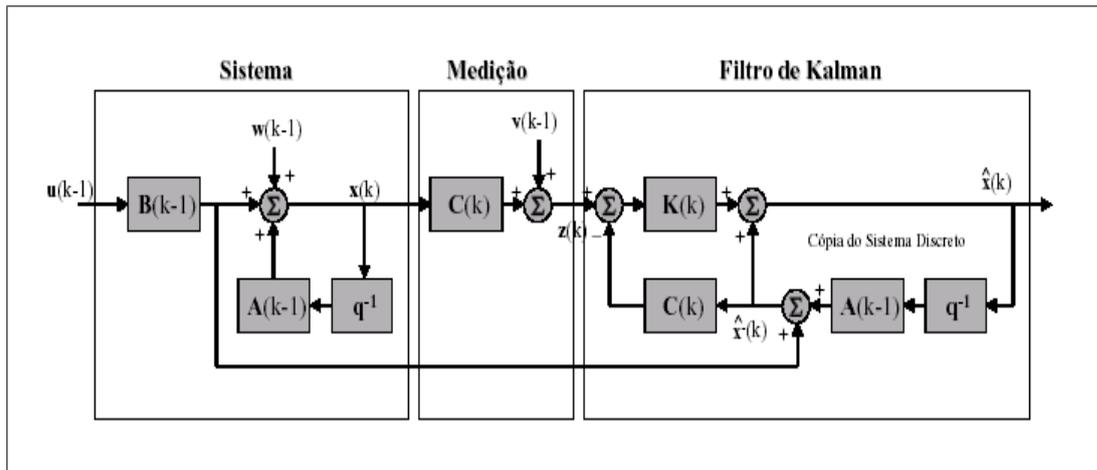


Figura 2.2: Filtro de Kalman discreto

O filtro pressupõe que o sistema em questão pode ser descrito por uma equação linear a diferenças como na equação (2.9):

$$\mathbf{x}(k) = \mathbf{A}(k-1)\mathbf{x}(k-1) + \mathbf{B}(k-1)\mathbf{u}(k-1) + \mathbf{w}(k-1), \quad (2.9)$$

onde $\mathbf{x} \in \mathfrak{R}^n$ é o vetor dos estados, $\mathbf{u} \in \mathfrak{R}^l$ é o vetor das entradas de controle, $\mathbf{A} \in \mathfrak{R}^{n \times n}$ é a matriz de transição de estados, $\mathbf{B} \in \mathfrak{R}^{n \times l}$ é a matriz de transição de entrada e $\mathbf{w} \in \mathfrak{R}^n$ representa o ruído do processo. Os índices k e $k-1$ representam

³Figura obtida de Welch & Bishop (1998)

os instantes de tempo atual e anterior, respectivamente. Assume-se, também, que as medições são linearmente relacionadas aos estados pela equação (2.10):

$$\mathbf{z}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{v}(k), \quad (2.10)$$

onde $\mathbf{y} \in \mathfrak{R}^m$ é o vetor de medições, $\mathbf{C} \in \mathfrak{R}^{m \times n}$ é a matriz de observação e $\mathbf{v} \in \mathfrak{R}^m$ é o vetor de erros de medição (Chen, 1984). Os vetores \mathbf{w} e \mathbf{v} são variáveis aleatórias gaussianas não correlacionadas de média zero (equações (2.11) e (2.12)):

$$E[\mathbf{w}(k)] = E[\mathbf{v}(k)] = \mathbf{0} \quad \forall k, \quad (2.11)$$

com covariância:

$$E[\mathbf{w}(k)\mathbf{w}^\top(k)] = \delta_{ki}\mathbf{Q}(k), E[\mathbf{v}(k)\mathbf{v}^\top(k)] = \delta_{ki}\mathbf{R}(k), \quad (2.12)$$

onde \mathbf{Q} e \mathbf{R} são as matrizes de covariância dos ruídos de processo (\mathbf{w}) e medição (\mathbf{v}), respectivamente.

Desta forma, o objetivo é encontrar uma estimação para o vetor $\mathbf{x}(k)$, representada por $\hat{\mathbf{x}}(k)$, que minimiza o erro quadrático médio. Seja $\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$ o vetor dos resíduos entre os estados reais e os estimados, a função custo a ser minimizada é o somatório do quadrado dos resíduos para cada instante de amostragem (Welch & Bishop, 1998) (equação (2.13)):

$$J = \sum_{i=1}^n \mathbf{e}(i) = \mathbf{e}^\top \mathbf{e} = \|\mathbf{e}\|^2, \quad (2.13)$$

onde a minimização da função custo J causa a minimização da covariância do resíduo ($\mathbf{P}(k)$), como mostra a equação (2.14):

$$\mathbf{P}(k) = E[\mathbf{e}^\top \mathbf{e}]. \quad (2.14)$$

Um vetor de estados estimados *a priori* no instante k pode ser definido como $\hat{\mathbf{x}}^- \in \mathfrak{R}^n$. Desta forma, pela equação (2.9) tem-se:

$$\hat{\mathbf{x}}^-(k) = \mathbf{A}(k-1)\hat{\mathbf{x}}(k-1) + \mathbf{B}(k-1)\mathbf{u}(k-1), \quad (2.15)$$

$$\mathbf{e}^-(k) = \mathbf{x}(k) - \hat{\mathbf{x}}^-(k), \quad e \quad (2.16)$$

$$\mathbf{P}^-(k) = E[\mathbf{e}^{-T} \mathbf{e}^-]. \quad (2.17)$$

O objetivo do filtro de Kalman é encontrar uma equação que calcule os estados $\hat{\mathbf{x}}(k)$ a partir de uma combinação linear dos estados estimados *a priori* $\hat{\mathbf{x}}^-(k)$ e do vetor de medições $\mathbf{y}(k)$ (equação (2.18)):

$$\hat{\mathbf{x}}(k) = \mathbf{K}^1(k) \hat{\mathbf{x}}^-(k) + \mathbf{K}(k) \mathbf{y}(k). \quad (2.18)$$

A partir da condição de ortogonalidade entre os resíduos e os estados imposta pelo método dos mínimos quadrados observa-se que:

$$\mathbf{K}^1(k) = \mathbf{I} - \mathbf{K}(k) \mathbf{C}(k), \quad (2.19)$$

e portanto,

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k) [\mathbf{y}(k) - \mathbf{C}(k) \hat{\mathbf{x}}^-(k)], \quad (2.20)$$

onde a matriz $\mathbf{K}(k) \in \Re^{n \times m}$ deve ser escolhida para ser a matriz de ganho que minimiza a covariância do resíduo (equação (2.14)), ou seja:

$$\mathbf{K}(k) = \mathbf{P}^-(k) \mathbf{C}^T(k) [\mathbf{C}(k) \mathbf{P}^-(k) \mathbf{C}^T(k) + \mathbf{R}(k)]^{-1}, \quad (2.21)$$

$$\mathbf{P}(k) = [\mathbf{I} - \mathbf{K}(k) \mathbf{C}(k)] \mathbf{P}^-(k). \quad (2.22)$$

As equações (2.20), (2.21) e (2.22) compõem o algoritmo do filtro de Kalman discreto (Balakrishnan, 1987; Grewal & Andrews, 1993). Na prática, a obtenção de $\mathbf{Q}(k)$ e $\mathbf{R}(k)$ é realizada através de experimentação e intuição (Welch & Bishop, 1998). No caso de $\mathbf{Q}(k)$, uma vez que o filtro de Kalman já foi implementado, $\mathbf{Q}(k)$ pode ser ajustada por tentativa e erro através da comparação entre os resultados obtidos. O valor de $\mathbf{Q}(k)$ final é aquele que forneceu os melhores resultados durante a fase de experimentação (Welch & Bishop, 1998). Duas possíveis simplificações em relação à estrutura de $\mathbf{Q}(k)$ podem ser feitas a fim de facilitar a sua obtenção. A

primeira é assumir que esta é constante e a outra é atualizar $\mathbf{Q}(k)$ de acordo com a variância do erro de estimação até o instante k (Doebelin, 1990).

Para a determinação de $\mathbf{R}(k)$ não existem métodos analíticos. Novamente, é necessária uma determinação empírica, levando-se em consideração que \mathbf{R} é uma matriz diagonal. Neste caso, esta consideração reflete muito bem a realidade, principalmente quando supõe-se que as medições vêm de sensores distintos. A matriz pode ser, então, considerada constante, permitindo que os valores das covariâncias possam ser derivados, ao menos aproximadamente, através dos dados fornecidos pelos fabricantes dos sensores: sensores mais precisos possuem covariância menor (Doebelin, 1990; Grewal & Andrews, 1993).

2.7 Inteligência Computacional

Como mencionado anteriormente, nos casos onde os processos apresentam alta complexidade e comportamento não-linear, a modelagem torna-se difícil quando são empregadas abordagens teóricas baseadas em técnicas convencionais de estimação. Nestes casos, abordagens alternativas, baseadas em inteligência computacional, podem ser utilizadas na tentativa de superar tais dificuldades. Dentre essas técnicas estão as redes neurais (Hopfield, 1982; Rumelhart et al., 1986), mapas auto-organizáveis (Kohonen, 1984), agrupamento *fuzzy* (Bezdek, 1987) e sistemas *fuzzy* (Zadeh, 1965). As próximas seções apresentam uma breve introdução sobre essas técnicas.

2.7.1 Redes Neurais Multicamadas

As redes neurais têm sido amplamente aplicadas na modelagem e controle de processos (Willis et al., 1992; Wang & Rong, 1999; Rallo et al., 2002; Fabro et al., 2005). Em geral, as redes neurais são utilizadas em conjunto com alguma ferramenta estatística, tais como *PCA* ou *PLS* para compor o modelo inferencial, com a intenção de diminuir a variância da predição e também de reduzir a dimensão do espaço de entrada (Qin & McAvoy, 1992; Qin, 1996; Bhartiya & Whiteley, 2001).

Esta seção apresenta uma breve introdução sobre redes neurais, enfatizando a rede neural do tipo Perceptron multicamadas (*MLP - Multilayer Perceptron*) (Rumelhart et al., 1986), pois trata-se da rede neural mais usualmente empregada para este propósito. A próxima seção contém os conceitos básicos sobre os mapas auto-organizáveis de Kohonen, cujo principal interesse é o seu emprego no procedi-

mento de seleção das variáveis de entrada do modelo da metodologia proposta neste trabalho.

Uma rede neural é um sistema composto de unidades elementares de processamento, chamadas de neurônios ou nós, organizadas em camadas e interconectadas entre si através de um conjunto de pesos sinápticos. A Figura 2.3⁴ ilustra um modelo de neurônio proposto por Rosenblatt (1962). Este neurônio consiste de:

- um conjunto de sinapses, cada uma delas representada por um peso característico. Especificamente, um sinal x_j na entrada da sinapse j conectada ao neurônio k é multiplicado pelo peso sináptico w_{jk} . Este valor pode assumir valores positivos ou negativos;
- um combinador linear para somar os sinais de entrada, ponderados pela respectiva sinapse do neurônio; e
- uma função de ativação, em geral uma função sigmóide (equação (2.25)) (Jang et al., 1997), para limitar a amplitude da saída do neurônio a algum valor finito. Comumente, a amplitude normalizada da saída de um neurônio é restrita ao intervalo fechado $[0, 1]$ ou, alternativamente, $[-1, 1]$.

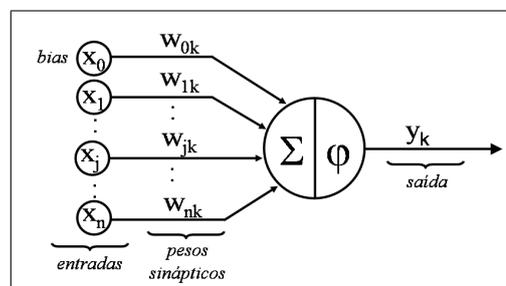


Figura 2.3: Modelo de neurônio proposto por Rosenblatt (1962).

O modelo da Figura 2.3 inclui uma polarização externa (*bias*), denotada por b_k (ou x_0). A polarização b_k tem o efeito de aumentar ou diminuir o argumento da função de ativação, caso seja positivo ou negativo, respectivamente. Desta forma, em termos matemáticos, um neurônio k pode ser descrito pelas equações (2.23) e (2.24):

$$un_k = \sum_{j=1}^n (w_{jk}x_j) + b_k, \quad (2.23)$$

$$y_k = \varphi(un_k), \quad (2.24)$$

⁴Figura obtida de Jang et al. (1997).

$$\varphi(un_k) = \frac{1}{1 + \exp(-\beta un_k)}, \quad (2.25)$$

onde: x_1, \dots, x_n são os sinais de entrada; $w_{11}, \dots, w_{jk}, \dots, w_{nk}$ são os pesos sinápticos do neurônio k , com $k = 1, \dots, m$ e un_k $j = 1, \dots, n$; un_k é a saída do combinador linear devida aos sinais de entrada; b_k é a polarização ou *bias*; $\varphi(\cdot)$ é a função de ativação do tipo sigmoidal; β é o parâmetro de inclinação da função sigmoidal; e y_k é o sinal de saída do neurônio k .

As redes multicamadas têm por característica possuir uma ou mais camadas escondidas, cujos neurônios são chamados de neurônios escondidos ou unidades escondidas. A Figura 2.4⁵ apresenta uma rede neural *MLP* para o caso com uma camada escondida.

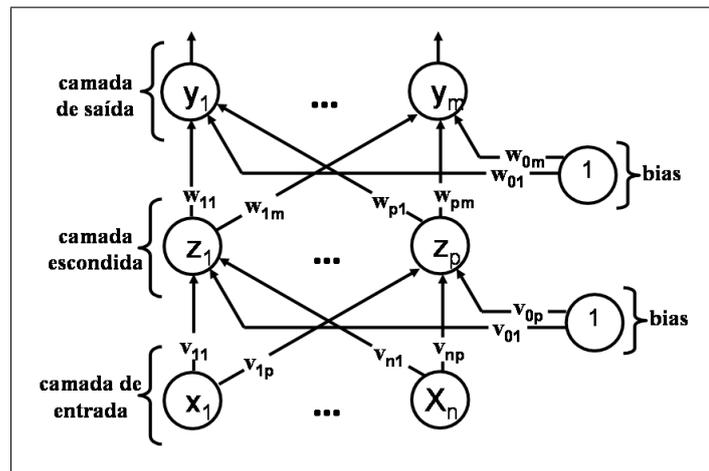


Figura 2.4: Exemplo de uma rede neural *MLP*.

Na Figura 2.4, os neurônios da camada de entrada da rede provêm os vetores de entrada, que correspondem aos sinais de entrada aplicados aos neurônios da primeira camada escondida (segunda camada da rede). Os sinais dessa segunda camada são usados como entradas para a terceira camada, e assim sucessivamente até a última camada (camada de saída). O conjunto de sinais de saída da última camada da rede constituem a resposta global da rede ao padrão apresentado na camada de entrada.

O número de neurônios na camada de entrada é determinado pela dimensionalidade do espaço de entrada. O número de neurônios da camada de saída é determinado pela dimensionalidade requerida da resposta desejada. Assim, a construção de uma rede neural *MLP* requer a consideração de três aspectos:

⁵Figura obtida de Jang et al. (1997).

1. determinação do número de camadas escondidas;
2. determinação do número de neurônios em cada uma das camadas escondidas;
3. especificação dos pesos sinápticos que interconectam os neurônios nas diferentes camadas da rede.

Os aspectos 1 e 2 determinam a complexidade do modelo de rede neural escolhido, e não há regras determinadas para tal especificação. No caso de somente uma camada escondida, o teorema de Kolmogorov-Nielsen (Kolmogorov, 1957; Hecht-Nielsen, 1987; Kurkova, 1992) pode ser empregado para estipular o número p de neurônios desta camada:

Teorema 2.1 *Teorema de Kolmogorov-Nilsen: Seja $f : [0, 1]^n \rightarrow \mathfrak{R}^m$ uma função contínua. Então f pode ser representada por uma rede neural multicamadas, com uma camada escondida, com n unidades de entrada, $p = 2^n + 1$ unidades na camada escondida e m unidades na camada de saída.*

O aspecto 3 envolve a utilização de algoritmos de treinamento para o ajuste dos pesos sinápticos.

As redes neurais possuem a capacidade de aprender por exemplos e fazer interpolações e extrapolações do que aprenderam. Isto é feito através de um processo iterativo de ajustes aplicado aos seus pesos, o treinamento. O aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas. Denomina-se algoritmo de aprendizado um conjunto de regras bem definidas para a solução de um problema de aprendizado. Existem muitos tipos de algoritmos de aprendizado específicos para determinados modelos de redes neurais. Estes algoritmos diferem entre si principalmente pelo modo como os pesos são modificados. Outro fator importante para a escolha do algoritmo de aprendizado é a maneira pela qual uma rede neural se relaciona com o ambiente. Neste contexto existem os seguintes paradigmas de aprendizado:

- Aprendizado Supervisionado, quando é utilizado um agente externo que indica à rede a resposta desejada para o padrão de entrada;
- Aprendizado Não-Supervisionado (auto-organização), quando não existe um agente externo indicando a resposta desejada para os padrões de entrada;

- Aprendizado por Reforço, quando um crítico externo avalia a resposta fornecida pela rede.

Denomina-se ciclo uma apresentação de todos os pares de entrada e saída do conjunto de treinamento no processo de aprendizagem. A correção dos pesos sinápticos num ciclo pode ser executado em dois modos:

- Modo Padrão: A correção dos pesos acontece a cada apresentação à rede de uma amostra do conjunto de treinamento. Cada correção de pesos baseia-se somente no erro do exemplo apresentado naquela iteração. Assim, a quantidade de correções depende do número de amostras contidas no conjunto de treinamento.
- Modo *Batch*: Apenas uma correção é feita por ciclo. Todos os exemplos do conjunto de treinamento são apresentados à rede, seu erro médio é calculado e a partir desse erro fazem-se as correções dos pesos.

Um dos algoritmos mais conhecidos e utilizados é o algoritmo de aprendizado por retropropagação de erros (*Backpropagation (of errors) Algorithm*) (Rumelhart et al., 1986), também conhecido como Regra Delta Generalizada, e suas variantes (Jang et al., 1997). Trata-se de um algoritmo de aprendizado supervisionado, cuja correção é feita em modo padrão.

Basicamente, o algoritmo *backpropagation* consiste de dois passos através das camadas da rede neural: um passo direto e um passo reverso. No passo direto, um padrão de atividade do processo a ser aprendido (ou vetor de entrada) é aplicado aos nós de entrada e o seu efeito se propaga através da rede, camada por camada, produzindo na camada de saída a resposta da rede à excitação aplicada (gerando o vetor de saída). Durante o passo direto, os pesos sinápticos são todos fixos. Durante o passo reverso, os pesos sinápticos são ajustados de acordo com a regra de aprendizagem por correção do erro. Este sinal de erro é propagado de volta através dos mesmos neurônios utilizados no passo direto, mas no caminho contrário do fluxo de sinal nas conexões sinápticas. Desta forma, os pesos sinápticos são ajustados de modo que a resposta final obtida pela rede seja a mais próxima possível da resposta desejada (Fausett, 1994).

2.7.2 Mapa Auto-Organizável de Kohonen

Os Mapas Auto-Organizáveis de Kohonen (*Self-Organizing Maps - SOMs*) são redes neurais competitivas que possuem a habilidade de formar mapeamentos que preservam a topologia entre os espaços de entrada e saída. As redes *SOM* resolvem problemas não-lineares de alta dimensionalidade, tais como: extração de características e classificação de imagens e padrões acústicos, controle adaptativo de robôs, equalização, demodulação e transmissão de sinais (Haykin, 1999; Zuchini, 2003).

A Figura 2.5⁶ apresenta a arquitetura reticulada unidimensional formada por m neurônios (unidades). Cada unidade está conectada a todas as entradas da rede neural. O peso sináptico de uma unidade j correspondente à entrada x_j na unidade k é w_{kj} . A organização espacial das unidades torna possível o estabelecimento do conceito de distância e de vizinhança dentre unidades. A vizinhança π_k da unidade de saída k é formada pelo conjunto de unidades que estão a uma distância menor que um valor máximo estipulado.

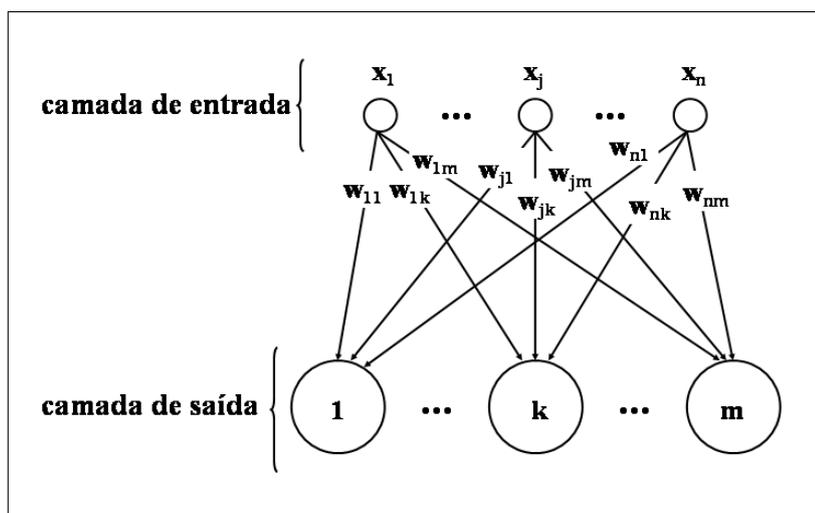


Figura 2.5: Mapa auto-organizável unidimensional.

As redes *SOM* são baseadas na aprendizagem competitiva não-supervisionada, onde os neurônios de saída competem entre si para serem ativados. Após receberem um sinal de entrada, os neurônios competem entre si disputando o controle sobre os procedimentos relacionados à aprendizagem. O vencedor da competição, ou seja, a melhor unidade do mapa (*best map unit - bmu*), é determinado de acordo com as distâncias dos vetores de pesos sinápticos ao vetor de entrada.

Os pesos sinápticos são ajustados a cada sinal de entrada apresentado ao mapa.

⁶Figura obtida de (Fausett, 1994).

Seja $\mathbf{x}(t)$ o t -ésimo vetor de entrada ($\mathbf{x} \in \mathfrak{R}^n$) apresentado ao mapa, e que para esta entrada o vetor de pesos sinápticos da unidade k seja $\mathbf{w}_k(t)$. Assim que as unidades recebem a entrada $\mathbf{x}(t)$, cada uma delas computa a distância $\mathcal{D}_k(t)$ entre esta entrada e seu vetor de pesos sinápticos $\mathbf{w}_k(t)$, tal como na equação (2.26):

$$\mathcal{D}_k(t) = \|\mathbf{x}(t) - \mathbf{w}_k(t)\|, \quad (2.26)$$

onde $\|\cdot\|$ é a norma Euclidiana, e:

$$\mathbf{w}_k(t) = (w_{1k}(t) \dots w_{jk}(t) \dots w_{nk}(t)); \quad (2.27)$$

$$\mathbf{x}(t) = (x_1(t) \dots x_j(t) \dots x_n(t)). \quad (2.28)$$

A *bmu* correspondente à entrada $\mathbf{x}(t)$ possui o vetor de pesos sinápticos mais próximo da entrada e pode ser determinado de acordo com a equação (2.29):

$$bmu(t) = \underset{k}{\operatorname{arg\,min}} (\mathcal{D}_k(t)). \quad (2.29)$$

A atualização dos pesos sinápticos é realizada de acordo com a equação (2.30). Conforme esta equação, o ajuste sináptico é fortemente dependente da unidade vencedora, pois somente as unidades pertencentes à vizinhança da *bmu* são ajustadas, tornando os respectivos vetores de pesos sinápticos mais próximos do vetor de entrada.

$$w_{jk}(t+1) = w_{jk}(t) + \eta(t)\pi_{bmu}(t)(x_j(t) - w_{jk}(t)), \quad (2.30)$$

onde $\eta(t)$ é a taxa de aprendizado $0 < \eta < 1$; e $\pi_{bmu}(t)$ é a vizinhança da *bmu* e ambos variam ao longo da aprendizagem conforme as equações (2.31) e (2.32), respectivamente (Kohonen, 1984).

$$\eta(t+1) = \vartheta\eta(t), \quad (2.31)$$

$$\pi_{bmu}(t) = \begin{cases} 0 & , \text{ se } \|bmu(t) - j\| > \delta(t) \\ 1 & , \text{ caso contrário} \end{cases}, \quad (2.32)$$

$$\delta(t+1) = \rho\delta(t), \quad (2.33)$$

em que $\delta(t) \geq 0$ (largura da vizinhança) define os elementos de $\pi_{bmu}(t)$ correspondentes à t -ésima entrada; $0 < \rho, \vartheta < 1$; ρ e ϑ são constantes; e os valores iniciais $\eta(0)$ e $\delta(0)$ devem ser definidos *a priori*.

A atualização do taxa de aprendizado ($\eta(t)$) e da vizinhança da *bmu* ($\pi_{bmu}(t)$) durante o processo de aprendizagem permite a formação de mapas topológicos sobre a arquitetura do *SOM* (Figueiredo, 1997), pois no início da aprendizagem a vizinhança $\pi_{bmu}(t)$ pode conter todas as unidades do mapa, e à medida que a aprendizagem progride, a vizinhança pode ser reduzida até que $\pi_{bmu}(t)$ contenha somente as unidades adjacentes à *bmu*, ou seja, $\delta(t) = 1$, ou somente a própria *bmu*, $\delta(t) = 0$.

A dinâmica do taxa de aprendizado $\eta(t)$ influencia tanto a formação dos mapas topológicos quanto a convergência do processo de aprendizagem. É aconselhável que este fator seja inicializado com um valor máximo e reduzido a cada iteração. Kohonen (1984) indica que uma função monotônica decrescente positiva mostra-se satisfatória, como descreve a equação (2.31).

O algoritmo responsável pela formação do *SOM* em primeiro lugar inicializa os pesos sinápticos da rede. Este procedimento pode ser feito atribuindo pequenos valores tomados de um gerador de números aleatórios; desta forma, nenhuma ordem prévia é imposta ao mapa de características. Desde que o mapa tenha sido adequadamente inicializado, há três processos essenciais envolvidos na formação do *SOM*, conforme descritos abaixo (Haykin, 1999):

- **Competição:** Para cada padrão de entrada, os neurônios da rede computam os seus respectivos valores de uma função discriminante. Esta função provê as bases para a competição entre os neurônios. O neurônio com o maior valor de função discriminante é declarado o vencedor da competição.
- **Cooperação:** O neurônio vencedor determina a localização espacial de uma vizinhança topológica de neurônios excitados, provendo, desta forma, as bases para a cooperação entre tais neurônios vizinhos.
- **Adaptação sináptica:** Este último mecanismo permite aos neurônios excitados aumentar seus valores individuais da função discriminante em relação ao padrão de entrada, através de ajustes adequados aplicados aos seus pesos sinápticos. Os ajustes feitos são tais que a resposta do neurônio vencedor à

subsequente aplicação de um padrão similar de entrada é realçada.

Cada padrão de entrada apresentado à rede consiste de uma região localizada de atividade. A localização e natureza de tal região usualmente varia de uma realização de padrão de entrada para outra. Todas as unidades do mapa devem, portanto, ser expostas a um número suficiente de diferentes realizações dos padrões de entrada, para garantir que o processo de auto-organização ocorra de forma apropriada (Haykin, 1999).

A Figura 2.6 descreve os passos do algoritmo de aprendizagem, obtida de Figueiredo (1997).

Algoritmo 1: Aprendizagem do Mapa Auto-Organizável de Kohonen.

- **Passo 1:** *Inicialização* - Escolher os valores para os pesos sinápticos w_{ij} de forma aleatória, preferivelmente considerando uma distribuição uniforme.
- **Passo 2:** *Estímulo* - Apresentar ao mapa um vetor $\mathbf{x}(t)$ pertencente ao conjunto de vetores de entrada, escolhido aleatoriamente;
- **Passo 3:** *Competição* - Computar as distâncias $\mathcal{D}_j(t)$, $j = 1, \dots, n$, com relação à entrada $\mathbf{x}(t)$ e determinar a bmu para o qual a distância $\mathcal{D}_j(t)$ é mínima, como indica a equação (2.29);
- **Passo 4:** *Ajuste e Atualização* - Ajustar os pesos sinápticos de acordo com a equação (2.30), e atualizar o taxa de aprendizado (equação (2.31)) e a função de vizinhança $\pi_{bmu}(t)$ (equação (2.32));
- **Passo 5:** *Avaliação de convergência* - Retornar ao passo 2 caso os mapas topológicos não estejam bem formados. Caso contrário, finalizar a aprendizagem.

Figura 2.6: Algoritmo de aprendizagem do SOM.

Quando os mapas topológicos são bem formados, duas características podem ser observadas (Figueiredo, 1997):

- Todo o espaço de entrada é representado na arquitetura do mapa. Cada unidade, através de seus pesos sinápticos, representa um subconjunto específico deste espaço. Além disso, a relação topológica existente entre os vetores sinápticos se mantém sobre as respectivas unidades a que estão associados, ou seja, unidades próximas estão associadas a vetores sinápticos próximos.

- O mapeamento topológico reflete não só as relações topológicas mas também as características estatísticas presentes na distribuição dos vetores de entrada. Cada região do espaço de entrada é representada de maneira proporcional à probabilidade com que seus vetores são apresentados ao mapa durante o processo de aprendizagem. Um número maior de unidades é associado às regiões cujos vetores de entrada são escolhidos com maior probabilidade.

2.7.3 Sistemas *Fuzzy*

Sistemas *fuzzy* são sistemas fundamentados na teoria de conjuntos *fuzzy* (Zadeh, 1965). Essa teoria possibilita a representação do conhecimento lingüístico em termos matemáticos computacionalmente convenientes, e o seu processamento é realizado através de técnicas de raciocínio aproximado oferecidas por mecanismos de inferência específicos (Nagai, 2002).

A teoria dos conjuntos *fuzzy* é uma generalização da teoria clássica dos conjuntos. Esta teoria abandona a rigidez da teoria clássica, introduzindo os conceitos de conjunto *fuzzy* e grau de pertinência (Zadeh, 1965). Diferentemente da teoria clássica dos conjuntos, onde um elemento simplesmente pertence ou não a um conjunto, na teoria dos conjuntos *fuzzy* um elemento pode pertencer a um ou mais conjuntos *fuzzy* com um determinado grau de pertinência que varia no intervalo $[0, 1]$, onde o valor 0 indica uma completa não-pertinência ao conjunto e o valor 1 representa a completa pertinência.

A função que define os graus de pertinência dos elementos é chamada de função de pertinência. Trata-se de uma generalização da função característica da teoria clássica, uma vez que associa para todo elemento do universo de discurso um valor pertencente ao intervalo $[0, 1]$, ao invés do conjunto de apenas dois elementos $\{0, 1\}$.

Os sistemas *fuzzy* têm sido empregados nas mais diversas aplicações práticas. Em controle de processos industriais, as primeiras experiências datam de 1974, quando Mamdani (1974), após inúmeras tentativas frustradas em controlar uma máquina a vapor com tipos diferentes de controladores, incluindo o *PID*, demonstrou de forma pioneira que um controlador *fuzzy* controlava eficientemente a máquina a vapor.

Um sistema *fuzzy* é composto, basicamente, por quatro módulos: (i) fuzificação (interface de entrada); (ii) base de conhecimento (base de dados e base de regras); (iii) módulo de inferência; e (iv) defuzificação (interface de saída), como ilustrado

pela Figura 2.7 ⁷.

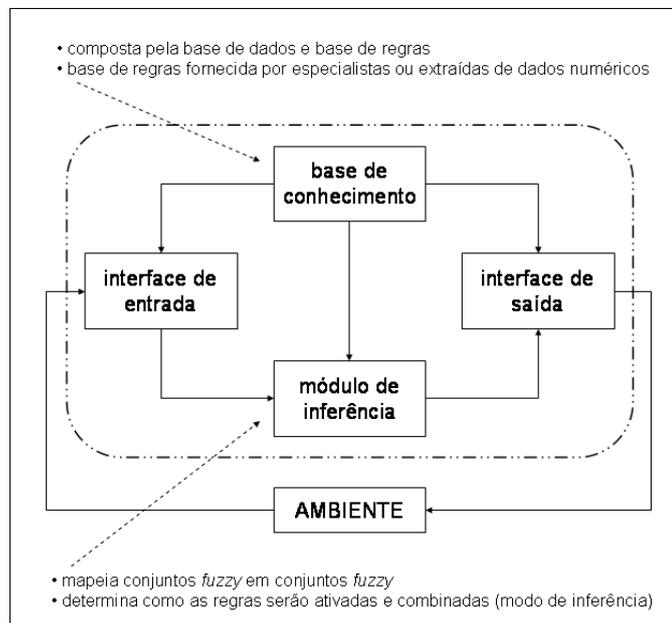


Figura 2.7: Diagrama de um sistema fuzzy básico.

A fuzificação está relacionada à necessidade de mapear entradas numéricas em representações lingüísticas através de conjuntos *fuzzy*. Esta interface é responsável por manipular os dados de entrada de modo que estes se apresentem de forma conveniente para o módulo de inferência. É neste módulo que também é realizada a discretização e normalização do universo de discurso de uma variável lingüística.

Segundo Zadeh (1975), variáveis lingüísticas são variáveis cujos valores não são numéricos, mas palavras ou sentenças em linguagem natural ou artificial. Esses valores que a variável lingüística pode assumir são chamados de termos lingüísticos ou conceitos. O universo de discurso de uma variável lingüística é o conjunto de todos os valores possíveis para os conceitos dessa variável lingüística.

A base de conhecimento de um sistema *fuzzy* é composta pela base de dados e pela base de regras. A base de dados armazena todas as informações referentes às variáveis lingüísticas do sistema, p.ex., termos lingüísticos associados e o universo de discurso. Essas informações, além de descreverem as variáveis lingüísticas, são utilizadas para caracterizar as regras *fuzzy* e a manipulação de dados em um sistema *fuzzy*. A base de regras contém as regras *fuzzy* que descrevem o comportamento do sistema na forma de declarações do tipo *se-então*, onde tais regras podem ser obtidas através de conhecimento especialista e/ou através de dados amostrados do sistema.

⁷Figura obtida de Figueiredo (1997).

O módulo de inferência é o responsável por inferir a resposta do sistema a partir das entradas apresentadas. É responsável por avaliar, encadear e determinar quando e como as regras e os dados serão utilizados. O modo como as regras serão combinadas e ativadas depende dos operadores de agregação de antecedentes, de projeção e de agregação de regras, assim como do tipo de modelo *fuzzy* utilizado.

Finalmente, a função do módulo de defuzificação é o de transformar o resultado da inferência de maneira que a saída seja compatível com o ambiente externo no qual o sistema *fuzzy* está inserido. Para aplicações de controle, a saída é transformada em um valor numérico.

Modelo *Fuzzy*

Sistemas *fuzzy* podem ser utilizados para vários propósitos, tais como modelagem, análise de dados, predição ou controle. No caso de modelagem de processos, as regras *fuzzy* que compõem a base de regras de um sistema *fuzzy* descrevem o comportamento do processo e correspondem ao modelo *fuzzy* do processo.

Em modelos *fuzzy*, os relacionamentos entre as variáveis são representados por regras *fuzzy se-então* como ilustra a Figura 2.8. A proposição antecedente é sempre uma proposição *fuzzy* do tipo “ x é A ”, onde x é uma variável lingüística (p.ex. velocidade) e A é um termo lingüístico (p. ex. baixa).

Se (antecedente) Então (conseqüente)

Figura 2.8: Regra se-então.

Dependendo da forma do conseqüente, três tipos de modelos *fuzzy* podem ser destacados:

- Modelo *fuzzy* lingüístico, onde tanto o antecedente quanto o conseqüente da regra são proposições *fuzzy*;
- Modelo *fuzzy Takagi-Sugeno*, onde o antecedente é uma proposição *fuzzy* e o conseqüente é uma função numérica.
- Modelo *fuzzy* relacional, é uma generalização do modelo lingüístico, onde o mapeamento entre os conjuntos *fuzzy* das variáveis de entrada e saída são representados por uma relação *fuzzy* (Babůska & Vebruggen, 1996).

Modelo Fuzzy Lingüístico O modelo *fuzzy* lingüístico (Mamdani, 1974) foi introduzido como um modo de capturar o conhecimento qualitativo disponível na forma de regras *se-então*. A Figura 2.9 ilustra uma base de regras de um modelo *fuzzy* lingüístico composto por M regras, para um sistema com n entradas e uma saída (sistema *MISO*, *multi-input single-output*).

$$\begin{array}{l}
 R_1 : \text{Se } x_1 \text{ é } A_{11} \text{ e } \dots \text{ e } x_n \text{ é } A_{1n} \text{ Então } y \text{ é } B_1 \\
 \vdots \\
 R_i : \text{Se } x_1 \text{ é } A_{i1} \text{ e } \dots \text{ e } x_n \text{ é } A_{in} \text{ Então } y \text{ é } B_i \\
 \vdots \\
 R_M : \text{Se } x_1 \text{ é } A_{M1} \text{ e } \dots \text{ e } x_n \text{ é } A_{Mn} \text{ Então } y \text{ é } B_M
 \end{array}$$

Figura 2.9: Modelo fuzzy lingüístico.

Na Figura 2.9, x_j é uma variável lingüística de entrada (antecedente) e A_{ij} , $i = 1, \dots, M$, $j = 1, \dots, n$, $n =$ número de variáveis de entrada; y é a variável de saída (conseqüente) e B_i são os termos lingüísticos de y . O conector “e” é implementado por um operador de conjunção (interseção), definido por uma função de duas variáveis, denominada t-norma (Definição 2.7.1) (Zadeh, 1965).

Definição 2.7.1 A conjunção de dois conjuntos fuzzy A e B , pertencentes ao universo de discurso U , com suas respectivas funções de pertinência $\mu_A(u)$ $\mu_B(u)$, $u \in U$, é um conjunto fuzzy C , escrito como $C = A \cap B$, cuja função de pertinência é definida como:

$$\mu_C(u) = t(\mu_A(u), \mu_B(u)), \forall u \in U. \quad (2.34)$$

Similarmente, denomina-se uma s-norma um operador de disjunção (união) definido por uma função de duas variáveis, como na Definição 2.7.2 (Zadeh, 1965).

Definição 2.7.2 A disjunção de dois conjuntos fuzzy A e B , pertencentes ao universo de discurso U , com suas respectivas funções de pertinência $\mu_A(u)$ $\mu_B(u)$, $u \in U$, é um conjunto fuzzy C , escrito como $C = A \cup B$, cuja função de pertinência é definida como:

$$\mu_C(u) = s(\mu_A(u), \mu_B(u)), \forall u \in U. \quad (2.35)$$

Dentre os operadores de conjunção estão os operadores mínimo e produto, e dentre os operadores de disjunção estão os operadores máximo e soma algébrica (Jang et al., 1997).

$$\text{Mínimo} : t(\mu_A(u), \mu_B(u)) = \min(\mu_A(u), \mu_B(u)). \quad (2.36)$$

$$\text{Produto} : t(\mu_A(u), \mu_B(u)) = (\mu_A(u) \cdot \mu_B(u)). \quad (2.37)$$

$$\text{Máximo} : s(\mu_A(u), \mu_B(u)) = \max(\mu_A(u), \mu_B(u)). \quad (2.38)$$

$$\text{Soma Algébrica} : s(\mu_A(u), \mu_B(u)) = (\mu_A(u) + \mu_B(u)) - (\mu_A(u) \cdot \mu_B(u)). \quad (2.39)$$

Existem diversos mecanismos de inferência para a obtenção da saída do modelo *fuzzy* (Jang et al., 1997). Neste trabalho empregou-se a Inferência *Max-Min*, também chamada de Inferência Mamdani (Mamdani, 1974). Este mecanismo de inferência é realizada em três passos:

- (i) Calcular o grau de ativação (outros operadores de conjunção podem ser usados; neste exemplo é utilizado o operador mínimo):

$$\beta_i^M = \min(\mu_{A_{i1}}, \dots, \mu_{A_{ip}}), \quad i = 1, 2, \dots, M. \quad (2.40)$$

- (ii) Calcular os conjuntos *fuzzy* B'_i :

$$\mu_{B'_i}(y) = \min(\beta_i^M, \mu_{B_i}) \quad i = 1, 2, \dots, M. \quad (2.41)$$

- (iii) Agregar os conjuntos *fuzzy* B'_i em um único conjunto *fuzzy* B' :

$$\mu_{B'}(y) = \max_{1 \leq i \leq M} \mu_{B'_i}(y). \quad (2.42)$$

A saída numérica da variável y é obtida através da defuzificação do conjunto *fuzzy* B' . Dentre os métodos existentes, estão o Centro de Área (*COA - Center Of Area*) e o Bissetor de Área (*BOA - Bisector Of Area*).

O método centro de área (*COA*) gera o centro de gravidade da possível distribuição do conjunto *fuzzy* resultante B' . No caso do universo de discurso discreto, o resultado da defuzificação é gerado a partir da equação (equação (2.43)):

$$\mathbf{b} = COA(B') = \frac{\sum_{d=1}^D \mu_{B'}(y_d) y_d}{\sum_{d=1}^D \mu_{B'}(y_d)}, \quad (2.43)$$

onde D é o número de discretizações do universo de discurso, y_d é o valor representativo do d -ésimo intervalo de discretização do universo de discurso da variável de saída e $\mu_{B'}$ é a função de pertinência do conjunto *fuzzy* resultante B' .

No método bisetor da área (*BOA*) (equação (2.44)), o resultado da defuzificação, \mathbf{b} , divide a área do conjunto *fuzzy* B' em duas partes iguais, tal que:

$$\sum_{d < \mathbf{b}} \mu_{B'}(d) = \sum_{\mathbf{b} < d} \mu_{B'}(d). \quad (2.44)$$

Modelo *Fuzzy Takagi-Sugeno* O modelo *fuzzy Takagi-Sugeno* (*TS*) (Takagi & Sugeno, 1985) utiliza regras nas quais os conseqüentes são funções matemáticas ao invés de conjuntos *fuzzy*. O modelo *TS* é baseado na idéia de descrever o sistema como um conjunto de relações locais entre as entradas e a saída, como ilustra a Figura 2.10.

$$\boxed{R_i: \text{Se } x_1 \text{ é } A_{i1} \text{ e } \dots \text{ e } x_n \text{ é } A_{in} \text{ Então } y = f_i(\cdot)}$$

Figura 2.10: Exemplo de regra fuzzy Takagi-Sugeno.

Geralmente, $f_i(\cdot)$ é uma função linear das variáveis do antecedente, como denota a equação (2.45).

$$f_i(x_1, \dots, x_n) = p_{i0} + p_{i1}x_1 + \dots + p_{in}x_n. \quad (2.45)$$

A fórmula de inferência para o modelo *TS* é dada pela equação (2.46):

$$z_0 = \frac{\sum_{i=1}^M \beta_i f_i(\cdot)}{\sum_{i=1}^M \beta_i}, \quad (2.46)$$

onde β_i é o grau de ativação da premissa da i -ésima regra R_i e $\mu_{A_{ij}}$ é a função de pertinência do i -ésimo conjunto *fuzzy* da variável j , $j = 1, \dots, n$:

$$\beta_i = (\mu_{A_{i1}}(x_1) \text{ t } \dots \text{ t } \mu_{A_{in}}(x_n)), \quad (2.47)$$

onde t é uma t-norma.

Tipo *Singleton* O modelo *fuzzy* do tipo *Singleton* é um caso especial do modelo *fuzzy* Takagi-Sugeno, onde as funções matemáticas do conseqüente são valores numéricos constantes, ou seja, trata-se de um modelo *fuzzy* Takagi-Sugeno de ordem zero, tal como ilustra a Figura 2.11, onde b_i é uma constante.

$$R_i: \text{Se } x_1 \text{ é } A_{i1} \text{ e } \dots \text{ e } x_n \text{ é } A_{in} \text{ Então } y = b_i$$

Figura 2.11: Exemplo de regra *Singleton*.

Para este tipo de modelo, um método de inferência/defuzificação é usualmente empregado, como na equação (2.48):

$$z_0 = \frac{\sum_{i=1}^M \beta_i b_i}{\sum_{i=1}^M \beta_i}. \quad (2.48)$$

Modelo *Fuzzy Relacional* O modelo *fuzzy* relacional (Pedrycz, 1984; Pedrycz, 1993) pode ser visto como uma extensão do modelo *fuzzy* lingüístico, onde o mapeamento entre os conjuntos de entrada e de saída é representado por uma relação *fuzzy*.

Segundo Klir & Folger (1988), uma relação *crisp* (clássica ou não-*fuzzy*) representa a presença ou ausência de associação, interação ou interligação entre elementos de dois ou mais conjuntos. Formalmente (Mendel, 1995):

Definição 2.7.3 *Dados dois universos de discurso A e B , a relação binária \mathbf{R} , sobre $A \times B$, é qualquer subconjunto do produto cartesiano dos dois universos, $A \times B$, de tal forma que:*

$$\mathbf{R} : A \times B \rightarrow \{0, 1\}. \quad (2.49)$$

A relação $\mathbf{R}(A, B)$ pode ser definida pela seguinte função de pertinência:

$$\mu_{\mathbf{R}}(a, b) = \begin{cases} 1, & \text{se e somente se } (a, b) \in \mathbf{R} \\ 0, & \text{em caso contrário.} \end{cases} \quad (2.50)$$

Relações *fuzzy* binárias (Definição 2.7.4) (Zadeh, 1965; Zadeh, 1971) generalizam o conceito de relações e representam o grau da associação, interação ou inteligência entre os elementos de dois conjuntos *fuzzy* (Mendel, 1995). Um exemplo de relação binária *fuzzy* seria: “ a é próximo de b ”.

Definição 2.7.4 *Sejam A e B dois universos de discurso. Uma relação binária fuzzy $\mathbf{R}(A, B)$ é um conjunto fuzzy pertencente ao produto cartesiano $A \times B$:*

$$\mathbf{R} : A \times B \rightarrow [0, 1], \quad (2.51)$$

caracterizada por uma função de pertinência $\mu_{\mathbf{R}}(A, B) \in [0, 1]$, onde $a \in A$ e $b \in B$.

A forma mais comum de se representar uma relação binária (*crisp* ou *fuzzy*) em $A \times B$ (em universos de discurso discretos) é através de uma matriz, chamada matriz relacional, cujos elementos são $\mathbf{R}(a_i, b_j) = [r_{ij}]_{n \times m}$, com $i = 1, \dots, n$ e $j = 1, \dots, m$.

Por exemplo, sejam \mathcal{A}_j e \mathcal{B} os conjuntos de termos lingüísticos definidos para as variáveis de entrada (antecedentes) e saída (conseqüente), respectivamente, de um modelo *fuzzy*.

$$\mathcal{A}_j = \{A_{j1}, A_{j2}, \dots, A_{jN_j}\}, \quad (2.52)$$

$$\mathcal{B} = \{B_1, B_2, \dots, B_K\}, \quad (2.53)$$

onde $j = 1, \dots, n$, n é o número de variáveis de entrada, N_j é o número de termos da j -ésima variável de entrada e K é o número de termos da variável de saída.

A base de regras pode ser representada por uma relação *crisp* \mathbf{R} entre os termos lingüísticos dos antecedentes e do conseqüente da seguinte forma:

$$\mathbf{R} : \mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_n \times \mathcal{B} \rightarrow \{0, 1\}. \quad (2.54)$$

Denotando $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_n$, o produto cartesiano dos termos lingüísticos do antecedente, a equação (2.54) pode ser simplificada para:

$$\mathbf{R} : \mathcal{A} \times \mathcal{B} \rightarrow \{0, 1\}. \quad (2.55)$$

Um modelo *fuzzy* relacional pode ser obtido pela generalização de \mathbf{R} (equação (2.55)) para uma relação *fuzzy*:

$$\mathbf{R} : \mathcal{A} \times \mathcal{B} \rightarrow [0, 1]. \quad (2.56)$$

A inferência do modelo *fuzzy* relacional é realizada em três passos (Babůska, 1999).

(i) Calcular o grau de ativação:

$$\beta_i^R = \min(\mu_{A_{i1}}, \dots, \mu_{A_{ip}}), \quad i = 1, 2, \dots, M. \quad (2.57)$$

(ii) Aplicar a composição relacional $\varpi = \beta^R \circ \mathbf{R}$, $\mathcal{R} = [r_{ij}]_{n \times K}$, dada pela equação (2.57):

$$\varpi_j = \max_{1 \leq i \leq M} (\beta_i^R \wedge r_{ij}), \quad j = 1, \dots, K. \quad (2.58)$$

(iii) Defuzificar o conjunto *fuzzy* conseqüente:

$$y = \frac{\sum_{l=1}^K \varpi_l b_l}{\sum_{l=1}^K \varpi_l} \quad (2.59)$$

onde b_l é o resultado da defuzificação do conjunto *fuzzy* \mathcal{B}_l através do método centro de área, dada pela equação (2.43).

2.7.4 Algoritmo *Fuzzy C-Means* - *FCM*

Um algoritmo de agrupamento realiza, essencialmente, a tarefa de particionamento de um conjunto de padrões em um número de *clusters* (grupos) homogêneos com respeito a uma determinada medida de similaridade. Os padrões pertencentes a qualquer um dos *clusters* são similares entre si, e os padrões de *clusters* distintos são tão dissimilares quanto possível (Xie & Beni, 1991). Na análise clássica de *clusters*, o limite dos diferentes *clusters* é *crisp* (i.e., rígido), tal que um padrão é atribuído a somente um *cluster*. Na prática, os dados geralmente não são tão bem distribuídos, sendo que o limite pode não ser precisamente definido. Portanto, um mesmo dado pode pertencer a dois ou mais *clusters* com diferentes graus de pertinência. Esta é a idéia por trás dos algoritmos de agrupamento *fuzzy*. Formalmente (Bezdek, 1987):

Definição 2.7.5 *Partição-C Fuzzy*: Seja um conjunto de dados $X = \{\mathbf{x}_1, \dots, \mathbf{x}_{N_d}\}$, onde $\mathbf{x}_k \in \mathbb{R}^n$ e N_d é o número de amostras em X . V_c é o conjunto de todas as $c \times N_d$ matrizes $H = [h_{ik}]$ e c é o número de *clusters*. O espaço da partição-*c fuzzy* para X é o conjunto:

$$M_{fc} = \{H \in V_c \mid h_{ik} \in [0, 1], 1 \leq i \leq c, 1 \leq k \leq N_d; e \sum_{i=1}^c h_{ik} = 1, \forall k \in \{1, 2, \dots, N_d\}\}, \quad (2.60)$$

onde h_{ik} é o valor de pertinência de \mathbf{x}_k ao cluster c_i .

Na literatura, existem três tipos de métodos para encontrar a partição ótima no espaço M_{fc} : métodos hierárquicos, métodos baseados em grafos, e métodos baseados em função objetivo. Os métodos baseados em função objetivo permitem, normalmente, a mais precisa (porém não necessariamente a mais válida) formulação do critério de agrupamento (Bezdek, 1987). Para o algoritmo *FCM* (*Fuzzy C-Means*) (Figura 2.12), o objetivo é encontrar $H = [h_{ik}] \in M_{fc}$ e $V = (\mathbf{v}_1, \dots, \mathbf{v}_c)$, $V \in V_c$, $\mathbf{v}_i \in \mathbb{R}^n$ tal que $J_m(H, V)$ (equação (2.61)) seja minimizada.

$$J_m(H, V) = \sum_{k=1}^{N_d} \sum_{i=1}^c (h_{ik})^m \|\mathbf{x}_k - \mathbf{v}_i\|^2, \quad (2.61)$$

onde $m \in (1, \infty)$ é uma constante de ponderação que determina a sobreposição dos *clusters* e \mathbf{v}_i é o centro do *cluster* i .

A otimização aproximada de J_m pelo algoritmo *FCM* é um procedimento iterativo baseado nas condições necessárias apresentadas pelo Teorema 2.2 ((Bezdek & Pal, 1995), (Bezdek, 1987)):

Teorema 2.2 *Se $\|z^l - \mathbf{v}_i\|^2 > 0 \forall i, l$, então $\mathcal{U} = [u_{il}] \in M_{fc}$ e $V = (\mathbf{v}_1, \dots, \mathbf{v}_c)$ é um mínimo local para $J_m(\mathcal{U}, V)$ somente se:*

$$u_{il} = \frac{1}{\sum_{j=1}^c \left(\frac{\|z^l - \mathbf{v}_i\|}{\|z^l - \mathbf{v}_j\|} \right)^{\frac{2}{m-1}}}, \quad 1 \leq i \leq c, \quad 1 \leq l \leq N; \quad (2.64)$$

e

$$\mathbf{v}_i = \frac{\sum_{l=1}^N (u_{il})^m z^l}{\sum_{l=1}^N (u_{il})^m}, \quad 1 \leq i \leq c. \quad (2.65)$$

Algoritmo 2: Agrupamento *Fuzzy C-Means*.

- **Passo 1:** Para um dado conjunto de dados X , estipule $c \in \{2, 3, \dots, N_d - 1\}$, $m \in (1, \infty)$, e inicialize $H^{(0)} \in M_{fc}$.
- **Passo 2:** Na iteração q , $q = 0, 1, 2, \dots$, calcule os vetores de V , com o novo centro de cada *cluster* C_i :

$$\mathbf{v}_i^{(q)} = \frac{\sum_{k=1}^{N_d} \mathbf{x}_k (h_{ik}^{(q)})^m}{\sum_{k=1}^{N_d} (h_{ik}^{(q)})^m} \quad (2.62)$$

- **Passo 3:** Atualize $H^{(q)} = [h_{ik}^{(q)}]$ para $H^{(q+1)} = [h_{ik}^{(q+1)}]$ tal que:

$$h_{ik}^{(q+1)} = \frac{1}{\sum_{j=1}^c \left(\frac{\|\mathbf{x}_k - \mathbf{v}_i^{(q)}\|}{\|\mathbf{x}_k - \mathbf{v}_j^{(q)}\|} \right)^{\frac{2}{m-1}}} \quad (2.63)$$

onde: $1 \leq i \leq c$, $1 \leq k \leq N_d$

- **Passo 4:** Se $\|H^{(q+1)} - H^{(q)}\| < \epsilon$, então pare o processo; caso contrário, incremente q , $q = q + 1$ e volte ao passo 2.

Figura 2.12: Algoritmo FCM.

2.8 Identificação de Modelos Fuzzy

Esta seção refere-se à fase de desenvolvimento do modelo (conforme a Figura 2.1), enfatizando, porém, a identificação de modelos *fuzzy*.

As duas fontes de informação disponíveis para a construção de modelos *fuzzy* lingüísticos são o conhecimento especialista e dados amostrados. Os dados estão disponíveis como registros da operação do processo ou podem ser obtidos através de experimentos especiais projetados para se obter dados relevantes. A construção de modelos *fuzzy* a partir de dados não envolve somente métodos baseados na lógica *fuzzy* e no raciocínio aproximado, mas também utiliza conceitos provenientes da identificação clássica de sistemas (Ljung, 2000). A aquisição ou sintonia de modelos *fuzzy* a partir de dados é usualmente chamada de identificação *fuzzy*.

Assim como no procedimento usual de identificação de sistemas (Ljung, 2000; Soderström & Stoica, 1989), com respeito ao projeto/identificação de modelos *fuzzy*, dois itens básicos devem ser considerados: a estrutura e os parâmetros do modelo. A estrutura determina a flexibilidade do modelo na aproximação dos mapeamentos. Os parâmetros são estimados de forma a ajustá-los aos dados disponíveis. Um

modelo com uma boa estrutura (*rich structure*) é capaz de aproximar funções mais complexas, mas ao mesmo tempo, possui propriedades inferiores de generalização. Boa generalização significa que o modelo “ajustado” para um determinado conjunto de dados também deverá apresentar um bom desempenho para um outro conjunto de dados do mesmo processo.

Em modelos *fuzzy*, a determinação da estrutura envolve as seguintes escolhas:

- *Variáveis de entrada e saída*: Em sistemas complexos, a escolha das variáveis que devem ser utilizadas como entrada do modelo nem sempre é trivial. No caso de sistemas dinâmicos, é necessário, também, estimar a ordem do sistema. Isso pode ser feito através do conhecimento prévio a respeito do sistema, ou através de uma seleção automática via dados amostrados;
- *Estrutura das regras*: Esta escolha está relacionada ao tipo do modelo (lingüístico, *Takagi-Sugeno*) e a forma do antecedente da regra;
- Número e forma das funções de pertinência para cada variável: Estas definições determinam o nível de precisão (granularidade ⁸) do modelo;
- *Mecanismo de inferência e método de defuzificação*: Estas duas escolhas dependem do tipo de modelo selecionado.

Após a definição da estrutura, a qualidade do modelo *fuzzy* pode ser otimizada ajustando-se os seus parâmetros. Os parâmetros sintonizáveis de um modelo lingüístico são os parâmetros das funções de pertinência do antecedente e conseqüente (formato e posição), e as regras (mapeamento entre as regiões antecedente e conseqüente). Nos modelos *Takagi-Sugeno*, são os parâmetros das funções de pertinência do antecedente e das funções matemáticas do conseqüente.

Nos casos em que há a disponibilidade de conhecimento especialista, é possível realizar a sua integração à identificação via dados amostrados de acordo com duas abordagens principais (Babůska & Vebruggen, 1996):

- O conhecimento especialista expresso na forma verbal é traduzido em uma coleção de regras *fuzzy se-então*. Desta forma, a estrutura inicial do modelo é criada. Os parâmetros desta estrutura (p. ex. formato e posicionamento das funções de pertinência) podem ser sintonizados utilizando-se dados de entrada e saída;

⁸Quanto maior o número de funções de pertinência, maior a granularidade.

- Nenhum conhecimento *a priori* a respeito do problema é inicialmente utilizado para a formulação das regras. O modelo *fuzzy* é obtido a partir dos dados amostrados. Com isto, espera-se que as regras e as funções de pertinência extraídas forneçam uma descrição inicial do comportamento do sistema. Em posse dessa descrição, um especialista pode, então, confrontá-la com o seu conhecimento a respeito do problema, e realizar alterações na base de regras obtidas como uma forma de sintonia.

Existe uma propriedade da base de conhecimento que deve ser observada. Trata-se da propriedade de completude do sistema *fuzzy*, relacionada tanto à base de dados quanto à base de regras, sendo tratada de forma distinta em cada caso. A propriedade de completude refere-se à capacidade dos sistemas *fuzzy* de, para qualquer entrada, inferir uma saída apropriada, ou seja, tais sistemas devem cobrir todos os estados possíveis, dentro de seus domínios. Sob o ponto de vista da base de dados, a união do conjunto suporte⁹ de todos os conjuntos *fuzzy* de uma determinada variável do sistema deve cobrir todo o universo de discurso desta variável. Na base de regras esta propriedade é satisfeita se for garantido que sempre existirá ao menos uma regra que será disparada para qualquer entrada possível (Lee, 1990).

2.9 Validação do Modelo

A fase de validação do modelo gerado verifica se este modelo é suficientemente adequado conforme um determinado critério de qualidade estipulado *a priori*. No caso do procedimento de identificação de sistemas convencional, a escolha do melhor modelo deve considerar o compromisso entre a flexibilidade e a simplicidade. Segundo este compromisso, um modelo deve ser suficientemente abrangente, ou seja, aplicável a vários pontos de operação, mas sem empregar muitos parâmetros, pois a variância costuma aumentar conforme o número de parâmetros estimados aumenta (Ljung, 2000).

Considerando-se identificação de modelos *fuzzy*, em especial a metodologia proposta neste trabalho, a validação do modelo é realizada através da análise da estrutura do modelo gerado e do seu desempenho alcançado durante a aplicação dos dados de validação. O compromisso entre a precisão do modelo, na geração das estimativas, e a sua capacidade de generalização é o fator determinante na escolha do

⁹O conjunto suporte é a área efetiva de um conjunto *fuzzy*, que representa os valores da função de pertinência maiores que zero (Zimmermann, 1991).

modelo final. Além disso, é necessário manter a transparência do modelo *fuzzy*, pois trata-se de uma das principais características e vantagens destes modelos.

A interpretabilidade de um modelo *fuzzy* está relacionada com a possibilidade de análise do processo modelado através da base de regras do modelo *fuzzy* e das funções de pertinência das variáveis do modelo. Quanto menor o número de funções de pertinência, que descrevem o universo de discurso de cada variável, e de regras, que descrevem o comportamento do processo, maior é o nível de interpretabilidade do modelo *fuzzy*. Porém, a precisão das estimativas pode ser comprometida pelo número reduzido de regras e funções de pertinência. Desta forma, durante a fase de validação, deve-se avaliar a qualidade do modelo *fuzzy* de modo que este gere estimativas precisas sem comprometer a sua interpretabilidade (Guillaume, 2001; Babůska, 2002; Abonyi et al., 2003; Jiménez et al., 2003).

2.10 Metodologia Proposta para a Identificação de Modelos *Fuzzy*

Na metodologia proposta neste trabalho, modelos *fuzzy* são identificados via dados amostrados, sem o uso de conhecimento especialista sobre o sistema. Alguns itens referentes à estrutura do modelo *fuzzy* são estabelecidos *a priori*. Determinou-se o uso de modelos lingüísticos, funções de pertinência do tipo *gbell*¹⁰, o mecanismo de inferência e o método de defuzificação (os dois últimos itens foram determinados de forma empírica). Por conseguinte, a metodologia proposta visa os seguintes problemas:

- seleção das variáveis de entrada do modelo *fuzzy*, dentre um conjunto de variáveis secundárias disponíveis;
- seleção da ordem do modelo *fuzzy*, i.e., o número de atrasos das variáveis de entrada e saída para compor o conjunto de regressores que será utilizado no procedimento de identificação do modelo; e
- geração da base de regras do modelo *fuzzy* e especificação dos parâmetros referentes às funções de pertinência de cada variável do modelo.

Na Figura 2.13 é possível visualizar a seqüência dos passos da metodologia proposta para a identificação de modelos *fuzzy*. O fluxograma é similar ao apresentado

¹⁰Generalized bell (Jang et al., 1997).

pela Figura 2.1. O primeiro passo é a seleção da estrutura do modelo *fuzzy*, que corresponde à fase de seleção do conjunto de variáveis de entrada e do conjunto de regressores. O segundo passo refere-se à geração da base de regras inicial. Esse modelo inicial é simplificado no terceiro passo e, finalizando, o quarto passo verifica e trata das possíveis regras inconsistentes¹¹. O modelo *fuzzy* inferencial identificado passa por uma fase de validação, onde é calculado o erro de estimação do modelo. Se o erro alcançado for menor que um limite pré-estabelecido, então a identificação é finalizada. Caso contrário, é necessário retornar ao início do procedimento de identificação.

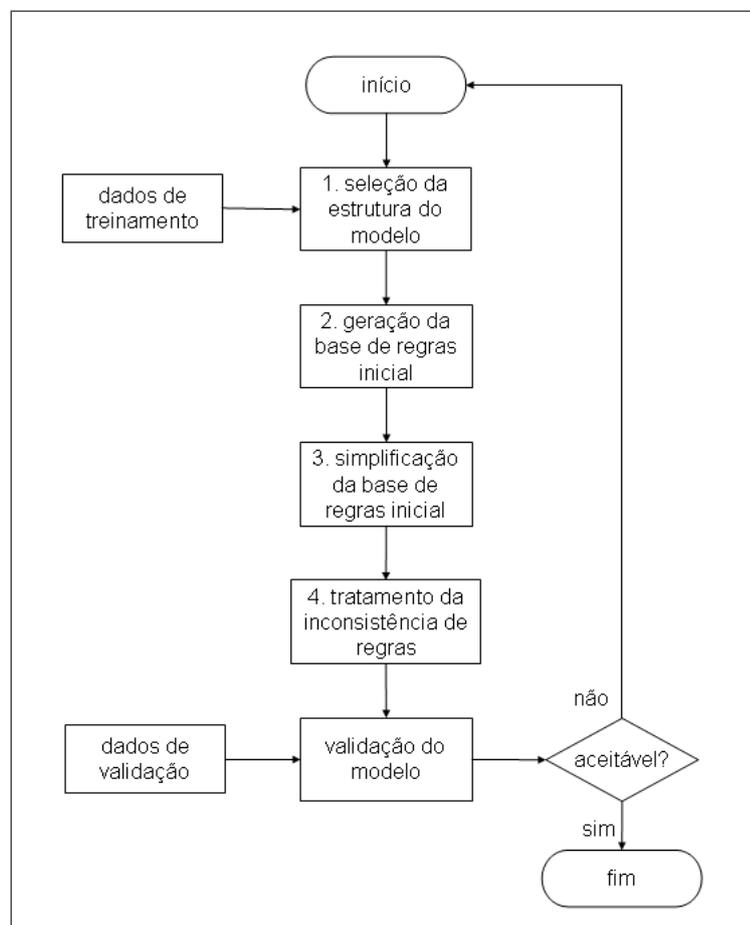


Figura 2.13: Fluxograma da metodologia proposta para a identificação de modelos fuzzy inferenciais.

A metodologia para a construção de modelos inferenciais, dividida em quatro etapas, é uma das principais contribuições deste trabalho. Para o desenvolvimento de cada etapa, além de se pesquisar as diversas técnicas de modelagem possíveis, levou-se em consideração, principalmente, os aspectos práticos para a identificação

¹¹Regras inconsistentes são aquelas que possuem o mesmo antecedente e conseqüentes distintos.

de sistemas conforme proposto em Ljung (2000) e Soderström & Stoica (1989).

Para a escolha da metodologia mais apropriada para cada etapa realizou-se um extenso levantamento bibliográfico sobre o assunto, o qual é resumido no capítulo 3. No capítulo 4 a metodologia proposta é abordada com mais detalhes.

Capítulo 3

Estado da Arte

Este capítulo é dedicado ao levantamento bibliográfico referente ao desenvolvimento de modelos inferenciais. Os trabalhos foram classificados de acordo com a metodologia utilizada para a inferência da variável de qualidade.

3.1 Métodos Estatísticos

Kresta et al. (1994) apresentam uma metodologia para o desenvolvimento de modelos inferenciais baseado em *PLS*, onde os dados utilizados na inferência são linearizados através de uma transformação logarítmica. Nesta proposta foram consideradas todas as variáveis medidas (disponíveis) para realizar a inferência, transformando-as em variáveis latentes, reduzindo o espaço de entrada de 45 variáveis secundárias para 3 variáveis latentes. Segundo os autores, o emprego de todas as variáveis secundárias através das variáveis latentes torna o modelo inferencial mais robusto às possíveis falhas dos sensores físicos.

Dayal & MacGregort (1997) apresentam um algoritmo para atualizar modelos *PLS* de forma recursiva reduzindo-se exponencialmente os dados passados que já foram apresentados. Este algoritmo foi desenvolvido combinando-se um algoritmo *kernel* (Dayal & MacGregort, 1997) melhorado e mais rápido com a atualização recursiva das matrizes de covariância ($X^T X$) e ($X^T Y$). O potencial deste algoritmo *PLS* recursivo foi apresentado através de dois exemplos, e o seu desempenho foi superior ao algoritmo *RLS* (*Recursive Least Squares*) em ambos os casos. Uma das vantagens do algoritmo *PLS* recursivo é que ele não sofre severamente dos problemas associados às variáveis correlacionadas do processo e ao tamanho reduzido da janela

de dados. Além disso, este algoritmo fornece um erro quadrático médio menor do que o algoritmo *RLS* no caso do controle adaptativo.

Em Kano et al. (2000) modelos inferenciais para a estimação de composição de produtos são construídos utilizando-se um *PLS* dinâmico. Nesta metodologia, um modelo *PLS* dinâmico, que pode estimar a composição dos produtos a partir de múltiplas temperaturas e outras variáveis de processo disponíveis *on-line*, é projetado para uma coluna de destilação multi-componentes. Medições de vários instantes de tempo passados são utilizados como variáveis de entrada de modo a incorporar a dinâmica do processo no modelo *PLS*. Os exemplos deste artigo indicam que o uso de medições de temperatura de cinco pratos são o suficiente para estimar a composição, tanto da base quanto do topo da coluna. A aplicação do modelo inferencial para controlar a composição de produtos também foi investigada. Foi desenvolvido um sistema de controle em cascata composto por uma malha interna de controle de temperatura e uma malha externa de controle inferencial da composição, cujos resultados apresentaram-se satisfatórios.

Em Barolo et al. (2002) várias abordagens baseadas em *PLS* são utilizadas para estimar a qualidade dos produtos obtidos de uma coluna de destilação em batelada, não somente no final do processo, mas também durante toda a duração da batelada. Dentre essas abordagens estão a *PLS* linear, *PLS* não-linear, e *PLS* baseada em redes neurais. Os modelos são estimados a partir de medidas de variáveis secundárias do processo e de um banco de dados contendo o histórico de simulações do processo. Neste mesmo artigo é proposta uma nova abordagem baseada em *PLS*, o *MPLS* (*Multiple PLS regression*), onde são desenvolvidos múltiplos modelos de regressão para as diferentes fases do processo em batelada. Foram realizados testes de validação dos modelos baseados nas diferentes abordagens, e concluiu-se que o *MPLS* fornece a melhor representação do processo, em particular quando o objetivo é estimar a composição do produto final da destilação.

O melhor desempenho apresentado pelo *MPLS* deve-se ao fato desta abordagem levar em consideração as características peculiares do processo de destilação em batelada. Este processo apresenta uma série de fases substancialmente distintas entre si, e as mudanças de regime da dinâmica da coluna e da distribuição da composição são refletidas na variabilidade no perfil da temperatura, justamente os dados que a regressão *PLS* utiliza para a estimar o modelo. Desta forma, em contraste com o *PLS* convencional, o *MPLS* é capaz de lidar diretamente com essa complexidade e variedade de informações sobre as diferentes fase do processo.

Em Kano et al. (2003) é proposta uma solução, baseada nos resultados obtidos em Kano et al. (2000), para o problema de como projetar um sistema de controle inferencial, para melhorar o desempenho do controle de composição de uma coluna de destilação. Neste artigo é proposto o “controle inferencial preditivo”, onde é desenvolvido um sistema de controle baseado em um modelo inferencial que prediz a composição futura do produto, ao invés de predizer a atual composição, empregando-se o método de regressão *PLS*. Ao realizar a predição, espera-se que as perturbações sejam detectadas e compensadas antes que elas afetem a composição dos produtos, melhorando, desta forma, o desempenho do controlador. A arquitetura de controle proposta foi, então, comparada com alguns sistemas de controle inferencial, e demonstrou-se que nem sempre os melhores desempenhos são obtidos por sistemas de controle baseados em modelos dinâmicos, embora os modelos dinâmicos sejam superiores aos modelos estáticos do ponto de vista de precisão das estimativas. Os resultados da pesquisa realizada neste artigo sugerem o uso do controle inferencial preditivo, com um modelo inferencial integrado a uma configuração de controle em cascata, para alcançar um bom desempenho sem a necessidade do uso de uma abordagem de modelagem dinâmica.

Uma abordagem para o desenvolvimento de modelos dinâmicos *on-line* para a estimação preditiva de variáveis de qualidade, na mesma taxa de amostragem das variáveis do processo, é proposta em (Lu et al., 2004). Esta metodologia baseia-se na proposta apresentada em Lu & Fisher (1992), com algumas modificações para tratar sistemas multivariáveis. Dependendo da natureza da correlação entre as variáveis do processo e as variáveis de qualidade, duas formas diferentes do modelo são propostas, onde o *MwPLS* (*Multiway PLS*) (Wold et al., 1996; Wangen & Kowalshi, 1988) é empregado para explorar o relacionamento entre as variáveis. Para processos sem um controle de qualidade em malha fechada, um modelo com dois blocos deve ser utilizado, enquanto que para o caso do controle de qualidade em malha fechada, um modelo com um bloco mostra-se mais apropriado. Ambos os modelos utilizam diretamente as variáveis do processo (com alta taxa de amostragem) e as variáveis de qualidade (com baixa taxa de amostragem), sem a necessidade de uma re-amostragem, para predizer as variáveis de qualidade em uma alta taxa de amostragem. A metodologia proposta foi utilizada no desenvolvimento de um modelo inferencial do processo *Tennessee Eastman (TE)* (Downs & Vogel, 1993) a fim de mostrar a habilidade de predição *on-line* do modelo desenvolvido.

Plantas de dessalinização/desidratação (*DDP - Desalting/Dehydration Plants*) são geralmente instaladas em unidades de produção de óleo cru a fim de remover o

sal dissolvido em água no óleo de alimentação. Abdul-Wahab et al. (2006) descrevem o desenvolvimento de estimadores para a inferência da qualidade de produto do processo de dessalinização/desidratação. Os estimadores inferenciais foram construídos para capturar o relacionamento entre a qualidade do produto da planta e as variáveis de entrada do processo. Foram selecionadas cinco variáveis de entrada cuja influência sobre a qualidade do produto é conhecida. Essas variáveis são: temperatura, tempo de ajuste, tempo de mistura, dosagem química e vazão específica. A qualidade do produto do processo de dessalinização/desidratação foi identificada pela eficiência na remoção do sal e no teor de água. Portanto, os estimadores inferenciais foram usados para inferir a eficiência na remoção de sal e no teor de água a partir das cinco variáveis de entrada do processo. Estes estimadores inferenciais foram construídos baseados na aplicação da análise de componentes principais (*PCA*) e também de regressão não-linear. Os relacionamentos não-lineares entre as variáveis de entrada e as variáveis de qualidade foram identificados através da regressão não-linear, e esses modelos não-lineares apresentaram resultados superiores aos modelos lineares gerados pelo *PCA*. Segundo os autores, este foi uma pesquisa preliminar e estudos futuros para compreender de forma mais aprofundada os efeitos das variáveis de entrada nas variáveis de qualidade ainda serão realizados.

3.2 Métodos Baseados em Filtro de Kalman

Em Tham et al. (1991) são apresentados dois algoritmos que utilizam dados amostrados do processo a diferentes taxas para estimar a saída controlada (cujas medidas são obtidas de forma infreqüente). O primeiro algoritmo de estimação é baseado em uma representação geral de entrada e saída do processo. Neste primeiro caso, o estimador é obtido representando-se a dinâmica do processo através dos relacionamentos entre as variáveis de entradas e saídas. Como as funções de transferência que representam esses relacionamentos não são conhecidas *a priori*, os seus parâmetros são identificados *on-line*. O segundo algoritmo é derivado de uma representação de espaço de estados da planta. Nesta abordagem, o objetivo é estimar os estados do sistema utilizando-se observações das saídas secundárias e, então, obter uma estimativa da saída controlada utilizando-se um Filtro de Kalman. Neste artigo, os dois algoritmos são aplicados em processos industriais, tais como coluna de destilação e reator de polimerização, de forma bem sucedida. Os estimadores adaptativos apresentados também podem ser incluídos como parte de um sistema de controle com realimentação para fornecer uma estratégia de controle inferencial adaptativo. A

praticabilidade deste esquema para efetuar o controle de composição do produto de destilação foi avaliado por uma simulação não-linear. A habilidade deste esquema em antecipar os efeitos da perturbação resulta em uma melhora significativa no comportamento do sistema de controle, apesar da escolha da variável secundária não ter sido a ideal.

Wilson et al. (1998) desenvolveram um estimador baseado em um filtro de Kalman estendido, *Extended Kalman Filter (EKF)*, para um reator em batelada industrial, onde tentou-se enfatizar tanto as vantagens quanto as deficiências do uso *on-line* deste estimador em condições industriais. Para a aplicação escolhida (reator em batelada), a estimação *on-line* não alcançou o desempenho esperado devido a três fatores principais: (i) pobre observabilidade do modelo (exceto próximo do fim da batelada (*batch*)); (ii) pobre medição da taxa da geração de calor de reação; e (iii) pobres medições de concentração para a verificação, especialmente no final da batelada. Outra dificuldade apresentada foi encontrar processos industriais que não fossem tão triviais, demasiadamente confidenciais ou até mesmo arriscados para o desenvolvimento do estimador, e cuja amostragem para a verificação *off-line* fosse viável. A partir da experiência obtida durante a implementação do estimador, os autores apresentaram uma lista de considerações que se mostraram relevantes durante o processo de implementação de um modelo inferencial.

Herbst & Pate (1999) desenvolveram uma abordagem orientada a objeto para o desenvolvimento de *soft sensors*. Trata-se de um *framework* genérico composto por quatro componentes (objetos), onde cada um desses componentes deve ser criado de acordo com o processo em questão. Os componentes são: um para o modelo do processo, um para o estimador sequencial, um para a interface de gerenciamento das entradas do sensor físico e um para a interface de gerenciamento das saídas do *soft sensor*. Esses quatro componentes são “encapsulados” em um *container* para montar o *soft sensor*. O componente do estimador sequencial utiliza um algoritmo do filtro de Kalman generalizado, *Generalized Kalman Filter (GKF)*. Os componentes de interface permitem que o engenheiro configure o problema, isto é, especifique os estados, medidas e parâmetros, e gereencie as entradas do *soft sensor* e as saídas do sensor físico. A vantagem apresentada por esta abordagem é a relativa facilidade com que um *soft sensor* pode ser desenvolvido para um dado processo utilizando-se o esquema de componentes, sendo necessário somente uma reconfiguração para cada nova aplicação.

Em Oisiovici & Cruz (2000) um estimador não-linear e discreto baseado em um

filtro de Kalman estendido (*EKF*) é desenvolvido e testado. A meta do *EKF* é fornecer um perfil confiável e em tempo real da composição de uma coluna de destilação a partir de poucas medidas de temperatura e informações facilmente disponíveis. Estimativas precisas da composição e a rápida convergência foram obtidas e o *EKF* confirmou a sua habilidade para incorporar os efeitos de ruído (de medida e modelagem). O número de sensores e a frequência de observação mostraram-se como variáveis importantes no projeto do *EKF*, especialmente para sistemas com dinâmica rápida.

3.3 Métodos Baseados em Inteligência Computacional

Numa das primeiras contribuições na área de *soft sensing*, Willis et al. (1992) discutiram a aplicabilidade das redes neurais no desenvolvimento de estimadores inferenciais e controladores preditivos não-lineares. Verificou-se que um estimador neural poderia fornecer uma rápida inferência de uma variável de saída difícil de se medir a partir de outras variáveis do processo (facilmente obtidas). Num experimento, onde dados obtidos de um processo industrial foram utilizados para inferir um modelo, mostrou-se que, dada uma topologia apropriada para a rede neural, esta rede poderia ser treinada para caracterizar efetivamente o comportamento do processo em questão, confirmando o potencial das redes neurais como *soft sensors*. Além disso, foi mostrado que melhorias significativas no controle de processos poderiam ser alcançadas se as estimativas produzidas pelo estimador neural fossem utilizadas como sinais de realimentação para o esquema de controle. Por fim, o desenvolvimento e aplicação de um esquema de controle preditivo baseado no modelo neural do processo são considerados, onde este controlador não-linear mostrou-se superior às técnicas lineares convencionais. Porém, os autores ressaltam que o procedimento empregado no controle preditivo neural é iterativo e, desta forma, o algoritmo pode ser aplicado somente em casos onde o período de amostragem é suficientemente grande.

Wang & Rong (1999) propuseram o (*NNFS - Neural-Network -Based Fuzzy System*), ou rede neural *fuzzy*, para a construção de um *soft sensor* para uma coluna de destilação. Trata-se de uma rede neural auto-organizável capaz de particionar o espaço de entrada de forma flexível, baseando-se na distribuição dos dados de treinamento, de modo a reduzir o número de regras e mantendo a precisão da modelagem. Um algoritmo de aprendizagem híbrido em duas fases é empregado. A

primeira fase utiliza um esquema de agrupamento do vizinho mais próximo para a aprendizagem da estrutura e a determinação dos parâmetros iniciais do modelo. E a segunda fase utiliza o método de gradiente para a sintonia fina dos parâmetros da rede neural. Combinando-se estes dois métodos, a aprendizagem converge muito mais rapidamente do que o algoritmo *backpropagation* original. Os resultados de simulação sugerem que os bons resultados obtidos pelo *NNFS* devem-se aos seguintes fatores: simplicidade da estrutura, rapidez da aprendizagem, número reduzido de regras *fuzzy* e exatidão relativa da modelagem.

Em Meleiro & Maciel-Filho (2000), um controlador auto-sintonizável (*STC - Self - Tuning Controller*) foi desenvolvido para a aplicação em um processo biotecnológico. Devido às dificuldades envolvendo as medições ou ao número excessivo de variáveis, normalmente encontradas na indústria de processos, os autores recomendam o uso de redes neurais para o desenvolvimento de modelos inferenciais, pois estas são especialmente adequadas para a identificação de modelos não-lineares e variantes no tempo. Uma estratégia avançada de controle baseada em *STC* foi aplicada a um processo de fermentação para produzir etanol (álcool etílico) em escala industrial. Os resultados obtidos mostram que o procedimento proposto neste trabalho tem um grande potencial para esta área de aplicação. A aplicação da estratégia proposta ao sistema contínuo de produção de etanol disponibilizou *on-line* os valores das variáveis desejadas. Esses valores forneceram uma boa descrição dos estados do sistema, permitindo a monitoração e a execução detalhadas de um controle automático avançado eficiente do processo.

Três metodologias diferentes baseadas em redes neurais para o desenvolvimento de modelos inferenciais são propostas em Rallo et al. (2002), e seus resultados experimentais foram comparados. O objetivo do desenvolvimento destes modelos é inferir a qualidade de um produto a partir de outras variáveis do processo medidas mais facilmente. Um algoritmo preditivo *fuzzy ARTMAP* (Carpenter et al., 1991) é uma das arquiteturas consideradas. As outras duas arquiteturas são baseadas em dois algoritmos supervisionados que conectam uma camada *RBF* (*Radial Basis Function*) com a saída requerida. Esta camada *RBF* foi construída seguindo um procedimento inovador, que permite o crescimento dinâmico da rede durante a fase de treinamento. Diferentemente da maioria das abordagens de desenvolvimento de *soft sensors*, que, e.g., empregam análises estatísticas, neste artigo foi utilizado um mapa de Kohonen (*SOM*) para selecionar as variáveis secundárias utilizadas na inferência do modelo. Para fins de validação, as abordagens foram aplicadas para inferir (*on-line*) a qualidade de seis diferentes classes de polietilenos de baixa densidade (*LDPE - Low*

Density Polyethylene) (Lines et al., 1993). O desempenho obtido pelas três abordagens foi satisfatório, inclusive na rápida adaptação às novas condições de operação, principalmente devido ao reduzido conjunto de variáveis de entrada. Porém, essa redução de variáveis de entrada aumenta o desvio padrão, indicando que um algoritmo de treinamento mais elaborado deveria ser empregado.

Diferentemente da maioria das aplicações apresentadas na área de modelagem e controle inferencial, Yeh et al. (2003) desenvolveram uma metodologia para a estimação inferencial de composição em *plantwide* (sistemas de grande porte). Segundo os autores, os processos são, em geral, representados como um conjunto de unidades de operações interconectadas, e as composições de uma coluna podem ser influenciadas pelas condições de operação de outras unidades, e vice-versa. Desta forma, a estimação de composição sob as considerações de uma *plantwide* apresenta-se mais interessante e apropriada. O estimador foi implementado com uma rede neural onde todos os neurônios da camada escondida e da camada de saída são idênticos, e a função de ativação de cada neurônio é uma tangente hiperbólica. Com o intuito de melhorar o desempenho na predição das saídas do processo, foram consideradas as redes neurais recorrentes, onde existe a realimentação da rede neural com saídas da própria rede. A estrutura da rede foi selecionada através da técnica de validação cruzada (Pollard et al., 1992) e o treinamento foi realizado com o algoritmo *back-propagation* (foram utilizados os valores de transiente da composição dos produtos e das variáveis secundárias como dados de treinamento). Para verificar a validade do estimador neural, duas técnicas de controle industrial, o clássico *PI* e o controle por modelo preditivo (*MPC - Model Predicted Control*), juntamente com o estimador neural, foram investigadas para regular as composições do processo na planta *TE* (Downs & Vogel, 1993). Os resultados de simulação mostraram que um estimador neural com calibração em laboratório é capaz de realizar estimações *on-line* das composições do processo a partir de medidas secundárias de forma satisfatória.

Um interessante estudo comparativo é apresentado em Fortuna et al. (2003), onde quatro estratégias para o desenvolvimento de *soft sensors* foram implementadas e testadas. O objetivo desses *soft sensors* é medir a concentração dos chamados gases ácidos H_2S e SO_2 em uma unidade de recuperação de enxofre (*Sulfur Recovery Unit - SRU*). Esses *soft sensors* consistem de modelos dinâmicos não-lineares capazes de prever a concentração dos dois gases separadamente. As variáveis empregadas para realizar a inferência foram selecionadas com base na observação da tendência de cada uma delas, ou seja, fez-se uso de conhecimento especialista para realizar a seleção de variáveis. As estratégias utilizadas foram: (i) rede neural *MLP*; (ii) rede neural *RBF*

(*Radial Basis Function*); (iii) rede neural Neuro-Fuzzy (*NF*); e (iv) mínimos quadrados não-linear (*Nonlinear Least Squares - NLSQ*). Na rede *MLP*, foi considerada uma rede com somente uma camada escondida, onde o número de neurônios desta camada foi selecionado por tentativa e erro, e todos com uma função de ativação sigmoideal. A rede *MLP* foi treinada com o algoritmo *backpropagation* Levenberg-Marquardt (Hagan & Menhaj, 1994). O segundo *soft sensor* implementado foi o baseado em uma rede neural *RBF*. Para selecionar a melhor topologia dessa rede foi adicionado um neurônio por vez até que um determinado índice de desempenho fosse atendido. Estipularam-se 298 neurônios para o modelo do H_2S e 268 neurônios para o modelo do SO_2 apesar de não apresentarem bons resultados. Melhores resultados poderiam ser obtidos com um número maior de neurônios, porém esta estratégia geraria uma complexidade proibitiva para a implementação *on-line*. A terceira metodologia é uma rede *NF*, onde o *ANFIS* (*Adaptive Neuro Fuzzy Inference System*) (Jang, 1993) foi empregado para a sua implementação. Os parâmetros das funções de pertinência das variáveis do sistema *fuzzy* foram escolhidos via agrupamento *FCM* e posteriormente refinados manualmente. Para a quarta metodologia *LSQ*, foi adotada uma função não-linear de segunda ordem para implementar o modelo dinâmico em consideração. Os coeficientes desta função foram calculados com um algoritmo não-linear de mínimos quadrados (Ponton & Klemes, 1993). Após a definição das metodologias, elas foram implementadas *off-line* com dados históricos do processo, que foram separados em dados de treinamento (aprendizagem) e dados de teste. Os modelos baseados em redes *RBF* exibiram os melhores desempenhos com os dados de aprendizagem, porém, não foram capazes de confirmar o mesmo desempenho com os dados de teste, apresentando os piores resultados. Os modelos *NF*, *LSQ* não-linear e *MLP* apresentaram desempenhos aproximados, com um leve destaque para os modelos *LSQ* não-linear e *MLP*. Baseando-se nesses resultados, os modelos *LSQ* não-linear e *MLP* foram implementados *on-line*, e após vários testes verificou-se que o modelo *MLP* foi capaz de manter o melhor desempenho em todos os períodos de operação observados.

Em Fabro (2003) foram desenvolvidos modelos inferenciais para estimar a composição dos produtos de base (água) e topo (metanol) durante o processo de inicialização de uma coluna de destilação binária. Para realizar a estimativa dessas composições foram utilizadas redes neurais. Estas redes recebem como entrada as temperaturas lidas em quatro pratos, e não possuem conexões recorrentes. Ambas as redes são do tipo direta (*feedforward*), com três camadas (51 neurônios na camada de entrada, 11 neurônios na camada escondida e 1 neurônio da camada de saída) e

foram treinadas com o algoritmo *quick-propagation* (Fahlmann, 1988) obtendo erros menores que 1%.

Fortuna et al. (2005) também desenvolveram uma abordagem onde redes neurais são empregadas no desenvolvimento de modelos inferenciais. Neste caso, o modelo inferencial é utilizado para o monitoramento e controle em uma refinaria, onde os valores de concentração da gasolina ($C5$) no topo e do butano ($C4$) na base de uma coluna debutanizadora são estimados através das variáveis secundárias. Foram construídas duas redes neurais, uma para cada medida de concentração. Esses modelos neurais são modelos dinâmicos preditivos em três passos, que fornecem as estimativas em tempo real. Os dados foram coletados de uma planta real. O número de amostras atrasadas foi determinado por tentativa e erro, através de conhecimento especialista do processo. Trata-se de uma rede neural *MLP*, com uma camada escondida e função de ativação sigmoideal. O número de neurônios da camada escondida também foi obtido através de tentativa e erro. As variáveis de entrada de cada modelo foram selecionadas baseado em conhecimento do processo. Para o modelo de inferência de $C5$ as variáveis de entrada são: (1) concentração do $C5$ no topo da coluna; (2) temperatura no topo; (3) pressão no topo; (4) refluxo no topo; (5) vazão no topo; (6) temperatura lateral; e (7) média das temperaturas na base da coluna. Para a inferência de $C4$, as entradas são as mesmas, exceto pela variável de realimentação, a concentração do $C4$ na base da coluna. Os dois modelos foram implementados para fornecer estimativas *on-line* numa planta real, sendo os modelos re-sintonizados periodicamente.

Mais uma abordagem baseada em redes neurais é apresentada em Singh et al. (2005). Neste artigo um estimador baseado em redes neurais multicamadas é construído para inferir a composição de produtos de uma coluna de destilação. Trata-se de uma rede neural cujas entradas são: a temperatura nos pratos da coluna, temperatura do refluxo e a temperatura no refeedor; e saídas são: a composição dos produtos de saída (sistema *MISO*). Segundo os autores, para incorporar as não-linearidades do processo, foram utilizadas três camadas escondidas. Foram realizados testes com mais camadas escondidas, porém não houve melhoria na precisão das estimativas geradas, e além disso, um número menor de camadas também não se mostrou apropriado. Essa rede neural foi avaliada para uma coluna com 15 pratos para estimar a composição de 5 produtos de saída. Os resultados obtidos mostraram-se precisos, com um baixo tempo de resposta.

3.4 Abordagens Híbridas

Em Park & Han (2000) é proposta uma nova metodologia não-linear para a construção de *soft sensors* capazes de manipular as correlações entre as variáveis do processo e as não-linearidades, cujo desenvolvimento foi motivado pela regressão localmente ponderada (*Locally Weighted Regression - LWR*) proposta por Cleveland & Devlin (1988). O método *LWR* é um procedimento para a estimação de uma superfície de regressão não-linear para dados através de *smoothing* multivariáveis. Como originalmente o método *LWR* não é capaz de lidar com alta dimensionalidade e colinearidade, o primeiro passo para a construção dos *soft sensors* é transformar as variáveis originais em novas variáveis (em número menor) através de transformações lineares como *PLS* ou *PCA*. Após esta redução, é realizada a modelagem através do *LWR*. O método proposto foi comparado com os métodos *PLS*, *PLS* não-linear e redes neurais, e embora não tenha obtido o melhor desempenho nos estudos de caso, a metodologia apresenta-se como uma boa opção para o desenvolvimento de *soft sensors* devido a sua simplicidade de aplicação em processos altamente não-lineares e com colinearidade significativa.

Bhartiya & Whiteley (2001) apresentam uma metodologia sistemática para o desenvolvimento de modelos inferenciais utilizando-se redes neurais. Esta metodologia é composta por três passos. O primeiro passo consiste da coleta e pré-processamento dos dados a serem utilizados na inferência. Durante esta fase, a técnica de componentes principais (*PC*) é empregada para a identificação de *outliers*. No próximo passo, as variáveis do processo são sujeitas a uma análise estatística simples para identificar um subconjunto de variáveis a serem utilizadas no esquema inferencial. Inicialmente os coeficientes de correlação parcial são utilizados para identificar as variáveis independentes do processo (aquelas que têm um efeito significativo na predição da resposta da variável) a partir da lista das variáveis disponíveis para realizar a inferência. Então, a estatística C_p (Mallow, 1973) é empregada para refinar a escolha do subconjunto de preditores a partir do subconjunto obtido com os coeficientes de correlação. O terceiro passo envolve a geração do esquema inferencial. Nesta fase, é empregada uma rede neural multicamadas com duas camadas, cujo treinamento é realizado com o algoritmo *backpropagation*. Para fins de validação a metodologia desenvolvida foi aplicada na inferência de um processo de refino. Verificou-se que o desempenho do modelo obtido depende do conhecimento *a priori* sobre o processo, principalmente na fase de identificação do conjunto de variáveis a serem empregadas na regressão.

Uma técnica de modelagem inferencial em batelada (*batch*) para identificar a correlação, batelada a batelada, dos dados do processo e de qualidade é proposta em Dorsey & Lee (2003). O método permite o uso em conjunto de informações contidas nas bateladas anteriores e também da batelada atual, com o intuito de prever as variáveis de qualidade através do desenvolvimento periódico de um filtro de Kalman (variante no tempo), sendo necessário aplicar, anteriormente, a análise de componentes principais (*PCA*) para reduzir a alta dimensionalidade dos dados em cada batelada. Este método de inferência pode ser incorporado em uma estratégia de controle, tal com o *MPC* (*Modelo Predictive Control*). Durante a fase de validação, verificou-se que a incorporação de informações sobre as bateladas anteriores na inferência *on-line* do modelo permitiu um melhor desempenho em comparação a modelos inferidos com somente dados de bateladas anteriores ou somente com dados da batelada atual.

O projeto de uma rede neural para um determinado problema é feito, geralmente, através de tentativa e erro e a quantidade de parâmetros ajustáveis no projeto de uma rede neural torna enfadonha esta tarefa. Em Dam & Saraf (2006) é proposta uma metodologia onde esse desenvolvimento da rede neural é realizado com o auxílio de um algoritmo genético. Nesse algoritmo genético o objetivo é encontrar o melhor conjunto de parâmetros para a rede neural, tais como o número de camadas escondidas, o número de neurônios em cada camada, a conectividade entre os neurônios e as funções de ativação. Além disso, um outro algoritmo genético é utilizado para selecionar o conjunto de variáveis de entrada da rede neural, dadas as variáveis secundárias disponíveis. Esta metodologia foi utilizada para a construção de redes neurais para a predição das propriedades de vários produtos laterais de uma coluna fracionadora de óleo cru, apresentando resultados satisfatórios, apesar da pouca disponibilidade de dados amostrados. Segundo os autores, trata-se de uma metodologia geral, que pode ser empregada no projeto de redes neurais para outras aplicações.

Geralmente, as variáveis de qualidade do polietileno são a densidade e o índice de fluidez (*MI - Melt Index*). Porém, a maioria das aplicações utiliza o controle do índice de fluidez, pois é mais difícil manter as especificações do *MI* do que da densidade (Kiparissides et al., 1993). O controle desse índice é normalmente realizado baseando-se em medidas de análise feitas em laboratório, realizadas a cada duas a quatro horas, tornando o procedimento de controle demorado e difícil. Neste cenário, o uso de modelos inferenciais é altamente apropriado para gerar as estimativas do *MI*. Em Liu (2006) é proposto um algoritmo para geração de um modelo inferencial (*soft sensor*), cujo objetivo é a obtenção de estimativas de *MI*. Este algoritmo

é capaz de gerar modelos de um sistema multi-variável, não-linear e variante no tempo. Inicialmente, o PCA é utilizado para eliminar as colinearidades e reduzir a dimensão do espaço de entrada. Então, o algoritmo FCM é empregado para decompor o espaço de operação em sub-regiões. Por último, é construído um modelo *fuzzy* Takagi-Sugeno para representar cada sub-região de operação. Os resultados de simulação mostraram que esta abordagem apresenta um desempenho superior ao apresentado por técnicas de modelagem não-linear, tais como o PLS não-linear. Segundo os autores, a grande vantagem desse algoritmo é a sua capacidade de adaptação à natureza variante no tempo dos eventos do processo. Porém, uma limitação é a impossibilidade de atualização *on-line* dos modelos *fuzzy*.

3.5 Comentários Finais

Através deste levantamento bibliográfico foi possível observar que o tema “modelos inferenciais” continua sendo atual e as pesquisas em torno desse assunto geram diversos artigos, onde os modelos inferenciais são aplicados nas mais diversas áreas da indústria. Este fato se deve principalmente à necessidade da indústria em aumentar a produtividade e também a qualidade do produto final.

Basicamente, o desenvolvimento de um modelo inferencial envolve algumas escolhas dentre elas a seleção das variáveis de entrada do modelo. Essa escolha é limitante do desempenho do modelo inferencial e, geralmente, é realizada por análise estatística, conhecimento *a priori* do processo ou por tentativa e erro.

Apesar de toda essa produção científica, o emprego da teoria dos conjuntos *fuzzy* neste problema ainda é pequena diante de outras técnicas de inteligência computacional. Emprega-se, em sua maioria, redes neurais devido, principalmente, a sua precisão e robustez. Porém, uma das desvantagens dessa metodologia é a falta de transparência do modelo inferencial obtido, pois as eventuais camadas escondidas e as diversas conexões entre os neurônios dificultam, ou até mesmo impossibilitam, a análise do comportamento do processo em questão através do modelo obtido.

A possibilidade da representação do conhecimento lingüístico em termos matemáticos e o raciocínio aproximado são algumas das características dos sistemas *fuzzy* que os tornam uma alternativa promissora na implementação de modelos inferenciais. Neste contexto propõe-se neste trabalho uma metodologia inovadora que tenta resolver os problemas relacionados ao desenvolvimento de modelos inferenciais através de um sistema baseado na teoria dos conjuntos *fuzzy*. O capítulo 4 discorre sobre

a metodologia proposta para a implementação de modelos inferenciais.

Capítulo 4

Metodologia Proposta

Este capítulo apresenta a metodologia proposta para a identificação de modelos *fuzzy* inferenciais que podem ser utilizados na construção de *soft sensors* para processos da indústria petroquímica.

Esta metodologia é composta por quatro etapas. A primeira etapa corresponde à seleção de variáveis de entrada do modelo inferencial, dado um conjunto de variáveis secundárias disponíveis do processo, com o auxílio de mapas auto-organizáveis de Kohonen. O conjunto de regressores, utilizado para realizar a identificação do modelo inferencial, é determinado através do cálculo dos quocientes de Lipschitz. Após esta fase inicial, um algoritmo de agrupamento *fuzzy* é aplicado ao espaço de entrada e saída, de modo a encontrar uma base de regras inicial capaz de modelar a interação entre as variáveis do processo, assim como as possíveis não-linearidades do processo. Na terceira etapa, as funções de pertinência similares, no domínio de cada variável, são agrupadas com o objetivo de aumentar a transparência do modelo obtido, tornando-o mais simples e conciso. Finalizando a metodologia, a existência de regras inconsistentes é verificada e tratada. Os procedimentos que compõem a metodologia proposta são descritos no decorrer deste capítulo.

4.1 Descrição do Problema

A metodologia proposta foi implementada com o intuito de identificar modelos *fuzzy* inferenciais para sistemas *MISO* (*multi-input single-output*) com n variáveis de entrada (selecionadas a partir das variáveis secundárias do processo) e uma variável de saída, onde a estrutura do modelo *fuzzy* consiste de uma coleção de regras *fuzzy*

do tipo Mamdani, como ilustra a Figura (2.9).

Assim como em um procedimento de identificação convencional, é necessária a definição dos dados que guiarão a identificação. O conjunto de dados é denotado por Z^{N_d} . O vetor definido pela equação (4.1) denota o t -ésimo par de dados de entrada e saída $\mathbf{z}(t) \in Z^{N_d}$, $t = 1, \dots, N_d$ coletado de um sistema com n entradas e uma saída, tratando-se, portanto, de um sistema *MISO*.

$$\mathbf{z}(t) = (\mathbf{x}(t), y(t)), \text{ onde } \mathbf{x}(t) = (x_1(t), \dots, x_n(t)). \quad (4.1)$$

Este conjunto de dados é pré-processado de modo a obter dados com média zero e desvio padrão unitário. Além disso, o universo de discurso de cada uma das variáveis do processo é normalizado no intervalo de $[0, 1]$. Da mesma forma, as variáveis do modelo *fuzzy* possuem seus valores nos universos de discurso U_1, \dots, U_{n+1} , onde $U_j = [0, 1]$, ($j = 1, \dots, n + 1$).

A primeira etapa da metodologia proposta preocupa-se com a seleção do conjunto apropriado de variáveis de entrada do modelo *fuzzy* e com a determinação do conjunto de regressores do modelo, ou seja, o número de amostras atrasadas das variáveis de entrada e saída. Estes dois problemas não são independentes, mas nesta metodologia eles são resolvidos de forma independente, através de dois procedimentos distintos, como descrevem as duas próximas seções.

4.2 Seleção das Variáveis de Entrada

Neste trabalho a técnica proposta por Rallo et al. (2002) é adaptada para selecionar o subconjunto apropriado de variáveis secundárias para compor o conjunto de entrada do modelo *fuzzy* inferencial. O procedimento de seleção das variáveis de entrada funciona como segue.

Seja U o conjunto das p variáveis secundárias e y a variável de saída. Calcula-se o coeficiente de correlação entre cada variável secundária e a variável y . As variáveis secundárias são ordenadas (de forma decrescente) conforme o respectivo valor absoluto do coeficiente de correlação, criando o conjunto S de variáveis secundárias ordenadas. A partir do conjunto S os p subconjuntos de variáveis secundárias são definidos. O conjunto c_1 é composto pela primeira variável em S , aquela que possui o maior valor absoluto de coeficiente de correlação, e a variável y . O conjunto c_2 é composto pelas duas variáveis secundárias que possuem os dois maiores valores

absolutos de coeficiente de correlação e a variável y . A composição de todos os subconjuntos seguem o mesmo esquema até o subconjunto c_p , composto por todas as variáveis secundárias e a variável y . Para cada subconjunto é criado um mapa de Kohonen unidimensional e, após o treinamento de todos os mapas, a qualidade desses mapas é verificada através de uma medida de dissimilaridade entre dois mapas distintos L_q e L_r , conforme as equações (4.2) e (4.3) (Kaski & Lagus, 1996):

$$D(L_q, L_r) = E \left[\frac{\|d_{L_q}(\mathbf{z}) - d_{L_r}(\mathbf{z})\|}{d_{L_q}(\mathbf{z}) + d_{L_r}(\mathbf{z})} \right], \quad (4.2)$$

$$d(\mathbf{z}) = \|\mathbf{z}(t) - \mathbf{w}_{bmu_1}(t)\| + \min_i \sum_{i=p_{bmu_1}}^{p_{bmu_2}-1} \|\mathbf{w}_i(t) - \mathbf{w}_{i+1}(t)\|, \quad (4.3)$$

onde: E é a esperança matemática; bmu é a melhor unidade do mapa p_{bmu_1} é a posição da primeira bmu ; p_{bmu_2} é a posição da segunda bmu ; e para cada par de dados $\mathbf{z}(t)$ é calculada a distância $d(z)$ de $\mathbf{z}(t)$ até a segunda bmu , denotada por $\mathbf{w}_{bmu_2}(t)$, passando inicialmente por $\mathbf{z}(t)$ até a primeira bmu , ou seja, da unidade vencedora, denotada por $\mathbf{w}_{bmu_1}(t)$, e então ao longo do menor caminho entre a $\mathbf{w}_{bmu_1}(t)$ e a $\mathbf{w}_{bmu_2}(t)$.

Segundo Kaski & Lagus (1996), $d(\mathbf{z})$ é uma medida de distância que combina um índice de continuidade do mapeamento do conjunto de dados para a estrutura do mapa com uma medida de precisão na representação dos dados.

Calcula-se, então, a média de dissimilaridade de cada mapa L_q em relação aos outros mapas L_r , como na equação (4.4):

$$Mdis(L_q) = \frac{\sum_{r=1}^{p-1} (D(L_q, L_r))}{p-1}, \text{ tal que } r \neq q. \quad (4.4)$$

O menor valor de $Mdis(L_q)$ indica a maior similaridade do mapa L_q em relação aos outros mapas e maior precisão na representação dos dados, também em relação aos demais mapas. Desta forma, o mapa que apresentar o menor valor de $Mdis$ contém o subconjunto mais apropriado de variáveis secundárias para compor o espaço de entrada do modelo *fuzzy*.

Retornando à equação (4.3), ao considerar um mapa de Kohonen unidimensional, como é o caso deste trabalho, nota-se que existe somente um caminho entre a primeira e a segunda bmu . Então a equação (4.3) pode ser simplificada para a equação (4.5):

$$d(\mathbf{z}) = |\mathbf{z}(t) - \mathbf{w}_{bmu_1}(t)| + \sum_{i=p_{bmu_1}}^{p_{bmu_2}-1} |\mathbf{w}_i(t) - \mathbf{w}_{i+1}(t)|. \quad (4.5)$$

Por exemplo, dado um mapa unidimensional \mathbf{m} composto por 6 unidades ilustrado pela Figura 4.1. Seja $bmu_1 = 2$ a melhor unidade do mapa e $bmu_2 = 6$ a segunda melhor unidade. Então aplicando-se esses valores à equação (4.5) resulta na equação (4.6):

$$d(\mathbf{z}) = |\mathbf{z}(t) - \mathbf{w}_2(t)| + \sum_{i=2}^{6-1} |\mathbf{w}_i(t) - \mathbf{w}_{i+1}(t)|, \quad (4.6)$$

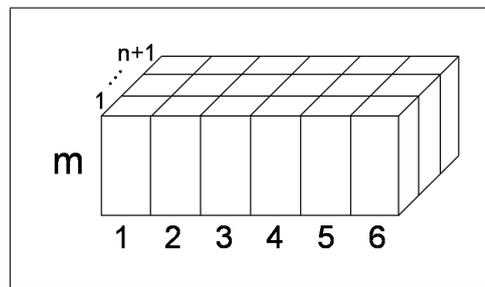


Figura 4.1: Exemplo de um mapa de Kohonen unidimensional.

Em Rallo et al. (2002), considera-se que o relacionamento entre as variáveis secundárias e a variável de saída y é linear, sendo calculado somente o coeficiente de correlação linear, como na equação (4.7) para verificar a relação entre a variável y e cada variável secundária, onde \mathcal{X} é o conjunto de amostras de uma das variáveis secundárias. Neste caso, a relação entre as duas variáveis é representada por uma reta, ou seja, por um modelo linear.

$$r = \frac{n \sum \mathcal{X}Y - (\sum \mathcal{X})(\sum Y)}{\sqrt{n \sum \mathcal{X}^2 - (\sum \mathcal{X})^2} \sqrt{n \sum Y^2 - (\sum Y)^2}}, \quad (4.7)$$

Na equação 4.7 r é o coeficiente de correlação linear, tal que $r \in [-1, 1]$, sendo tanto mais forte quanto mais afastado de zero. Y é a variável cujo valor pretende-se estimar e \mathcal{X} é a variável cujo valor é conhecido e será utilizado para a estimação.

Porém, há situações nas quais as relações entre as variáveis, embora existam, não são adequadamente expressas por um modelo linear (Spiegel, 2004), o que pode significar a existência de uma correlação não-linear. As relações não-lineares podem, às vezes, ser transformadas em lineares, mediante a transformação adequada das variáveis Y ou \mathcal{X} ou de ambas (Spiegel, 2004; Guimarães, 2005). Os modelos não-lineares

mais utilizados, na prática, para a representação da relação entre as variáveis Y e \mathcal{X} são: modelo exponencial (equação (4.8)), modelo potência (equação (4.9)) e modelo logarítmico (equação (4.10)). Neste trabalho consideraram-se as quatro formas de relacionamento e, para cada modelo, é realizada inicialmente a transformação dada pelas respectivas equações e então o coeficiente de correlação é calculado conforme a equação (4.7).

$$\text{Modelo exponencial: } Y = \ln Y. \quad (4.8)$$

$$\text{Modelo potência: } \begin{cases} Y = \ln Y \\ \mathcal{X} = \ln \mathcal{X}. \end{cases} \quad (4.9)$$

$$\text{Modelo logarítmico: } \mathcal{X} = \ln \mathcal{X}. \quad (4.10)$$

Desta forma, calculou-se o coeficiente de correlação para o modelo linear (i), modelo exponencial (ii), modelo potência (iii) e modelo logarítmico (iv), para cada variável secundária e considerou-se o maior valor dentre os coeficientes calculados. A Figura 4.2 apresenta um esquema gráfico para facilitar a compreensão desse procedimento e a Figura 4.3 sumariza o procedimento de seleção de variáveis secundárias.

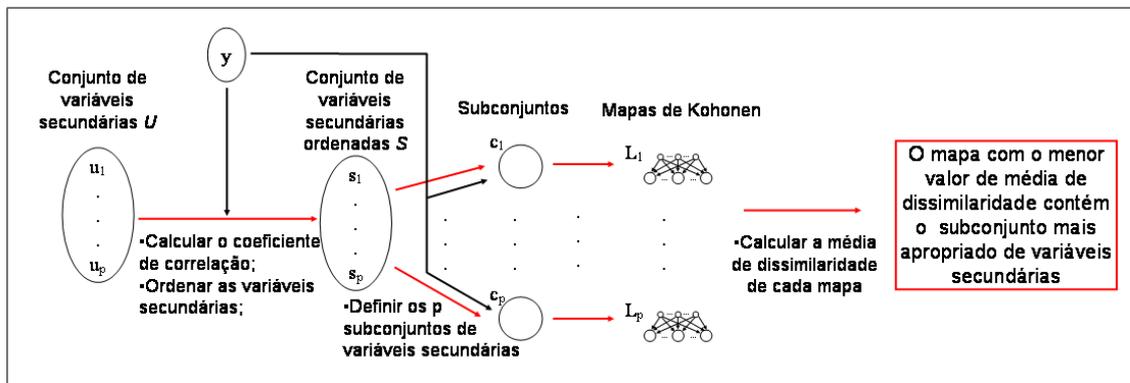


Figura 4.2: Procedimento de seleção de variáveis de entrada.

Como mencionado no início desta seção, o método proposto por Rallo et al. (2002), para a seleção de variáveis de entrada considera somente a correlação linear entre as variáveis secundárias e a variável de saída. Esta abordagem não se mostra apropriada quando existe relacionamentos não-lineares entre as variáveis, e com isso todo o procedimento de seleção pode ser comprometido. O presente trabalho propõe o uso de transformação dessas relações, de não-linear para linear, conforme as equações (4.8), (4.9) e (4.10), a fim de capturar os possíveis relacionamentos não-lineares entre

as variáveis.

Algoritmo 3: Seleção das Variáveis de Entrada.

- **Passo 1:** Seja $U = \{u_1, \dots, u_p\}$ o conjunto de variáveis secundárias e y a variável inferencial de saída. Calcular o coeficiente de correlação entre cada variável secundária e a variável de saída conforme as equações (4.7), (4.8), (4.9) e (4.10), e para cada variável considerar o maior valor dentre as correlações calculadas;
- **Passo 2:** Ordenar as variáveis secundárias de acordo com o valor absoluto do coeficiente de correlação calculado no passo 1, criando o conjunto $S = \{s_1, \dots, s_p\}$, onde $|\text{corr}(s_i, y)| \geq |\text{corr}(s_{i+1}, y)|$, e $|\cdot|$ é o valor absoluto;
- **Passo 3:** Seja p o número de variáveis secundárias. Definir os p subconjuntos de variáveis secundárias (c_1, \dots, c_p) a partir do conjunto S , onde $c_1 = \{s_1, y\}$, $c_2 = \{s_1, s_2, y\}$, \dots , $c_p = \{s_1, \dots, s_p, y\}$;
- **Passo 4:** Definir o número m de unidades dos mapas auto-organizáveis (*SOM's*);
- **Passo 5:** Definir um *SOM* com m unidades para cada subconjunto c_i , e treinar esses *SOMs* com os dados de treinamento;
- **Passo 6:** Para cada par de *SOMs* L_q e L_r , $q \neq r$, calcular a medida de dissimilaridade (equação (4.2));
- **Passo 7:** Calcular a média de dissimilaridade de cada *SOM* L_q com relação a todos os outros *SOMs* como na equação (4.4);
- **Passo 8:** Selecionar o *SOM* com o menor valor de $Mdiss$.

Figura 4.3: O algoritmo de seleção de variáveis.

4.3 Seleção da Ordem do Modelo

O procedimento de seleção da ordem do modelo *fuzzy* realiza a seleção dos regressores. Regressores são as amostras atrasadas das variáveis de entrada e saída que são utilizadas durante a identificação do modelo do processo, neste caso, a identificação do modelo *fuzzy*, como apresenta a Figura 4.4. Esta é uma questão delicada, pois de acordo com Nørgard et al. (2000), uma escolha equivocada do número de amostras atrasadas das variáveis, ou seja, do número de sinais atrasados utilizados como regressores, pode causar um impacto desastroso em algumas aplicações de controle. Um número de atrasos muito pequeno pode não ser capaz de modelar a dinâmica do

sistema. Em contrapartida, um número muito grande pode, no caso do uso da teoria *fuzzy*, deixar o modelo demasiadamente carregado, pois deve-se considerar que cada regressor será uma variável de entrada no modelo *fuzzy*, e cada variável é descrita por um determinado número de conjuntos *fuzzy*. Como o tamanho da base de regras é regido pelo número de variáveis do modelo *fuzzy* e também pelos respectivos conjuntos *fuzzy*, quanto maior o número de variáveis e conjuntos *fuzzy*, maior a base de regras.

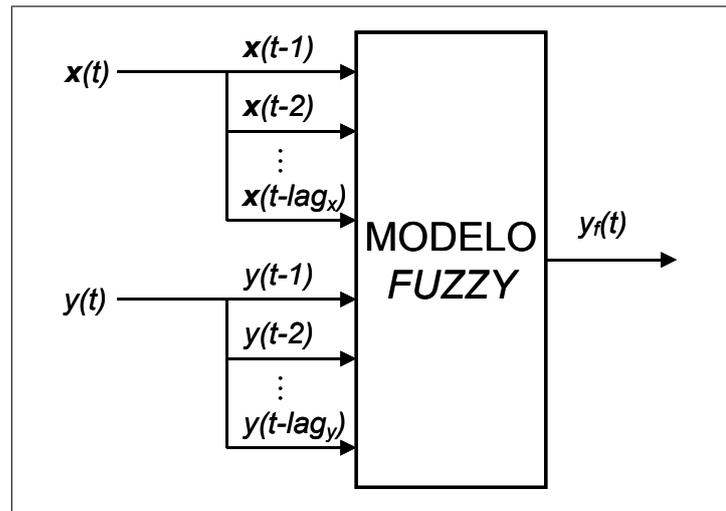


Figura 4.4: Conjunto de regressores.

Em He & Asada (1993) é proposto um método para a seleção da estrutura de uma rede neural. Este método é baseado na suposição de que o processo pode ser representado apropriadamente por uma função que é razoavelmente suave nos regressores (Nørgard et al., 2000). Seja $\{\mathbf{x}(t), y(t)\}$, $t = 1, \dots, N_d$, onde $\mathbf{x}(t)$ é o vetor contendo todas as n variáveis de entrada (selecionadas na etapa anterior) e $y(t)$ é a variável de saída. O conjunto φ de regressores de y_f é composto pelas amostras atrasadas das variáveis de entrada e saída do processo como denota a equação (4.11).

$$\varphi(t) = \begin{bmatrix} y(t-1) \\ \vdots \\ y(t-lag_y) \\ \mathbf{x}(t-d-1) \\ \vdots \\ \mathbf{x}(t-d-lag_x) \end{bmatrix} \quad (4.11)$$

onde lag_y é o número de amostras atrasadas da variável de saída; d é o atraso da entrada; lag_x é o número de amostras atrasadas da entrada; $\varphi(t)$ é o conjunto de regressores para estimar $y_f(t)$.

Utilizando a notação empregada em identificação de sistemas (Ljung, 2000), a saída $y(t)$ estimada pelo modelo *fuzzy* inferencial, denotada por $y_f(t)$, é dada pela equação (4.12), onde g_0 representa o modelo *fuzzy*, φ é o conjunto de regressores e θ é o conjunto de parâmetros do modelo *fuzzy*.

$$y_f(t) = g_0(\varphi(t), \theta). \quad (4.12)$$

Conforme a equação (4.11) e a equação (4.12), o objetivo é determinar o número de amostras atrasadas das variáveis de entrada e saída, lag_x e lag_y , respectivamente. Este método utiliza os quocientes de Lipschitz (He & Asada, 1993), calculados a partir do conjunto de dados amostrados do processo. Para todas as combinações de pares de entrada-saída, o quociente de Lipschitz é calculado como na equação (4.13):

$$q_{i,j} = \frac{\|y(t_i) - y(t_j)\|}{\|\varphi(t_i) - \varphi(t_j)\|}, \quad i \neq j, \quad (4.13)$$

onde $\|\cdot\|$ especifica norma Euclidiana e t_i é relativo à t_i -ésima amostra do conjunto de dados.

Para compreender como o atraso é selecionado a partir desses quocientes de Lipschitz, dois casos devem ser considerados. O primeiro caso é quando se seleciona menos regressores do que o necessário. Esta ausência de um regressor pode levar a quocientes com valores elevados. Além disso, quanto maior o número de regressores faltantes, maior é o valor de q (Nørgard et al., 2000). Por outro lado, a inclusão de muitos sinais passados no conjunto de regressores causa a redundância de informação. O impacto dessa inserção de regressores supérfluos no valor dos quocientes de Lipschitz não é muito significativo, o que em geral implica apenas em uma redução ínfima nos valores dos quocientes. Contudo, deve-se levar em conta que o aumento no conjunto de regressores implica no aumento da base de regras *fuzzy* e, consequentemente, na complexidade do modelo *fuzzy*. O procedimento completo para a seleção da ordem do modelo *fuzzy* é apresentado na Figura 4.5 (He & Asada, 1993).

Esta fase de seleção da estrutura do modelo apresenta-se como a segunda contribuição deste trabalho, onde foi verificado e aplicado de forma bem sucedida, a possibilidade do uso do algoritmo proposto por Nørgard et al. (2000) para a seleção da estrutura de um modelo *fuzzy*, em especial modelos do tipo *FARMA - Fuzzy Auto-Regressive Moving Average* (Park et al., 1995).

Algoritmo 4: Seleção da Ordem do Modelo *Fuzzy*.

- **Passo 1:** Para uma dada escolha de a , $a = lag_x = lag_y$, determinar os quocientes de Lipschitz para todas as combinações de pares de entrada-saída;
- **Passo 2:** Selecionar os r maiores quocientes, $r = 0.01N_d \approx 0.02N_d$. Os maiores quocientes ocorrem, em geral, quando as diferenças $\varphi(t_i) - \varphi(t_j)$ são pequenas.
- **Passo 3:** Avaliar o seguinte critério:

$$\bar{q}^{(a)} = \left(\prod_{t=1}^r \sqrt{r} q^{(a)}(t) \right)^{\frac{1}{r}}, \quad (4.14)$$

onde $q^{(a)}$ é o quociente referente ao atraso a e $\bar{q}^{(a)}$ é o valor do critério para o atraso a .

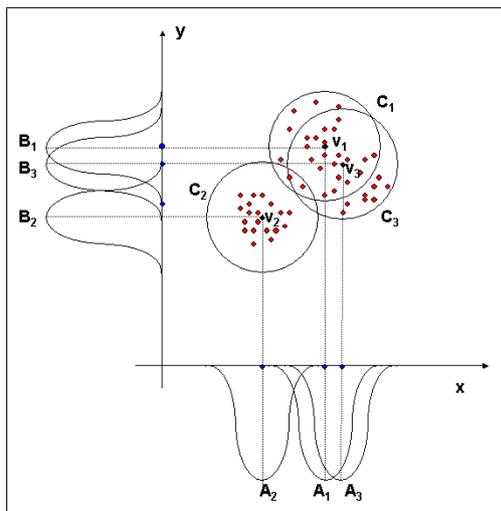
- **Passo 4:** Repetir os cálculos para um determinado número de atrasos distintos;
- **Passo 5:** Plotar o critério como uma função do atraso e selecionar o número ótimo de regressores como o cotovelo (“*knee-point*”) da curva. Este ponto é o selecionado porque a partir dele a diferença entre os valores de $\bar{q}^{(a)}$ não apresentam mais uma diminuição significativa em seus valores.

Figura 4.5: O algoritmo de seleção da ordem do modelo fuzzy.

4.4 Geração da Base de Regras Inicial

Nesta fase, um algoritmo de agrupamento *fuzzy FCM* (2.7.4) é aplicado ao espaço de entrada e saída, onde cada *cluster* corresponde a uma regra. Portanto, o objetivo é particionar este espaço em ncr *clusters* e gerar, a partir desses *clusters*, as funções de pertinência que descrevem o universo de discurso de cada variável, assim como a base de regras do modelo *fuzzy*. Um valor suficientemente grande para o tamanho da base de regras inicial (ncr) é empregado como uma tentativa de cobrir todas as regiões importantes do espaço de entrada e saída, tornando o resultado do agrupamento menos dependente da partição inicial do domínio das variáveis (Setnes, 2000).

De forma ilustrativa, a Figura 4.6 e a Figura 4.7 mostram como é realizada a obtenção da base de regras através do uso do algoritmo *FCM*. Trata-se de um exemplo com uma variável de entrada e uma variável de saída. O espaço de entrada e saída é particionado em três *clusters*, como ilustra a Figura 4.6, e estes *clusters* são projetados em cada universo de discurso de forma a gerar as funções de pertinência das variáveis do modelo. A base de regras resultante é descrita pela Figura 4.7.



C_1 : Se x é A_1 Então y é B_1 C_2 : Se x é A_2 Então y é B_2 C_3 : Se x é A_3 Então y é B_3
--

Figura 4.7: Base de regras resultante da projeção.

Figura 4.6: Projeção dos clusters.

Conforme a seção 2.7.4, o objetivo do algoritmo *FCM* é encontrar H , a matriz de partição ($ncr \times N_d$) que contém o grau de pertinência de cada par de dados a cada *cluster* considerado. Durante esse procedimento de agrupamento, é também calculada a matriz $V_{ncr \times (n+1)}$, onde $n + 1$ é o número total de variáveis do modelo *fuzzy*, contendo o centro dos *clusters*. Esses centros são calculados de acordo com a distribuição dos dados no espaço de entrada e saída.

Lembrando que cada *cluster* corresponde a uma regra, as funções de pertinência referentes aos conjuntos *fuzzy* dessas regras são obtidas a partir da matriz de partição H . Pode-se dizer que a matriz H contém os valores discretos das ncr funções de pertinência correspondentes aos ncr *clusters*. Porém, essas funções de pertinência estão num espaço N_d -dimensional. Os conjuntos *fuzzy* bidimensionais correspondentes a cada *cluster*/regra são obtidos a partir dos valores discretos da função de pertinência N_d -dimensional contida na i -ésima linha da matriz H , através de sua projeção *fuzzy* no domínio de cada variável.

Detalhando o procedimento, seja $H = [h_{ik}]_{ncr \times N_d}$ ($H \in \mathfrak{R}^{ncr \times N_d}$) a matriz de partição gerada pelo algoritmo *FCM* (com na seção 2.7.4), onde $h_{ik} \in H$ corresponde à pertinência da k -ésima amostra de dados ao i -ésimo *cluster*, $k = 1, \dots, N_d$, $i = 1, \dots, ncr$, N_d é o tamanho do conjunto de dados amostrados e ncr é o número de *clusters*. Seja $Z^{N_d} = [z_{kj}]_{N_d \times (n+1)}$ ($Z^{N_d} \in \mathfrak{R}^{N_d \times (n+1)}$) a matriz do conjunto de dados amostrados, onde $z_{kj} \in Z^{N_d}$ corresponde à k -ésima amostra da j -ésima variável, $j = 1, \dots, n+1$, e n é número de variáveis de entrada. A projeção *fuzzy* é efetuada como na equação (4.15):

$$proj_{ij}(H_i, Z_j^{N_d}) = \max_k \left(\text{prod}(H_i, Z_j^{N_d}) \right), \quad (4.15)$$

onde H_i é o vetor (linha) contendo a j -ésima linha de H e corresponde aos graus de pertinência das N_d amostras ao i -ésimo *cluster*; $Z_j^{N_d}$ é o vetor (coluna) contendo a j -ésima variável do conjunto de dados; **prod** é a t-norma produto e $proj_{ij}$ é o resultado da projeção do i -ésimo *cluster* no domínio da j -ésima variável;

Este procedimento de projeção gera os pontos discretos das funções de pertinência em cada domínio. As respectivas funções de pertinência paramétricas podem ser obtidas através de uma de interpolação, onde as funções de pertinência são obtidas conforme a curva descrita pelos pontos. Existem várias opções de funções que podem ser utilizadas para representar os conjuntos *fuzzy*.

Neste trabalho foi selecionada a função de pertinência *gbell* (*generalized bell*), dada pela equação (4.16), devido à sua notação concisa e à suavidade de sua curva. Segundo Jang et al. (1997), a vantagem do uso de funções *gbell* sobre as funções gaussianas está no parâmetro adicional σ , que fornece um grau adicional de liberdade para ajustar os pontos de cruzamento entre as funções de pertinência que descrevem um universo de discurso. Além disso, esse parâmetro permite a representação de regiões de platô, ou seja, regiões onde mais de um elemento possui grau de pertinência

unitário.

$$f(x; \sigma, \gamma, \omega) = \left[1 + \left| \frac{x - \omega}{\sigma} \right|^{2\gamma} \right]^{-1}, \quad (4.16)$$

onde o parâmetro ω é o centro da função *gbell*, $(\omega - \sigma)$ e $(\omega + \sigma)$ são os pontos de cruzamento (*crossover points*), cujo grau de pertinência é igual a 0.5 e o parâmetro γ controla a inclinação nesses pontos de cruzamento.

A Figura 4.8 apresenta um exemplo de função *gbell*, onde podem ser observados os parâmetros que determinam o perfil da função.

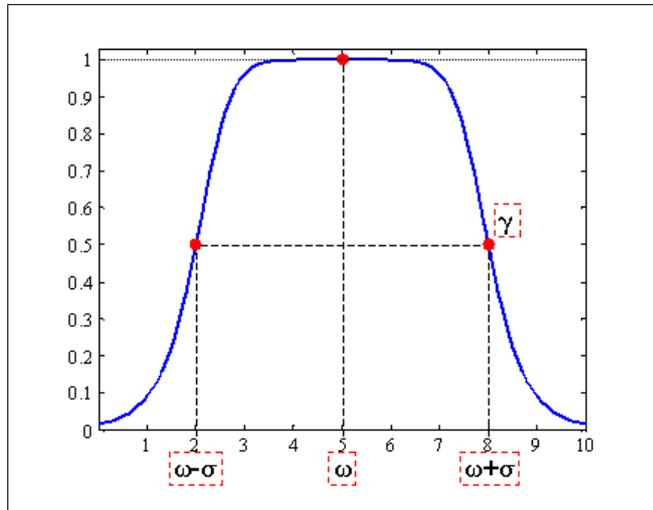


Figura 4.8: Função de pertinência do tipo *gbell*.

O vetor $proj_{ij}$, calculado com a equação (4.15), contém os valores discretos da i -ésima função de pertinência da j -ésima variável do modelo e, a partir dos dados contidos em $proj_{ij}$, são calculados os parâmetros ω_{ij} , σ_{ij} e γ_{ij} da função *gbell* que melhor represente o i -ésimo conjunto *fuzzy* da j -ésima variável do modelo *fuzzy*. Levando-se em consideração a equação (4.16) e a Figura 4.8, os parâmetros são especificados da seguinte forma. O primeiro parâmetro a ser especificado é o ω_{ij} , como sendo o elemento com o maior grau de pertinência em $proj_{ij}$, como na equação (4.17). Para encontrar o parâmetro σ_{ij} , é necessário verificar em $proj_{ij}$ o maior elemento σ_{ij1} , e o menor elemento σ_{ij2} com grau de pertinência igual a 0.5. A partir desses dois valores e do valor de ω_{ij} calcula-se σ_{ij} como na equação (4.18). Finalizando, o parâmetro γ_{ij} é calculado a partir de σ_{ij} , como na equação (4.19).

$$\omega_{ij} = \mathbf{z}(\max(\text{proj}_{ij}), j), \quad (4.17)$$

$$\begin{aligned} \sigma_{ij1} &= \mathbf{z}(\min(\text{proj}_{ij} \text{ tal que } \text{proj}_{ij} = 0.5), j), \\ \sigma_{ij2} &= \mathbf{z}(\max(\text{proj}_{ij} \text{ tal que } \text{proj}_{ij} = 0.5), j), \\ \sigma_{ij} &= \max(|\omega - \sigma_{ij1}|, |\sigma_{ij2} - \omega_{ij}|), \end{aligned} \quad (4.18)$$

$$\gamma_{ij} = 20 * \sigma_{ij}, \quad (4.19)$$

onde $\mathbf{z}(\cdot)$ é um par de dados de entrada e saída pertencente ao conjunto Z^{N_d} , equação (4.1), e $\mathbf{z}(t, j)$ é a j -ésima componente do vetor \mathbf{z} , ou seja, a j -ésima variável.

A Figura 4.9 apresenta os passos do procedimento de geração da base de regras inicial.

Algoritmo 5: Geração da Base de Regras Inicial.

- **Passo 1:** Estipular o número de *clusters ncr* (tamanho da base de regras inicial);
- **Passo 2:** Aplicar o algoritmo *FCM* no espaço de entrada e saída de modo a obter a localização dos *clusters* nesse espaço multidimensional;
- **Passo 3:** Projetar os *clusters* no subespaço de cada variável com a projeção *fuzzy* da matriz de partição *H* obtida com o algoritmo *FCM*;
- **Passo 4:** Calcular os parâmetros das *ncr* funções de pertinência gaussianas que descrevem o domínio de cada variável como nas equações(4.17), (4.18) e (4.19).

Figura 4.9: O algoritmo da geração da base de regras inicial.

Esta é a terceira contribuição deste trabalho, onde é proposta uma metodologia para a construção da base de regras do modelo *fuzzy* a partir do agrupamento de dados utilizando-se o algoritmo *FCM*. Em Nagai (2002), os conjuntos *fuzzy* eram representados por funções de pertinência gaussianas. Porém, após a fase de simplificação (detalhado na seção 4.5), as novas funções de pertinência não representavam adequadamente as funções substituídas. Esta falha deve-se ao fato das funções gaussianas não serem capazes de representar regiões de platô. A partir desta constatação optou-se por utilizar funções de pertinência do tipo *gbell*, cujo parâmetro adicional σ possibilita a apropriada representação dessas regiões de platô.

4.5 Simplificação da Base de Regras Inicial

A interpretabilidade lingüística de um modelo *fuzzy* é um dos aspectos que distinguem a modelagem *fuzzy* de métodos de modelagem tipo caixa-preta, como as redes neurais, por exemplo. Tal aspecto permite a interpretação e análise do modelo obtido (Babůska et al., 1998). Porém essa interpretabilidade não é alcançada automaticamente. Quando um modelo *fuzzy* é desenvolvido utilizando-se de conhecimento especialista, é possível controlar o número de regras e de funções de pertinência geradas. No entanto, torna-se difícil controlar a redundância e a complexidade desnecessária quando a identificação do modelo é realizada de forma automática, a partir de pares amostrados de dados de entrada e saída, como é o caso desta proposta.

Nesta metodologia, o número de funções de pertinência de cada variável não é pré-determinado, sendo possível a existência de funções de pertinência similares. Funções de pertinência similares são funções altamente sobrepostas, que descrevem quase a mesma região no domínio da variável a qual esses conjuntos pertencem (como os conjuntos A_1 e A_3 na Figura 4.6), fazendo com que o modelo utilize mais conjuntos *fuzzy* do que o necessário (Nagai, 2002).

Neste trabalho a similaridade entre conjuntos *fuzzy* é medida através da proximidade entre os centros das funções *gbell* que os representam. Para verificar a existência de conjuntos *fuzzy* similares, os parâmetros das funções de pertinência são verificadas da seguinte forma. Considerando-se o domínio de cada variável j , duas funções *gbell*, cujos centros são ω_{fj} e ω_{gj} , são consideradas similares se:

$$|\omega_{fj} - \omega_{gj}| \leq \zeta_\omega, \quad (4.20)$$

onde $|\cdot|$ é o valor absoluto e ζ_ω é a diferença mínima entre os centros e deve ser definida *a priori*.

Caso a inequação (4.20) seja satisfeita, então as respectivas funções de pertinência são substituídas por uma nova função de pertinência *gbell* cujos parâmetros são calculados como nas equações (4.21), (4.22) e (4.23) (considerando-se o domínio da j -ésima variável).

$$\omega_{newj} = (\omega_{fj} + \omega_{gj}) / 2, \quad (4.21)$$

$$\sigma_{newj} = \max(\sigma_{fj} + |\omega_{newj} - \omega_{fj}|, \sigma_{gj} + |\omega_{gj} - \omega_{newj}|), \quad (4.22)$$

$$\sigma_{newj} = 2 * \sigma_{newj}. \quad (4.23)$$

Através da Figura 4.10 é possível visualizar como os parâmetros da nova função de pertinência são calculados.

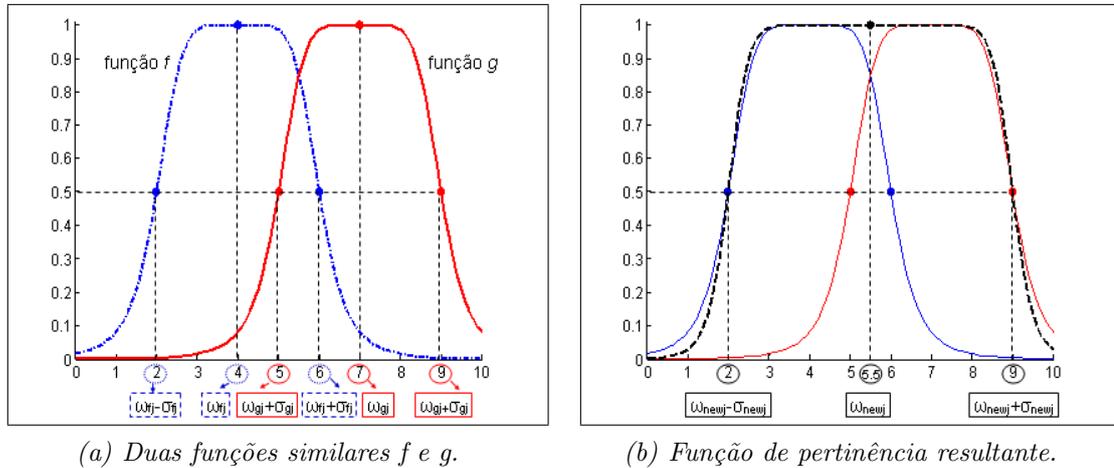


Figura 4.10: Exemplo de substituição de funções de pertinência similares.

Após este procedimento, as regras são atualizadas com as novas funções de pertinência. A Figura 4.11 ilustra um exemplo de simplificação, onde as funções de pertinência A_1 e A_3 são substituídas pela função $A_{1,3}$ (no domínio de x) e as funções B_1 e B_3 são substituídas pela função $B_{1,3}$ (no domínio de y), resultando na diminuição tanto do número de funções de pertinência que descrevem cada domínio quanto do número de regras que compõem a base. A Figura 4.12 sumariza o procedimento de simplificação da base de regras inicial.

Este procedimento de simplificação agrupa as funções de pertinência que descrevem o mesmo local no domínio de uma determinada variável. A diminuição do tamanho da base de regras é apenas uma consequência dessa simplificação. A otimização da estrutura realizada nesta fase desempenha um importante papel na identificação de modelos *fuzzy*, pois ela aumenta a interpretabilidade do modelo, uma das características essenciais dos sistemas *fuzzy* (Guillaume, 2001).

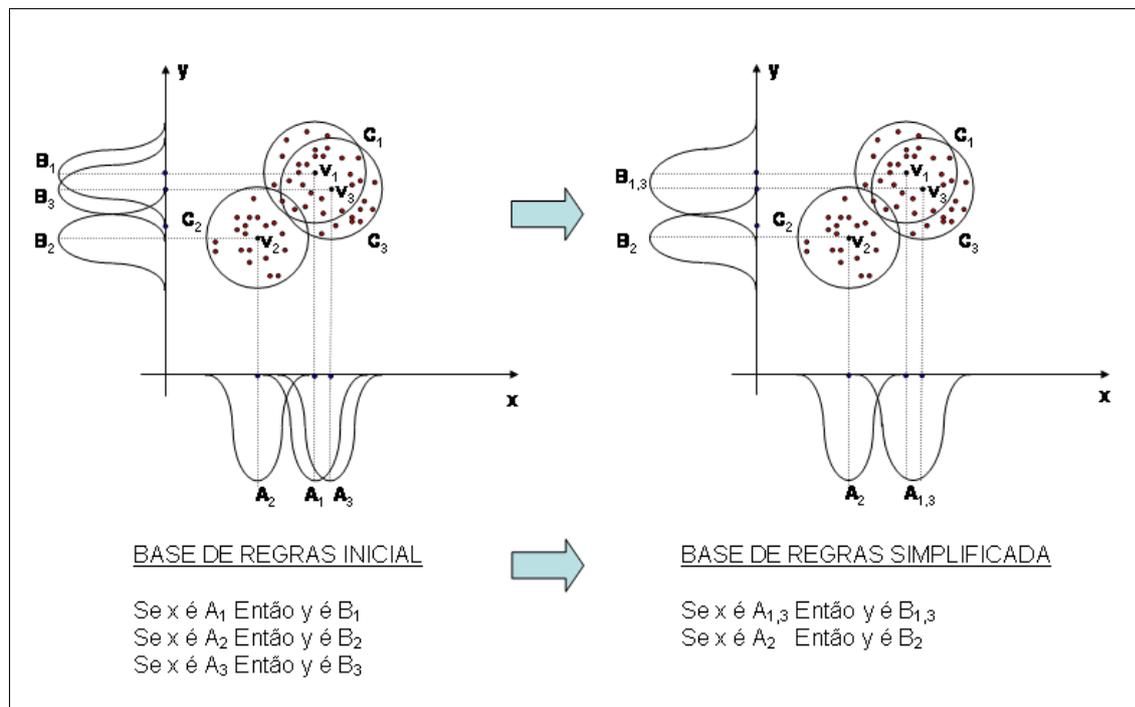


Figura 4.11: Simplificação da base de regras inicial.

Algoritmo 6: Simplificação da Base de Regras Inicial.

Para cada variável do modelo:

- **Passo 1:** Verificar a existência de funções de pertinência similares através da inequação (4.20);
- **Passo 2:** Caso existam funções de pertinência similares, estas funções são substituídas por uma nova função cujos parâmetros são calculados através das equações (4.21), (4.22) e (4.23) (considerando-se o domínio da j -ésima variável);
- **Passo 3:** Atualizar a base de regras, substituindo as funções de pertinência similares pela nova função de pertinência calculada no passo anterior;
- **Passo 4:** Eliminar as possíveis regras idênticas que, porventura, surgiram em consequência do agrupamento de funções de pertinência similares.

Figura 4.12: O algoritmo da geração da base de regras inicial.

4.6 Tratamento de Regras Inconsistentes

Considerando-se que o número de regras geradas pelo algoritmo de agrupamento é muito grande, é provável que existam regras inconsistentes. Além disso, a constante de ponderação que determina a sobreposição dos *clusters* e a união de conjuntos *fuzzy* similares também podem causar inconsistência em regras.

Para resolver esse conflito, é utilizada a medida do fator de confiança CF de uma regra (Quinlan, 1987). Seja $A \rightarrow C$ uma regra, onde A é o antecedente (uma conjunção de condições) e C é o conseqüente da regra. A medida CF é denotada pela equação (4.24) (Fertig et al., 1999):

$$CF(R_i) = \frac{|(A \wedge C)|}{|A|}, \quad (4.24)$$

onde $|\cdot|$ denota cardinalidade. A cardinalidade de um conjunto *fuzzy* $A(x)$, definido sobre um universo finito U_A , é a soma dos graus de pertinência de todos os elementos de U_A em A (equação (4.25)) (Setnes, 1999).

$$|A| = \sum_x \mu_A(x) \quad (4.25)$$

Em um grupo de regras inconsistentes, a regra com o maior valor de CF é mantida, e as demais são removidas da base de regras.

4.7 Metodologia de Avaliação dos Resultados

Nos capítulos 5 e 6 serão apresentados os resultados da aplicação, onde a metodologia proposta é utilizada para a identificação de modelos *fuzzy* inferenciais para processos simulados da indústria petroquímica.

Para realizar a identificação dos modelos o conjunto de dados amostrados é pré-processado de modo a obter dados com média zero e desvio padrão unitário. Além disso, o universo de discurso de cada uma das variáveis do processo é normalizado no intervalo de $[0, 1]$. Esse conjunto de dados amostrados normalizado é dividido em três conjuntos distintos:

- (i) Conjunto de dados de treinamento: é o conjunto de dados usado para a identificação (geração) do modelo *fuzzy* inferencial.

- (ii) Conjunto de dados de validação: é o conjunto de dados usado durante a fase de identificação para medir o grau de “aderência” excessiva do modelo aos dados de treinamento. Isto ocorre quando o erro medido sobre o conjunto de validação aumenta enquanto o erro sobre o conjunto de treinamento diminui, indicando uma situação de *overfitting*, ou seja, o modelo inferencial *fuzzy* apresenta baixa capacidade de generalização ao apresentar valores altos de erro para o conjunto de validação. Desta forma, deve-se buscar o menor erro possível para o conjunto de dados de validação.
- (iii) Conjunto de dados de teste: é o conjunto de dados usado após a identificação para uma avaliação quantitativa e efetiva da qualidade do modelo *fuzzy* inferencial.

Para todos os casos, duas métricas são calculadas a fim de avaliar o índice de desempenho dos modelos inferenciais *fuzzy* obtidos. Essas duas métricas são definidas pelas equações (4.26) e (4.27).

$$mse = \frac{1}{N} \sum_{t=1}^N (y(t) - y_f(t))^2; \quad (4.26)$$

$$mape = \frac{1}{N} \sum_{t=1}^N \left| \frac{y(t) - y_f(t)}{y(t)} \right| \cdot 100\%; \quad (4.27)$$

onde: y é a saída medida do sistema; y_f é a saída obtida pelo modelo *fuzzy*; mse (*mean-square error*) é a média do erro quadrático ocorrido entre y e y_f ; $mape$ (*mean absolute percentage error*) é a média do erro percentual absoluto (em percentagem) entre y e y_f a cada instante t ; e N é o número de amostras.

Capítulo 5

Coluna de Destilação Binária

5.1 Descrição da Coluna de Destilação Binária

Numa primeira aplicação, a metodologia proposta foi empregada na construção de um *soft sensor* para inferir a composição dos produtos de topo e fundo de uma coluna de destilação binária (Figura 5.1 ¹) durante a fase de inicialização da coluna. Os dados amostrados foram obtidos de Fabro et al. (2005), onde foi modelada uma coluna de destilação binária (metanol e água) com o auxílio da ferramenta computacional HYSYS[®] (HYSYS, 1989), numa aplicação semelhante à apresentada em Neves-Jr & Aguilar-Martin (1999).

A coluna possui 20 pratos, com as seguintes características de funcionamento em regime estacionário (Fabro et al., 2005):

- Composição da alimentação: 50% de água e 50% de metanol;
- Temperatura de alimentação: 75, 9°C;
- Vazão molar de alimentação: 236 kmol/h, inserido na coluna na altura do prato 14;
- Composição do produto de topo (destilado): ao menos 99,9% de metanol em regime;
- Vazão do produto de topo (destilado): 85 kmol/h
- Composição do produto de fundo (resíduo): ao menos 99,9% de água;
- Vazão do produto de fundo (resíduo): 151 kmol/h;
- Refluxo do condensador para a coluna: 247 kmol/h;

¹Figura obtida de (Neves-Jr & Aguilar-Martin, 1999).

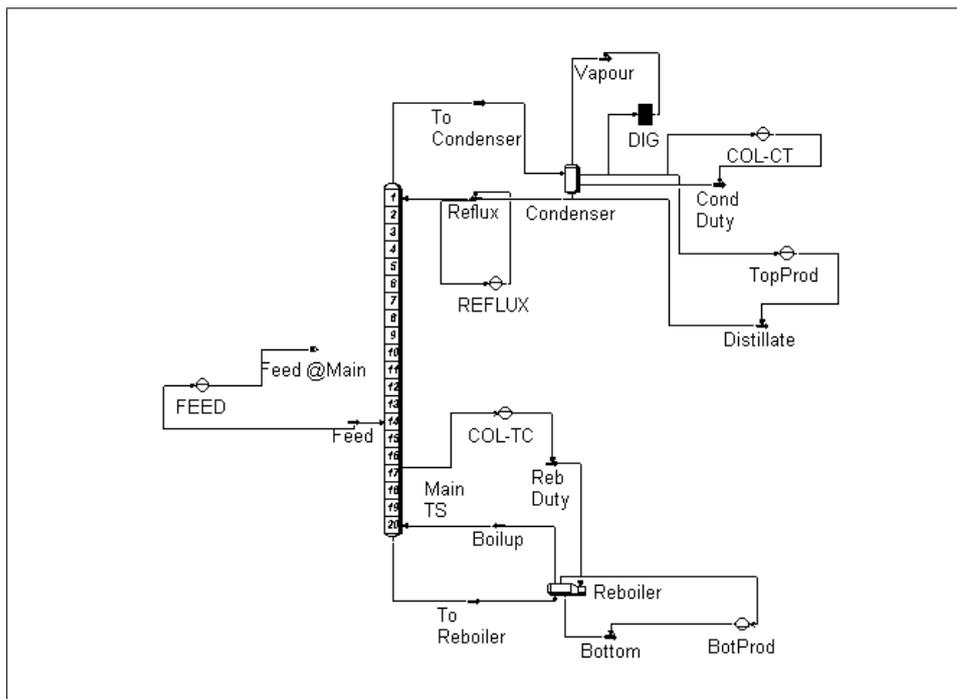


Figura 5.1: Coluna de destilação binária.

- Pressão no topo da coluna: $101,3\text{kPa}$;
- Perfil de temperatura (temperatura medida nos pratos):
 - Prato 9: $67,54^{\circ}\text{C}$;
 - Prato 13: $86,41^{\circ}\text{C}$;
 - Prato 16: $101,3^{\circ}\text{C}$;
 - Prato 18: $102,3^{\circ}\text{C}$;

O perfil de temperatura da coluna é determinado por quatro sensores posicionados nos pratos (dois acima do ponto de alimentação, nos pratos 9 e 13, e dois abaixo, nos pratos 16 e 18). Quando as temperaturas lidas nos pratos aproximam-se do perfil definido no projeto da coluna para o funcionamento em regime estacionário, o processo de inicialização chega ao final (Fabro et al., 2005). Nesta situação, a composição de destilado (metanol) retirado no topo da coluna apresenta composição maior que 99,9%.

Apesar das informações de composição estarem disponíveis diretamente no simulador *HYSYS*, isto nem sempre é possível em ambientes industriais, onde é comum o uso de testes laboratoriais para a determinação da composição do produto (Qin, 1996). O conhecimento sobre a medida das composições dos produtos é necessário durante o procedimento de inicialização, pois elas demarcam o momento onde

é possível iniciar a retirada de fluidos da coluna. De acordo com Fabro (2003), é importante obter estimativas das composições para não correr o risco de retirar da coluna um produto impuro (com mais de 2% de impurezas), ou iniciar a retirada do resíduo enquanto este ainda possui uma quantidade de metanol diluída maior que 2%.

Em Fabro (2003) foram desenvolvidos estimadores neurais (ou *soft sensors* neurais) para inferir a composição do produto de topo (metanol - y_1) e do produto de fundo (água - y_2). Neste trabalho a metodologia proposta é utilizada na construção de estimadores baseados em modelos *fuzzy* para inferir a composição dos produtos de topo e fundo durante o processo de inicialização dessa mesma coluna.

Segundo Fabro (2003), no início do procedimento de inicialização, enquanto a coluna está sendo aquecida, o conhecimento desta composição não é necessário. Com o avanço do processo de inicialização, as estimativas dessas composições tornam-se importantes, principalmente quando se aproximam do nível de pureza desejado para o início da retirada dos produtos da coluna.

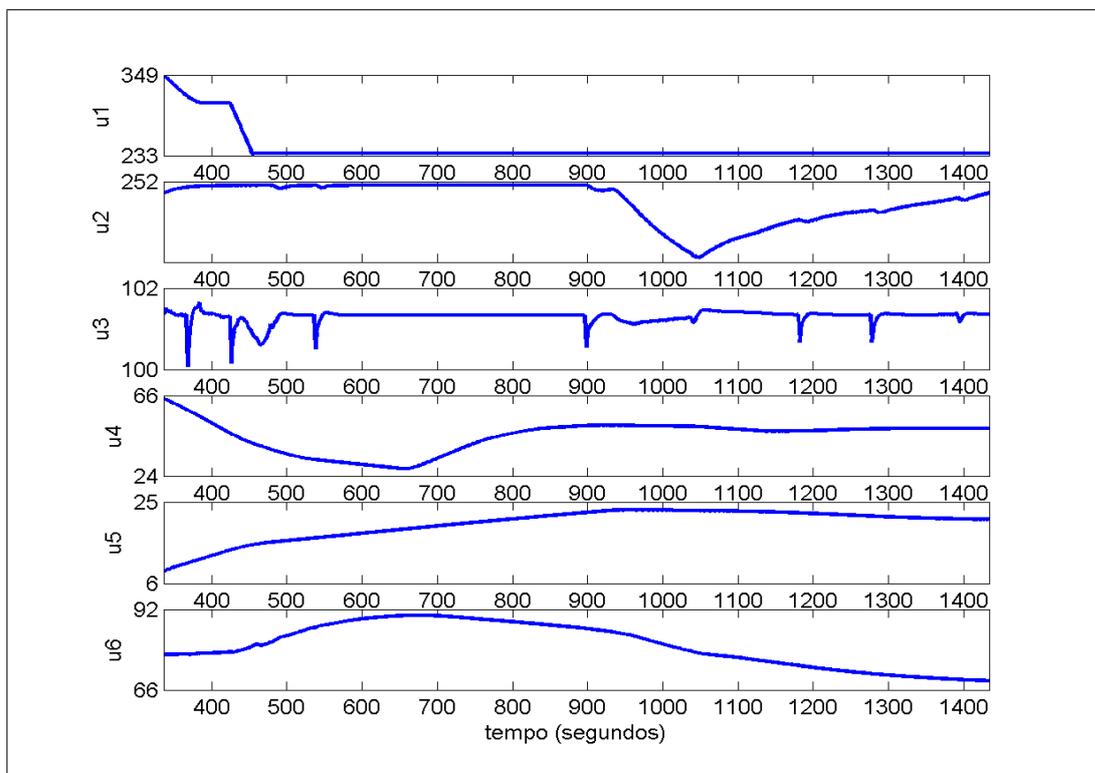
A partir dos dados obtidos no simulador, foram selecionadas as faixas de dados de interesse e realizadas as identificações dos modelos inferenciais. Para a identificação do modelo de inferência da composição do produto de fundo, foi selecionada a faixa de variação entre 80% e 99% de pureza da água, pois a obtenção de um nível de 98% nesta composição indica o final de um estágio e início de outro procedimento de inicialização (Fabro, 2003). Da mesma forma, para a composição de metanol no produto de topo, a faixa selecionada foi entre 90% e 99%.

As variáveis secundárias disponíveis, do ponto de vista de interface, são apresentadas na Tabela 5.1. Estas variáveis são consideradas para a identificação dos dois modelos inferenciais: o modelo para inferir a composição do produto de topo (metanol), e o modelo para inferir a composição do produto de fundo (água).

As Figuras 5.2 e 5.3 apresentam o perfil das variáveis secundárias descritas pela Tabela 5.1. Dadas essas variáveis secundárias, os perfis das variáveis de saída y_1 (composição do produto de topo - metanol) e y_2 (composição do produto de fundo - água) são ilustrados pela Figura 5.4.

Tabela 5.1: Descrição das variáveis secundárias da coluna binária.

#	Variável	Descrição
u_1	$feed1_{pv}$	vazão molar de alimentação
u_2	$reflux_{pv}$	refluxo do condensador para a coluna
u_3	pv_{press}	pressão no topo da coluna
u_4	$bottom_{pv}$	nível líquido do refeedor
u_5	top_{pv}	nível líquido no condensador
u_6	$tray9$	temperatura no prato 9
u_7	$tray13$	temperatura no prato 13
u_8	$tray16$	temperatura no prato 16
u_9	$tray18$	temperatura no prato 18
u_{10}	$cond_{flow}$	vazão para o condensador
u_{11}	top_{flow}	vazão do produto de topo (destilado)
u_{12}	$bottom_{flow}$	vazão do produto de fundo (resíduo)

Figura 5.2: Variáveis de entrada do modelo da coluna binária (Parte I) $\{ u_1, u_2, u_3, u_4, u_5, u_6 \}$.

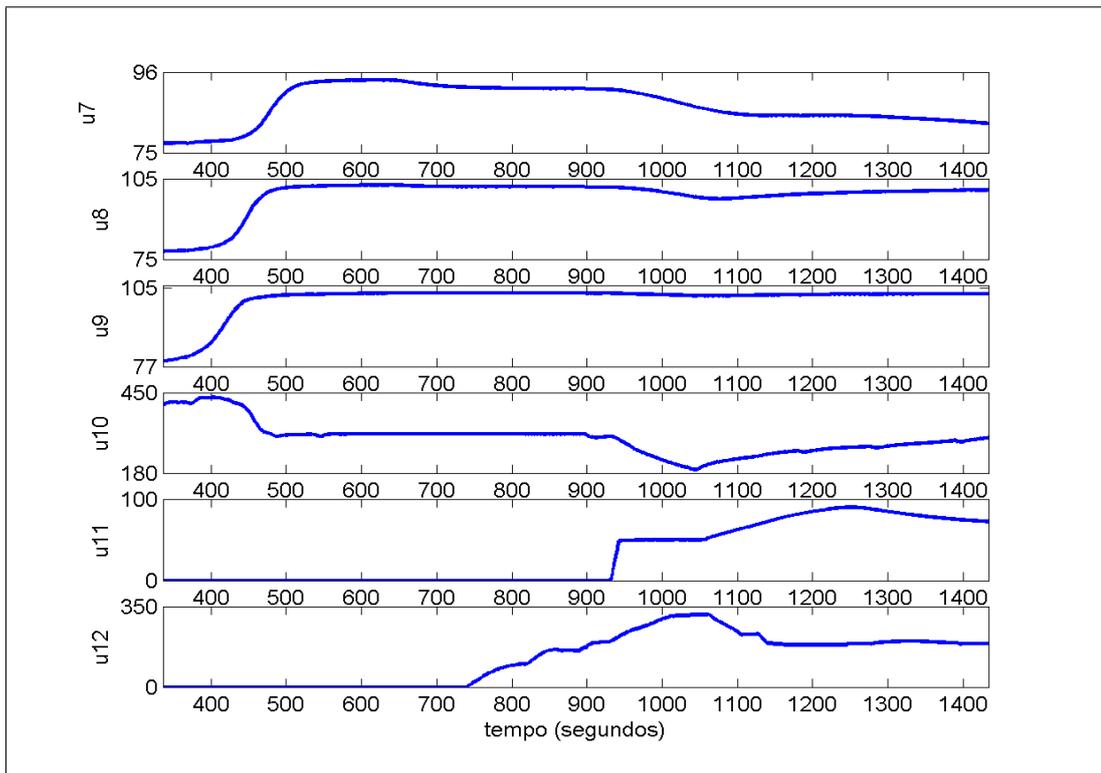


Figura 5.3: Variáveis de entrada do modelo da coluna binária (Parte II) $\{ u_7, u_8, u_9, u_{10}, u_{11}, u_{12} \}$.

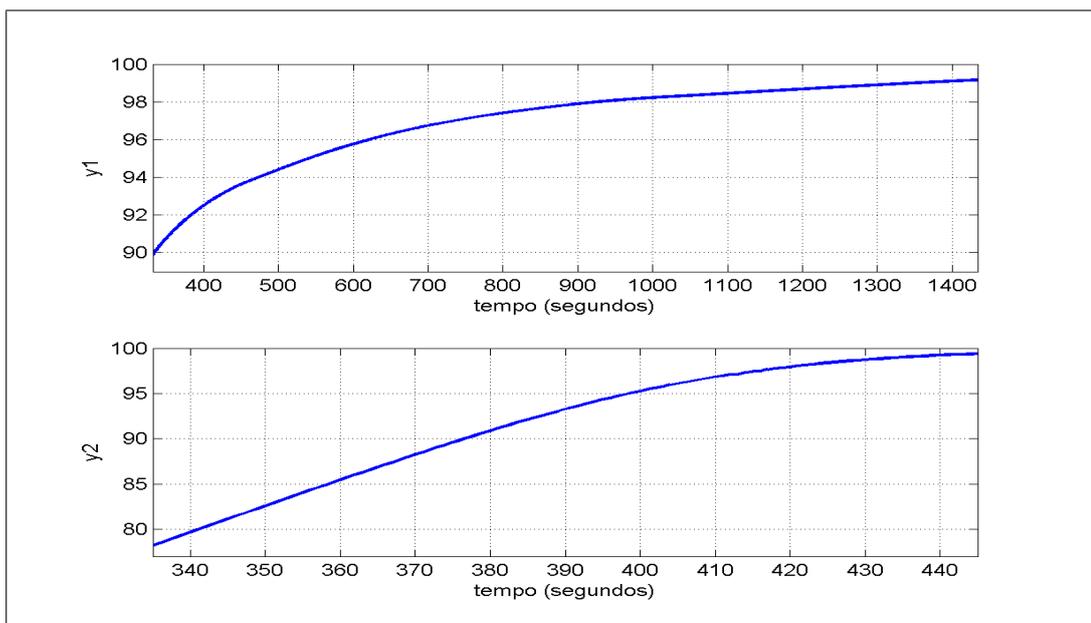


Figura 5.4: Variáveis de saída da coluna binária: composição do produto de topo (y_1) e de fundo (y_2).

5.2 Inferência da Composição do Produto de Topo da Coluna Binária

Esta seção apresenta os resultados obtidos durante o procedimento de identificação de um modelo *fuzzy* inferencial para estimar a composição do produto de topo (y_1).

De acordo com o algoritmo de identificação proposto, o primeiro passo é selecionar as variáveis de entrada, considerando-se todas as variáveis secundárias disponíveis (Tabela 5.1). Segundo o algoritmo de seleção das variáveis de entrada, descrito pela Figura 4.3, inicialmente, é necessário calcular o coeficiente de correlação entre cada variável secundária e a variável de composição (y_1). O resultado desse cálculo é ilustrado pela Figura 5.5.

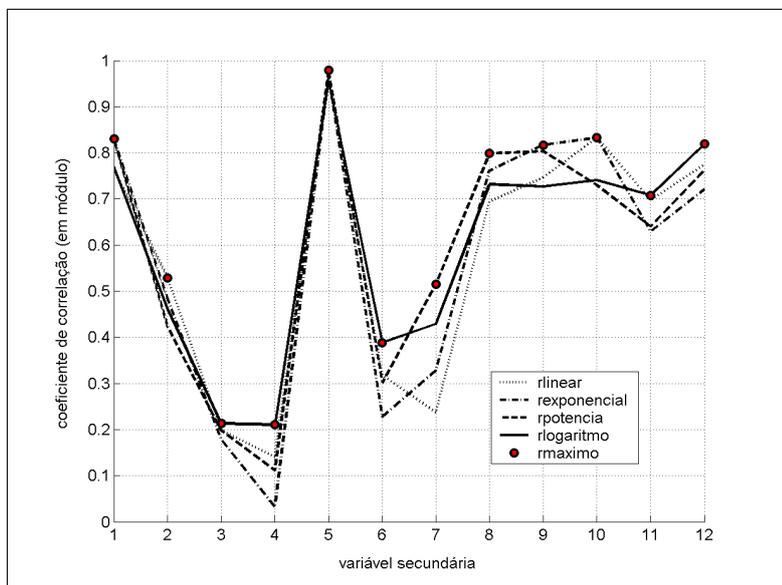


Figura 5.5: Cálculo da correlação para a inferência da composição do produto de topo (coluna binária).

Essas variáveis secundárias são ordenadas conforme o respectivo valor absoluto do coeficiente de correlação. A Tabela 5.2 especifica a ordenação dessas variáveis.

Tabela 5.2: Classificação das variáveis de acordo com o valor absoluto da correlação para a inferência do produto de topo (coluna binária).

posição	1º	2º	3º	4º	5º	6º
variável	u_5	u_8	u_9	u_1	u_{10}	u_7
valor absoluto de correlação	0.9909	0.8945	0.8933	0.8524	0.8319	0.7924
posição	7º	8º	9º	10º	11º	12º
variável	u_{12}	u_{11}	u_2	u_6	u_3	u_4
valor absoluto de correlação	0.7743	0.6973	0.5265	0.4062	0.1987	0.1948

A partir dos valores de coeficiente de correlação, os subconjuntos de variáveis secundárias são gerados conforme o **passo 2** do algoritmo de seleção das variáveis de entrada (Figura 4.3). A Tabela 5.3 especifica as variáveis secundárias que compõem cada subconjunto, onde c_i é o subconjunto composto por i variáveis secundárias, $i = 1, \dots, 12$, e u_i é a i -ésima variável secundária, especificada na Tabela 5.1.

Tabela 5.3: Especificação dos subconjuntos de variáveis secundárias para a inferência da composição do produto de topo (coluna binária).

Subconjuntos	Variáveis
c_1	u_5, y_1
c_2	u_5, u_8, y_1
c_3	u_5, u_8, u_9, y_1
c_4	u_5, u_8, u_9, u_1, y_1
c_5	$u_5, u_8, u_9, u_1, u_{10}, y_1$
c_6	$u_5, u_8, u_9, u_1, u_{10}, u_7, y_1$
c_7	$u_5, u_8, u_9, u_1, u_{10}, u_7, u_{12}, y_1$
c_8	$u_5, u_8, u_9, u_1, u_{10}, u_7, u_{12}, u_{11}, y_1$
c_9	$u_5, u_8, u_9, u_1, u_{10}, u_7, u_{12}, u_{11}, u_2, y_1$
c_{10}	$u_5, u_8, u_9, u_1, u_{10}, u_7, u_{12}, u_{11}, u_2, u_6, y_1$
c_{11}	$u_5, u_8, u_9, u_1, u_{10}, u_7, u_{12}, u_{11}, u_2, u_6, u_3, y_1$
c_{12}	$u_5, u_8, u_9, u_1, u_{10}, u_7, u_{12}, u_{11}, u_2, u_6, u_3, u_4, y_1$

Seria possível estabelecer um valor limite, isto é, apenas os conjuntos com correlação superior a 0.5 podem ser considerados, a fim de reduzir o tempo computacional. Neste caso, somente os conjuntos c_1 a c_8 seriam utilizados. No entanto, de modo a ilustrar a eficácia do método, nestas simulações todos os conjuntos foram considerados. Assim, para cada um dos subconjuntos da Tabela 5.3 foi criado um mapa de Kohonen (*SOM*), conforme o **passo 3** e o **passo 4** do algoritmo de seleção das variáveis de entrada. Depois que todos os *SOM*'s foram treinados (com o algoritmo de aprendizagem do mapa auto-organizável de Kohonen (Figura 2.6)), calculou-se a média de dissimilaridade referente a cada *SOM* em relação a todos os outros *SOM*'s (**passo 6** e **passo 7** do algoritmo da Figura 4.3). O resultado deste cálculo é ilustrado pela Figura 5.6.

Como pode ser observado, o subconjunto com o menor valor de média de dissimilaridade é aquele composto por seis variáveis secundárias, ou seja, o subconjunto c_6 da Tabela 5.3. Em resumo, a metodologia considera que as variáveis: *nível líquido no refervedor* (u_5), *temperatura no prato 16* (u_8), *temperatura no prato 18* (u_9), *vazão molar de alimentação* (u_1), *vazão para o condensador* (u_{10}) e *temperatura do prato 13* (u_7) podem ser usadas para inferir a composição do produto de topo.

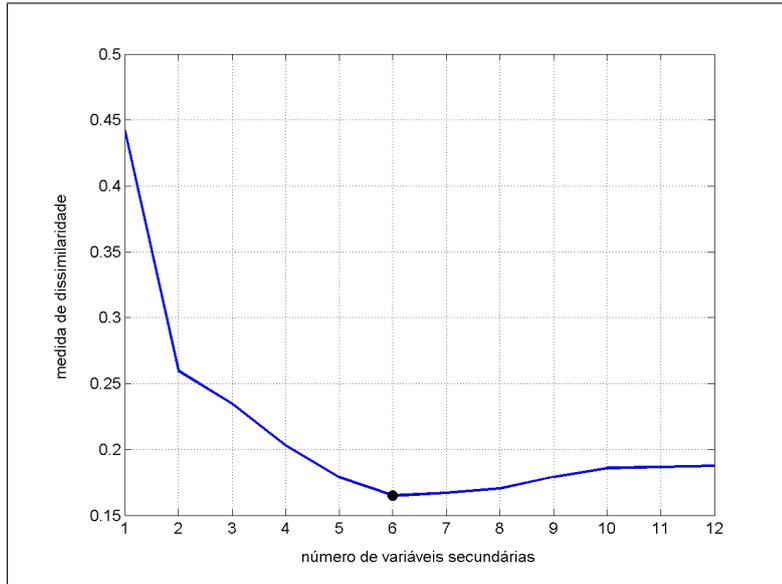


Figura 5.6: Seleção de variáveis para a inferência da composição do produto de topo (coluna binária).

Após a seleção das variáveis de entrada, é necessário determinar a estrutura do modelo *fuzzy*, isto é, a ordem da dinâmica do modelo *fuzzy*. Em modelos estatísticos do tipo *PLS* e *PCA* esta fase não é considerada. A Figura 4.5 descreve o procedimento de seleção da estrutura do modelo.

De modo a diminuir o esforço computacional, a implementação considera que o atraso de y (lag_y) e o atraso de x (lag_x) possuem valores iguais. Foram calculados os quocientes $q^{(z)}$ (equação 4.14) para $z = 0, \dots, 5$, ou seja, considerou-se $lag_x = lag_y = 0, \dots, 5$. A Figura 5.7 ilustra o resultado da aplicação do algoritmo da Figura 4.5, onde é possível observar que o cotovelo da curva fica entre o $lag=1$ e o $lag=2$. Neste caso, os dois $lags$ foram avaliados e o modelo com o $lag=1$ apresentou melhores resultados, sendo, portanto, este o lag selecionado.

Seja $y_1(t)$ a composição do metanol no instante t , x_1, \dots, x_6 as variáveis de entrada selecionadas, e x_7 a composição do metanol utilizada como realimentação do modelo, ou seja, $x_7(t-1) = y_1(t-1)$, o modelo *fuzzy* inferencial pode ser representado por uma função $g(\cdot)$ como na equação (5.1):

$$\hat{y}_1(t) = g(x_1(t-1), \dots, x_6(t-1), x_7(t-1)). \quad (5.1)$$

Definidas as variáveis de entrada e a estrutura do modelo *fuzzy*, a base de regras inicial foi gerada de acordo com o algoritmo de geração da base de regras inicial (Figura 4.9), onde o algoritmo *FCM* (Figura 2.12) foi aplicado ao espaço de entrada e saída. Para esta simulação havia a disponibilidade de um conjunto de dados amos-

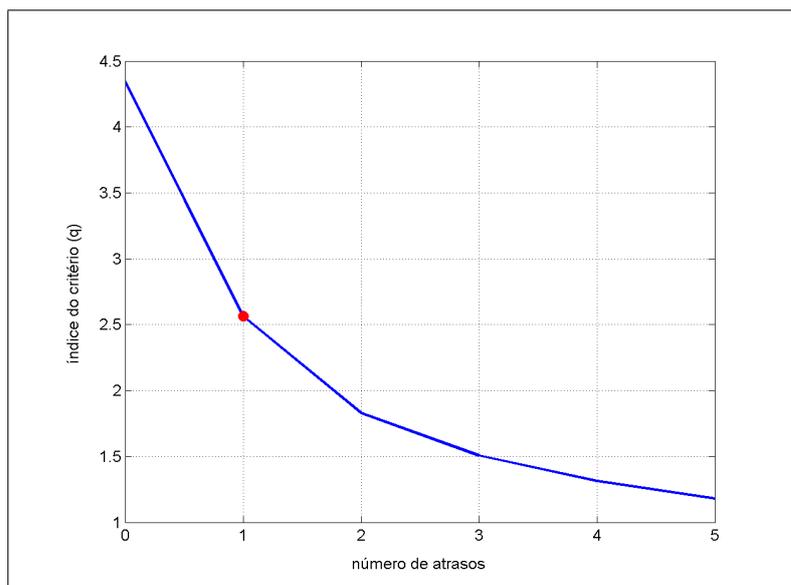


Figura 5.7: Seleção do atraso para a inferência da composição do produto de topo (coluna binária).

trados contendo 550 pares de amostras de entrada e saída. Considerando-se o atraso selecionado, o conjunto disponível foi reduzido para 549. Este conjunto foi dividido em três partes iguais: conjunto de treinamento, conjunto de validação e conjunto de teste. O número de *clusters* (ou o tamanho da base de regras inicial) foi definido como sendo a terça parte do número de amostras do conjunto de treinamento (base de regras inicial = 61 regras), e a faixa de dados considerada foi entre 90% e 99% de pureza do produto de topo (metanol).

As funções de pertinência para cada variável foram obtidas a partir da projeção dos *clusters* e, portanto, inicialmente, o universo de discurso de cada variável é descrito por 61 funções de pertinência. A base de regras inicial foi, então, simplificada conforme o algoritmo de simplificação da base de regras inicial (Figura 4.12). Após esta fase de simplificação, o número de funções de pertinência para cada variável sofreu uma considerável redução, como pode ser observado pela Tabela 5.4, onde é especificada a quantidade de funções de pertinência para cada variável do modelo.

A Figura 5.8 apresenta a partição do universo de discurso de cada uma das variáveis de entrada e saída do modelo *fuzzy* inferencial. Estas partições foram obtidas após a fase de simplificação da base de regras inicial, onde as funções de pertinências similares foram agrupadas. Nesta figura observa-se a partição do domínio antes (linhas tracejadas) e após (linhas contínuas) da simplificação da base de regras.

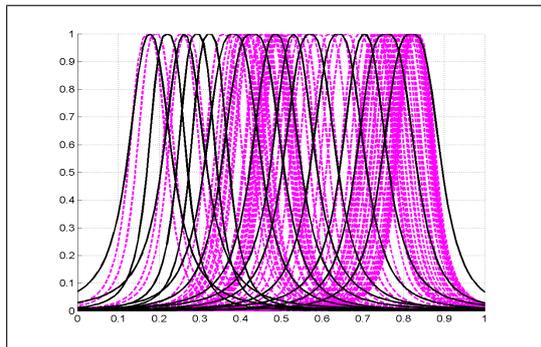
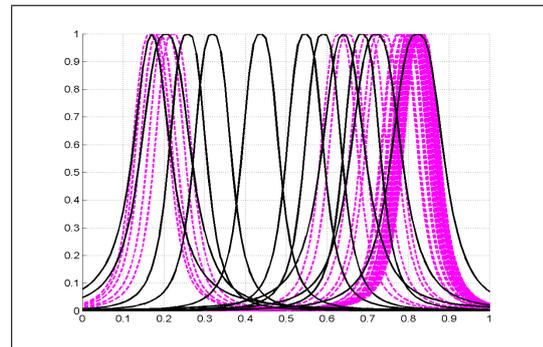
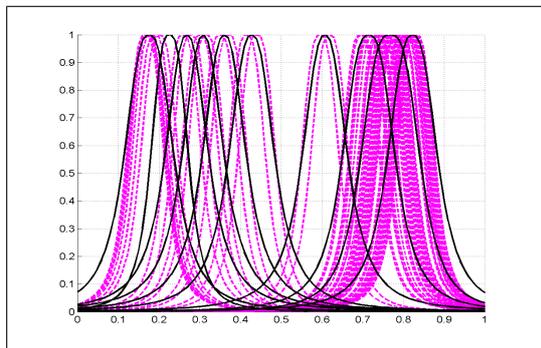
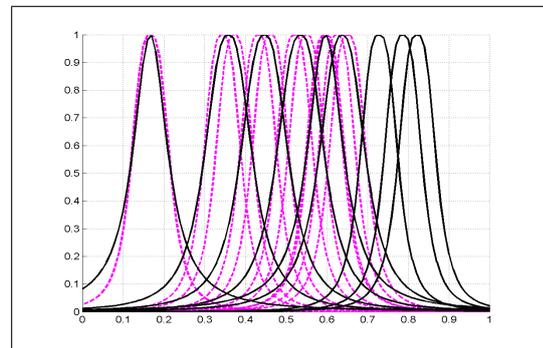
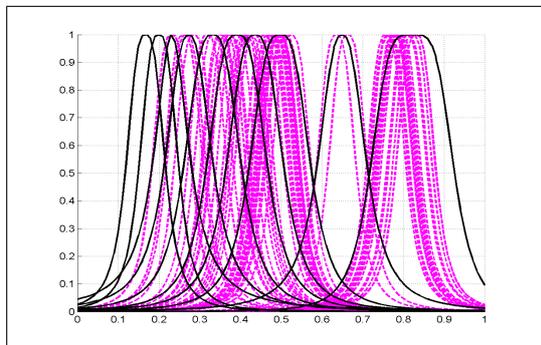
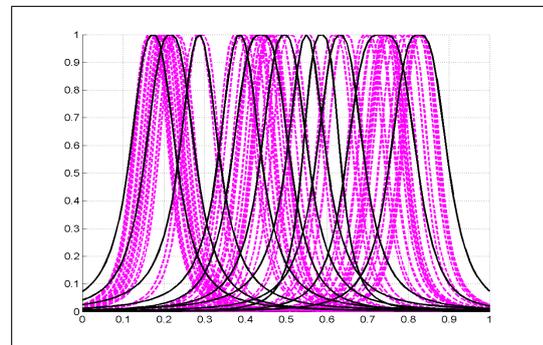
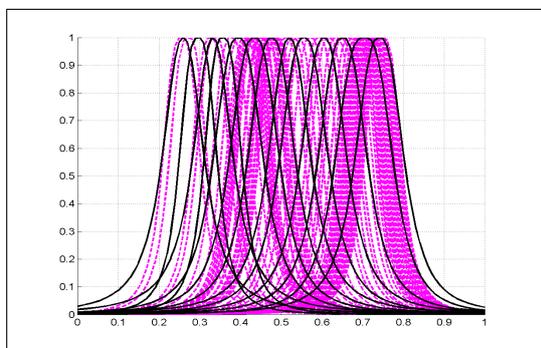
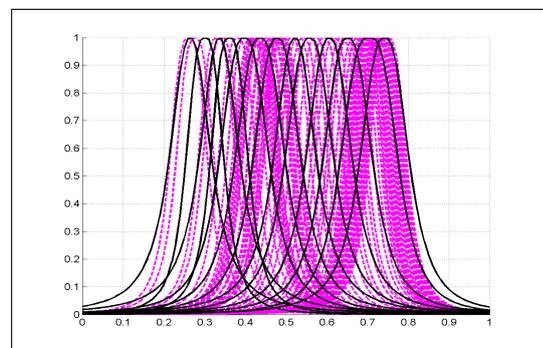
(a) *Nível líq. no condensador (x_1).*(b) *Temp. no prato 18 (x_2).*(c) *Temp. no prato 16 (x_3).*(d) *Vazão molar de alimentação (x_4).*(e) *Vazão para o alimentador (x_5).*(f) *Vazão do produto de fundo (x_6).*(g) *Composição do produto de topo - realimentação (x_7).*(h) *Composição do produto de topo (y_1).*

Figura 5.8: Funções de pertinência das variáveis do modelo fuzzy para a inferência da composição do produto de topo y_1 (coluna binária).

Tabela 5.4: Variáveis selecionadas para a inferência da composição do produto de topo (coluna binária).

#	Variável	# Funções de pertinência
x_1	nível líquido no refeedor (u_5)	14
x_2	temperatura no prato 18 (u_9)	11
x_3	temperatura no prato 16 (u_8)	10
x_4	vazão molar de alimentação (u_1)	9
x_5	vazão para o condensador (u_{10})	10
x_6	temperatura no prato 13 (u_7)	11
x_7	composição do metanol - realimentação (y_1)	13
y_1	composição do metanol	13

A Tabela 5.5 apresenta as informações relevantes sobre o modelo *fuzzy* inferencial obtido para estimar a composição do produto de topo, tais como o tamanho da base de regras, o número de variáveis do modelo e os operadores utilizados no módulo de inferência *fuzzy* (neste caso, operador de agregação de antecedentes t-norma mínimo, operador de implicação t-norma mínimo, operador de agregação das regras s-norma máximo e o método de defuzzificação bisetor da área. O conjunto de operadores foi selecionado de forma empírica, conforme o desempenho por ele alcançado.

Tabela 5.5: Parâmetros do modelo *fuzzy* para a inferência da composição do produto de topo (coluna binária).

Descrição	valor
tamanho da base de regras inicial (ncr)	61
tamanho da base de regras simplificada	42
número de regras inconsistentes encontradas	0
número de variáveis de entrada	7
número de variáveis de saída	1
atraso selecionado	1
operador de agregação dos antecedentes	t-norma mínimo
operador de implicação	t-norma mínimo
operador de agregação das regras	s-norma máximo
operador de defuzzificação	BOA

A Tabela 5.6 apresenta os índices de desempenho do modelo, onde os cálculos do mse (equação (4.26)) e $mape$ (equação (4.27)) foram realizados com os dados no domínio não normalizados. A diferença mínima entre os erros obtidos com o conjunto de treinamento e o conjunto de teste indicam que o modelo obtido apresenta uma boa capacidade de generalização.

Tabela 5.6: Índices de desempenho do modelo inferencial para estimar a composição do produto de topo (coluna binária).

	treinamento	validação	teste
<i>mse</i>	0.0078	0.0110	0.0099
<i>mape</i>	0.0714%	0.0844%	0.0802%

A Figura 5.9 apresenta a comparação entre a composição do produto de topo inferida pelo *soft sensor* (linha pontilhada) e a composição simulada (linha contínua).

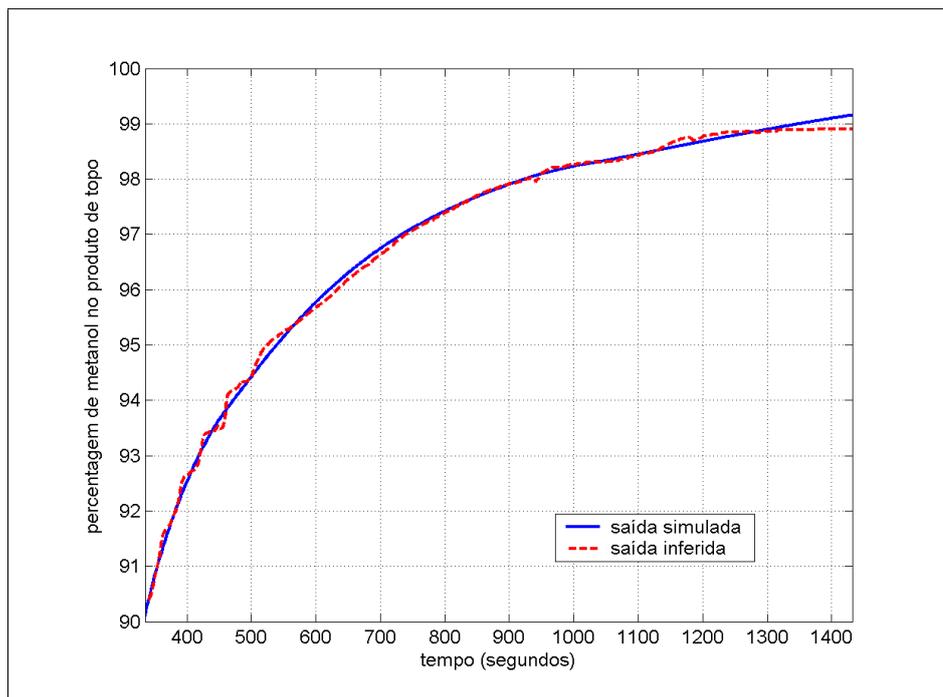


Figura 5.9: Inferência da composição do produto de topo da coluna binária.

Através da Figura 5.9 observa-se que a curva da saída inferida através do modelo *fuzzy* acompanha a tendência da curva simulada. Porém, o modelo obtido apresenta deficiências em gerar as estimativas para os trechos próximos aos limites inferior e superior (aproximadamente 0.20% e 0.25% de erro percentual absoluto, respectivamente).

5.2.1 Estudo Comparativo para a Inferência do Produto de Topo

Em Fabro (2003) foram desenvolvidos dois modelos inferenciais, baseados em redes neurais, para estimar a composição dos produtos de topo e fundo da coluna de destilação. Como somente os resultados do modelo inferencial (neural) para o produto

de topo estavam disponíveis, esta seção apresenta a comparação dos resultados do modelo *fuzzy* inferencial, obtido com a metodologia proposta neste trabalho, para a estimação do produto de topo. Como parâmetros adicionais de comparação, dois modelos inferenciais foram desenvolvidos utilizando-se a *PCR* (seção 2.5.1).

Modelos Inferenciais Baseado em *PCR*

Dois modelos foram gerados com a *PCR* a partir do resultado da *PCA*. Inicialmente, foram especificados os componentes principais, onde cada componente principal é uma combinação linear de todas as variáveis secundárias, descritas pela Tabela 5.1. A Tabela 5.7 apresenta, de forma resumida, a matriz de escores resultantes da análise de componentes principais. Nota-se que a partir da sétima coluna (cp_7) os valores são pequenos, então foram consideradas somente as seis primeiras componentes principais para realizar a regressão.

Tabela 5.7: Matriz de escores \mathbf{T} (produto de topo).

cp_1	cp_2	cp_3	cp_4	cp_5	cp_6	cp_7	cp_8	cp_9	cp_{10}	cp_{11}	cp_{12}
1.0695	0.1602	0.5849	-0.3432	-0.1977	0.1121	-0.0056	-0.0055	-0.0088	-0.0076	0.0099	-0.0000
1.0730	0.1582	0.5746	-0.3368	-0.1957	0.1159	-0.0054	-0.0055	-0.0102	-0.0062	0.0072	-0.0000
1.0764	0.1628	0.5620	-0.3473	-0.1744	0.1045	-0.0027	-0.0044	-0.0061	-0.0047	0.0079	-0.0000
\vdots	\vdots	\vdots									
-0.5610	0.3707	-0.2098	-0.0417	0.1294	0.0293	-0.0155	-0.0144	-0.0081	0.0051	0.0007	-0.0000
-0.5579	0.3700	-0.2150	-0.0402	0.1319	0.0332	-0.0150	-0.0145	-0.0097	0.0054	-0.0003	0.0000
-0.5556	0.3703	-0.2149	-0.0440	0.1344	0.0308	-0.0150	-0.0145	-0.0089	0.0054	0.0004	0.0000

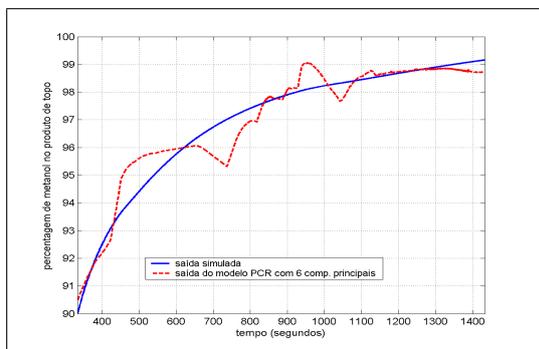
Desta forma, a regressão por *PCR* foi realizada utilizando as seis primeiras componentes principais. O resultado da regressão esta ilustrada na Figura 5.10(a), onde a saída do modelo PCR6, gerado pela *PCR* (linha tracejada), é comparada com a saída simulada (linha contínua). Um segundo modelo foi gerado pela *PCR*, o modelo PCR12 (Figura 5.10(b)), onde foram consideradas todas as componentes principais, para verificar se o desempenho do modelo gerado através da *PCR* poderia ser melhorado.

Modelo Inferencial Neural

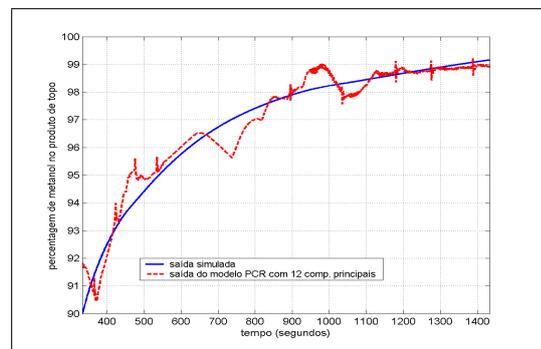
O terceiro modelo de comparação é o modelo inferencial neural desenvolvido em Fabro (2003). Esta rede neural recebe como entrada as temperaturas lidas nos pratos 9,13,16 e 18, e não possui conexões recorrentes. Esta rede é do tipo direta (*feed-forward*), possuindo 3 camadas (com 51,11 e 1 neurônios, respectivamente). A rede foi treinada com o algoritmo *Quick-Propagation* (Fahlmann, 1988), obtendo erros menores que 1%.

Resultados

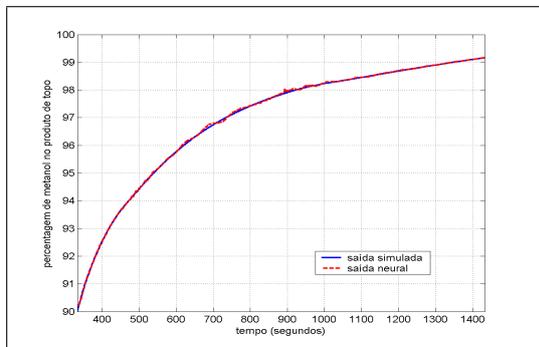
A Figura 5.10 ilustra a saída inferida por cada um dos modelos considerados nesta seção: (a) modelo PCR6 (modelo gerado com a *PCR* utilizando-se 6 componentes principais), (b) modelo PCR12 (modelo gerado com a *PCR* utilizando-se 12 componentes principais), (c) modelo neural e (d) o modelo *fuzzy*. A Tabela 5.8 apresenta os índices de desempenho dos quatro modelos. Observa-se pela Figura 5.10(c) e pela Tabela 5.8 que o desempenho da rede neural foi superior ao desempenho apresentado pelos demais modelos, seguido pelo modelo *fuzzy*.



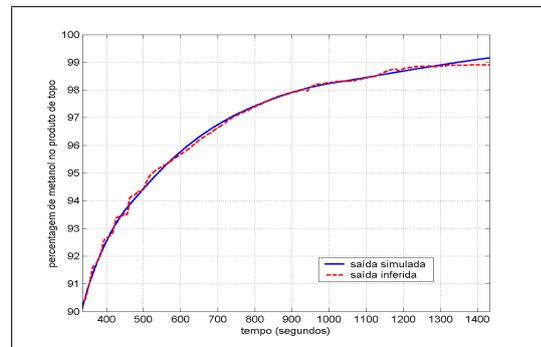
(a) Comparação do modelo PCR6.



(b) Comparação do modelo PCR12.



(c) Comparação do modelo neural.



(d) Comparação do modelo fuzzy.

Figura 5.10: Comparações de modelos para a inferência da composição do produto de topo.

Tabela 5.8: Comparação do desempenho dos modelos PCR6, PCR12, neural e fuzzy.

	PCR6	PCR12	neural	fuzzy
<i>mse</i>	0.3443	0.2374	$9.6844e - 004$	0.0096
<i>mape</i>	0.4323%	0.3710%	0.0215%	0.0787%

5.3 Inferência da Composição do Produto de Fundo

O procedimento para a identificação de um modelo *fuzzy* inferencial para estimar a composição do produto de fundo (y_2) é o mesmo que o utilizado no caso da composição do produto de topo. Em Fabro (2003) foi selecionada a faixa entre 80% a 99% de água na composição do produto de fundo para gerar as estimativas com a rede neural. Entretanto, durante esta faixa de tempo/valores, os valores das variáveis *vazão do produto de topo* e *vazão do produto de fundo* (as variáveis u_{11} e u_{12} da Tabela 5.1, respectivamente) permanecem constantes e iguais a zero. Portanto essas duas variáveis não foram consideradas durante a fase de seleção de variáveis de entrada neste caso.

Assim como no caso da inferência do produto de topo, foram calculados quatro tipos de coeficientes de correlação entre as variáveis secundárias e a variável de saída y_2 , e levou-se em consideração o maior coeficiente (em módulo) para cada variável, como ilustra a Figura 5.11. A Tabela 5.9 apresenta a ordenação das variáveis secundárias conforme o respectivo valor absoluto de correlação.

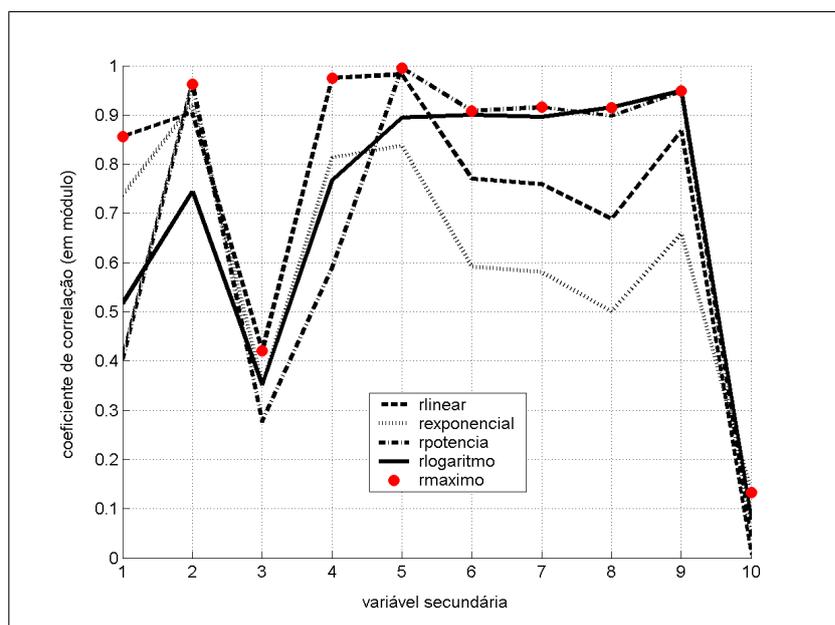


Figura 5.11: Cálculo da correlação para a inferência da composição do produto de fundo (coluna binária).

Com esses valores de coeficiente de correlação, os subconjuntos de variáveis secundárias foram criados de acordo com o **passo 2** do algoritmo de seleção de variáveis de entrada (Figura 4.3), e a composição de cada um desses subconjuntos é detalhada pela Tabela 5.10.

Para cada subconjunto foi criado um *SOM*, conforme o **passo 3** e o **passo 4** do

Tabela 5.9: Classificação das variáveis de acordo com o valor absoluto da correlação para a inferência da composição do produto de fundo (coluna binária).

posição	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º
variável	u_5	u_4	u_2	u_9	u_7	u_8	u_6	u_1	u_3	u_{10}
valor absoluto de correlação	0.9953	0.9752	0.9629	0.9489	0.9155	0.9151	0.9083	0.8562	0.4208	0.1335

Tabela 5.10: Especificação dos subconjuntos de variáveis secundárias para a inferência da composição do produto de fundo (coluna binária).

Subconjuntos	Variáveis
c_1	u_5, y_2
c_2	u_5, u_4, y_2
c_3	u_5, u_4, u, y_2
c_4	u_5, u_4, u_2, u_9, y_2
c_5	$u_5, u_4, u_2, u_9, u_7, y_2$
c_6	$u_5, u_4, u_2, u_9, u_7, u_8, y_2$
c_7	$u_5, u_4, u_2, u_9, u_7, u_8, u_6, y_2$
c_8	$u_5, u_4, u_2, u_9, u_7, u_8, u_6, u_1, y_2$
c_9	$u_5, u_4, u_2, u_9, u_7, u_8, u_6, u_1, u_3, y_2$
c_{10}	$u_5, u_4, u_2, u_9, u_7, u_8, u_6, u_1, u_3, u_{10}, y_2$

algoritmo de seleção de variáveis de entrada e foi calculada a média de dissimilaridade para cada *SOM*, cujo resultado é ilustrado pela Figura 5.12, onde o subconjunto c_6 com seis variáveis secundárias foi selecionado. Este conjunto é composto pelas variáveis *nível líquido no condensador* (u_5), *nível líquido do refeedor* (u_4), *refluxo do condensador para a coluna* (u_2), *temperatura no prato 18* (u_9), *temperatura no prato 13* (u_7), *temperatura no prato 16* (u_8).

O próximo procedimento empregado foi o de seleção da estrutura do modelo *fuzzy*, descrito pela Figura 4.5. Novamente, foi considerado $lag_x = lag_y$ durante o cálculo dos quocientes de Lipschitz $q^{(z)}$ (equação (4.14)), com $z = 0, \dots, 5$. A Figura 5.13 ilustra o resultado do procedimento. Assim como no caso do produto de topo, o cotovelo da curva fica entre o $lag=1$ e o $lag=2$. Novamente verificou-se qual escolha de atraso, $lag=1$ ou $lag=2$, apresentava melhor desempenho, e o atraso selecionado foi o $lag=1$, ou seja $lag_x = lag_y = 1$.

Sejam x_1, \dots, x_6 as variáveis selecionadas e x_7 a realimentação da saída y_2 (composição da água) as variáveis de entrada e y_2 a variável de saída do modelo *fuzzy* inferencial. O modelo *fuzzy* inferencial para a estimação da composição do produto de fundo pode ser representado por uma função $f(\cdot)$ como na equação (5.2):

$$\hat{y}_2(t) = f(x_1(t-1), \dots, x_6(t-1), x_7(t-1)). \quad (5.2)$$

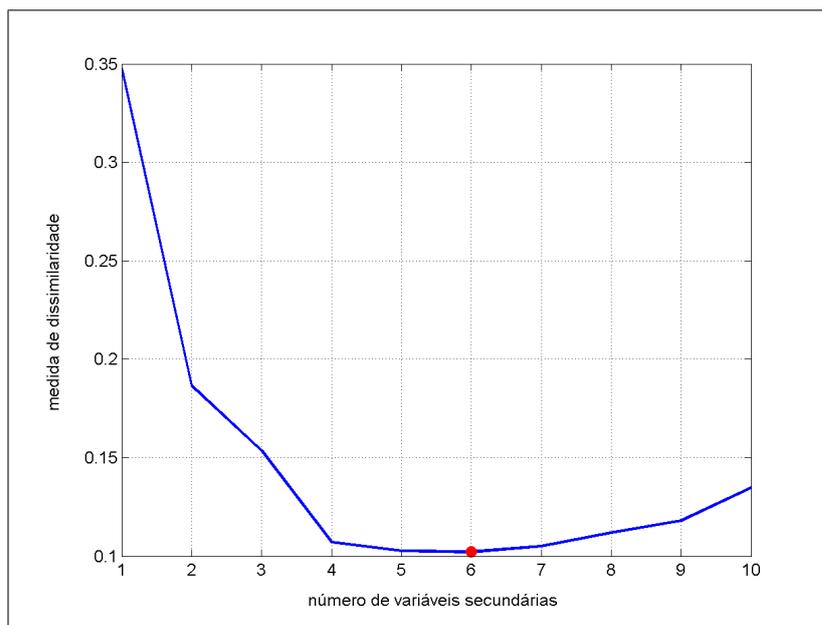


Figura 5.12: Seleção de variáveis para a inferência da composição do produto de fundo (coluna binária).

Após a definição das variáveis de entrada e da estrutura do modelo *fuzzy*, a base de regras inicial foi gerada conforme o algoritmo de geração da base de regras inicial (Figura 4.9). No caso da inferência da composição do produto de fundo, o conjunto de dados disponível é composto por 110 pares de dados e considerando-se o atraso selecionado ($lag=1$), o conjunto de dados amostrados passou a ter 109 pares de dados de entrada e saída. Devido ao reduzido tamanho do conjunto de dados amostrados neste caso, utilizou-se a metade das amostras para a identificação (dados de treinamento), um quarto dos dados para a validação e um quarto para o teste.

O tamanho da base de regras foi estipulado como sendo a metade do número de pares de dados do conjunto de treinamento. Dessa forma, um algoritmo *FCM* com 28 *clusters* foi aplicado ao espaço de entrada e saída. Estes *clusters* foram, então, projetados no domínio das variáveis para a obtenção dos valores discretos das funções de pertinência de cada uma dessas variáveis.

Os parâmetros das respectivas funções de pertinências são obtidos conforme descreve o algoritmo de geração da base de regras inicial. Este procedimento gera uma base de regras inicial composto por 28 regras *fuzzy*, e devido à projeção dos *fuzzy*, o universo de discurso de cada variável é inicialmente descrito por 28 funções de pertinência.

O procedimento de simplificação é aplicado à base de regras inicial, resultando

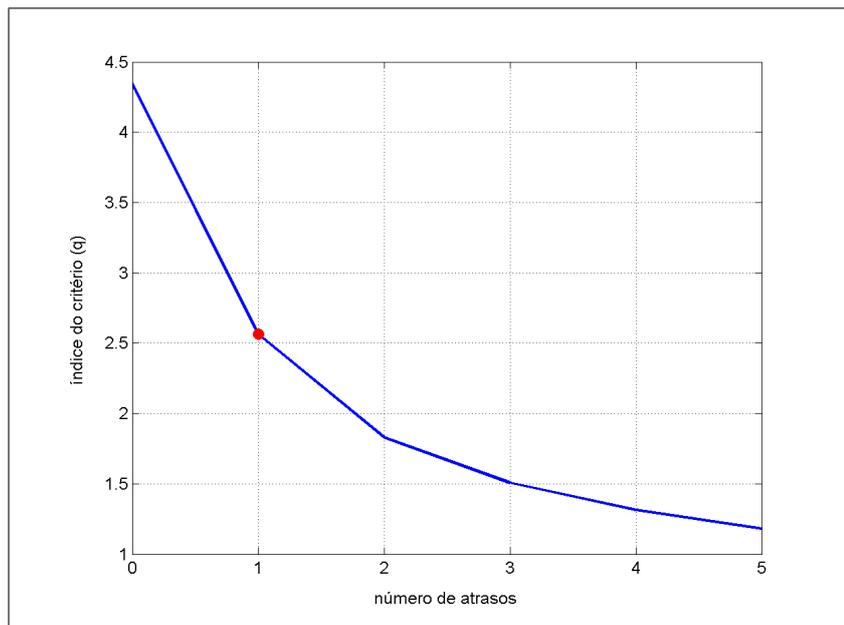


Figura 5.13: Seleção do atraso para a inferência da composição do produto de fundo (coluna binária).

na partição descrita pela Tabela 5.11. A partição do universo de discurso de cada variável do modelo *fuzzy* é descrito pela Figura 5.14.

Tabela 5.11: Variáveis selecionadas para a inferência da composição do produto de fundo (coluna binária).

#	Variável	# Funções de pertinência
x_1	nível líquido no condensador (u_5)	10
x_2	nível líquido do refeedor (u_4)	10
x_3	refluxo do condensador para a coluna (u_2)	10
x_4	temperatura no prato 18 (u_9)	10
x_5	temperatura no prato 13 (u_7)	9
x_6	temperatura no prato 16 (u_8)	11
x_7	composição da água - realimentação (y_2)	14
y_2	composição da água	13

As informações sobre o modelo *fuzzy* inferencial obtido para estimar a composição do produto de fundo são apresentados na Tabela 5.12. O módulo de inferência do modelo *fuzzy* inferencial utiliza os mesmos operadores do modelo gerado para a inferência da composição do produto de topo, ou seja, operador de agregação dos antecedentes t-norma mínimo, operador de implicação t-norma mínimo, operador de agregação de regras s-norma máximo e o método de defuzzificação bisetor da área.

Os índices de desempenho do modelo são ilustrados na Tabela 5.13. Como na seção anterior, os índices de erro foram calculados considerando-se os valores reais,

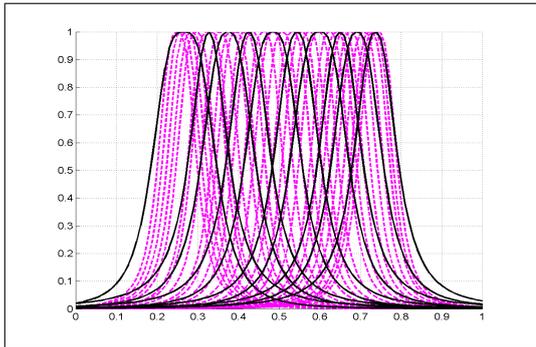
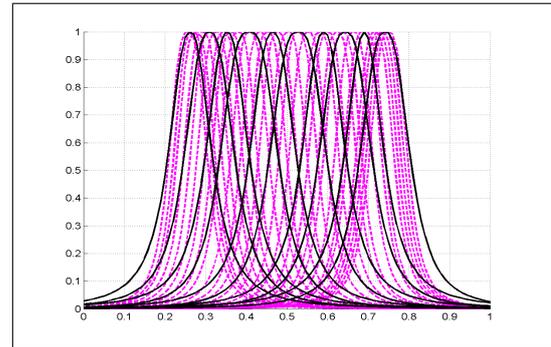
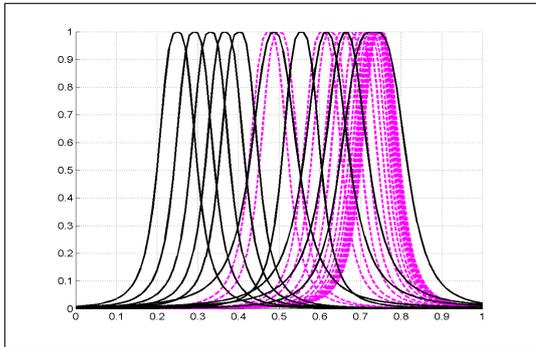
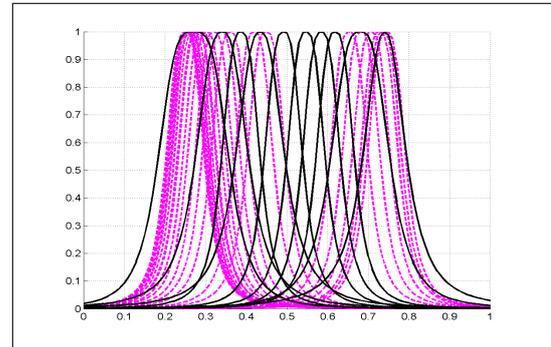
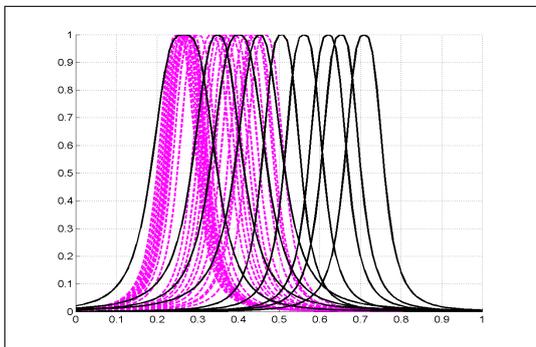
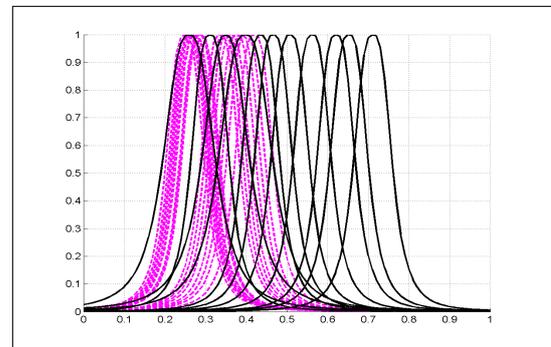
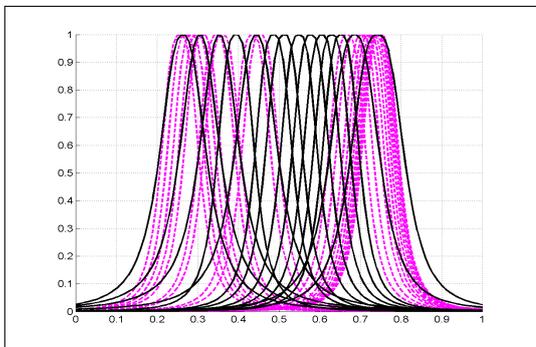
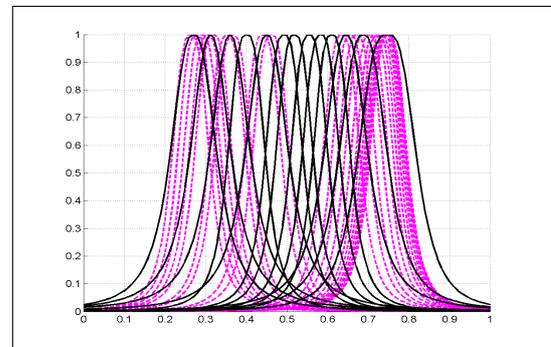
(a) *Nível líq. no condensador (x_1).*(b) *Nível líq. do refeedor (x_2).*(c) *Refluxo do condensador para a coluna (x_3).*(d) *Temp. no prato 18 (x_4).*(e) *Temp. no prato 13 (x_5).*(f) *Temp. no prato 16 (x_6).*(g) *Composição do produto de fundo - y_2 - realimentação (x_7).*(h) *Composição do produto de fundo (y_2).*

Figura 5.14: Funções de pertinência das variáveis do modelo fuzzy para a inferência da composição do produto de fundo (coluna binária).

Tabela 5.12: Parâmetros do modelo fuzzy para a inferência da composição do produto de fundo (coluna binária).

Descrição	valor
tamanho da base de regras inicial (<i>ncr</i>)	28
tamanho da base de regras simplificada	26
número de regras inconsistentes	0
número de variáveis de entrada	7
número de variáveis de saída	1
<i>lag</i> selecionado	1
operador de agregação dos antecedentes	t-norma mínimo
operador de implicação	t-norma mínimo
operador de agregação das regras	s-norma máximo
operador de defuzzificação	<i>BOA</i>

e não os normalizados. Novamente, o modelo obtido apresenta boa generalização, como descreve a Tabela 5.13.

Tabela 5.13: Índices de desempenho do modelo inferencial para estimar a composição do produto de fundo (coluna binária).

	treinamento	validação	teste
<i>mse</i>	0.0675	0.0447	0.0684
<i>mape</i>	0.2272%	0.1799%	0.2297%

A Figura 5.15 apresenta a comparação entre a composição do produto de fundo inferida pelo *soft sensor* (linha pontilhada) e a composição simulada (linha contínua). Como no caso do modelo *fuzzy* inferencial para a composição do produto de topo, este modelo também não é capaz de inferir adequadamente os valores próximos aos limites do universo de discurso (80% e 99%) apresentando erros relativos de aproximadamente 0.60% na região do limite inferior e 0.22% na região do limite superior, que mesmo assim são erros considerados pequenos no problema em questão.

Além disso, a quantidade de pares de dados disponíveis para a faixa de dados considerada é um pouco mais do que 100. Este número reduzido de pares de dados não se mostrou suficiente para realizar a identificação de um modelo inferencial de modo satisfatório, pois somente a metade destes dados são utilizados para a identificação, e para não ter problemas de *overfitting* ao gerar o modelo *fuzzy*, estipulou-se que o tamanho da base de regras inicial não fosse maior do que a metade do número de pares de dados do conjunto de treinamento. No caso da identificação do modelo para a inferência da composição do produto de topo, foi possível utilizar um tamanho de base de regras inicial maior, pois havia a disponibilidade de mais amostras

de dados de entrada e saída. No entanto, apesar deste problema, que é um dos mais críticos no levantamento de modelos estatísticos e neurais, a metodologia proposta apresentou bons resultados.

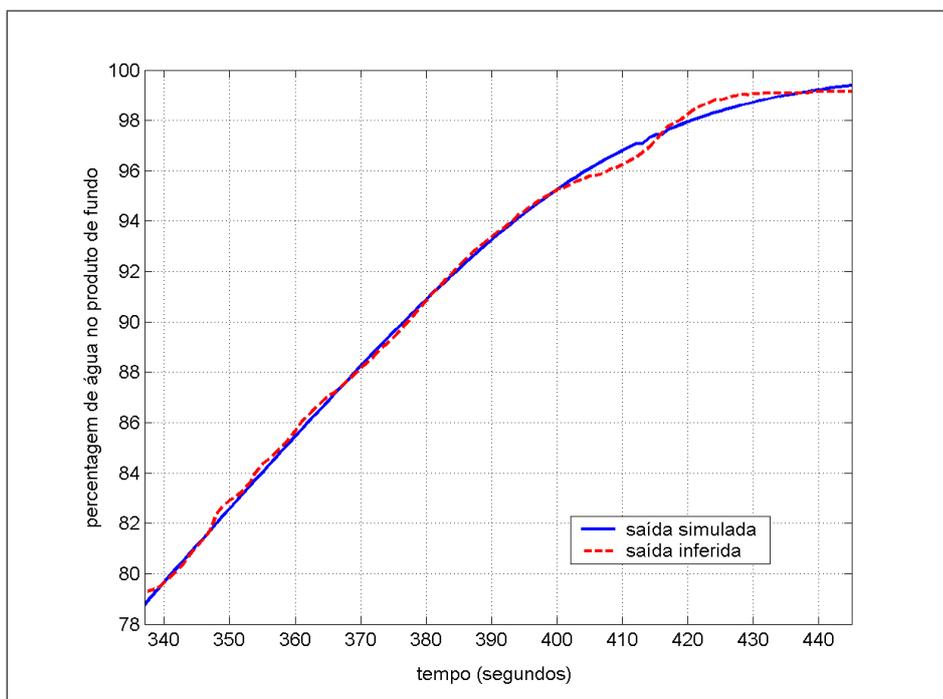


Figura 5.15: Inferência da composição do produto de base da coluna binária.

5.3.1 Estudo Comparativo para a Inferência do Produto de Fundo

Como mencionado anteriormente, não havia a disponibilidade dos dados referentes ao estimador neural desenvolvido em Fabro (2003). Desta forma, para realizar a comparação, uma rede neural foi construída para estimar o produto de fundo da coluna binária, assim como dois modelos baseados em *PCR*. A descrição dos modelos e os resultados obtidos estão descritos nesta seção.

Modelos Inferenciais Baseados em *PCR*

Da mesma forma que no estudo comparativo para a inferência do produto de topo (seção 5.2.1), o primeiro passo é especificar os componentes principais, onde cada componente principal é uma combinação linear de todas as variáveis secundárias, descritas pela Tabela 5.1, considerando-se que as variáveis *vazão do produto de topo*

e vazão do produto de fundo (as variáveis u_{11} e u_{12} da Tabela 5.1, respectivamente) não são utilizadas neste caso.

A Tabela 5.14 apresenta, de forma resumida, a matriz de escores resultantes da análise de componentes principais. Nota-se que a partir da oitava coluna (cp_8) os valores são pequenos, então foram consideradas somente as sete primeiras componentes principais para realizar a regressão.

Tabela 5.14: Matriz de escores \mathbf{T} (produto de fundo).

cp_1	cp_2	cp_3	cp_4	cp_5	cp_6	cp_7	cp_8	cp_9	cp_{10}
-0.2954	0.0634	0.0131	0.0133	-0.0040	-0.0030	-0.0020	0.0001	0.0001	-0.0000
-0.2814	0.0519	0.0138	0.0151	-0.0024	-0.0022	-0.0012	-0.0001	0.0000	0.0000
-0.2687	0.0421	0.0141	0.0165	-0.0010	-0.0012	-0.0006	-0.0003	-0.0000	0.0000
\vdots									
0.3497	0.0405	-0.0499	0.0106	-0.0041	0.0005	0.0004	-0.0001	-0.0000	0.0000
0.3644	0.0426	-0.0566	0.0115	-0.0057	0.0012	0.0004	-0.0001	-0.0000	0.0000
0.3763	0.0510	-0.0655	0.0100	-0.0080	0.0016	0.0004	-0.0003	-0.0001	-0.0000

Desta forma, a regressão por *PCR* foi realizada utilizando as sete primeiras componentes principais. O resultado da regressão esta ilustrada na Figura 5.16(a), onde a saída do modelo PCR7, gerada pela *PCR* (linha tracejada), é comparada com a saída simulada (linha contínua). Um segundo modelo foi gerado pela *PCR*, o modelo PCR10 (Figura 5.16(b)), onde foram consideradas todas as componentes principais, para verificar se o desempenho do modelo gerado através da *PCR* poderia ser melhorado.

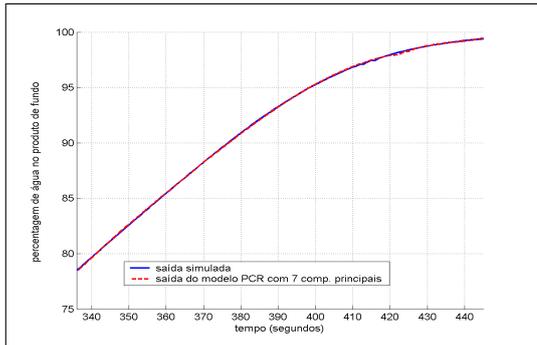
Modelo Inferencial Neural

Para o desenvolvimento do modelo inferencial neural foi utilizada uma rede neural *MLP* composta por três camadas, com dez neurônios na camada de entrada, correspondentes às dez variáveis secundárias disponíveis para a inferência do produto de topo, três neurônios na camada escondida e um neurônio na camada de saída. A escolha do número de neurônios da camada escondida foi realizada através de tentativa e erro. O algoritmo de treinamento utilizado foi o *backpropagation*. As estimativas geradas por este modelo neural pode ser observado pela Figura 5.16(c).

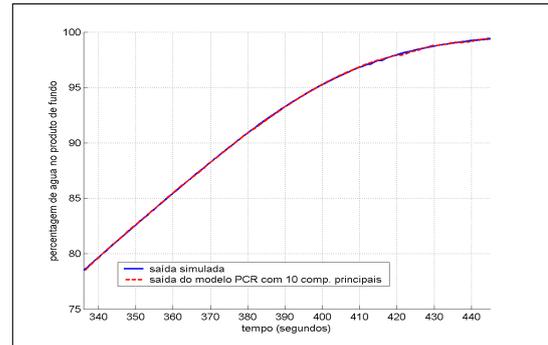
Resultados

A Figura 5.16 ilustra a saída inferida por cada um dos modelos considerados nesta seção: (a) modelo PCR7 (modelo gerado com a *PCR* utilizando-se sete componentes

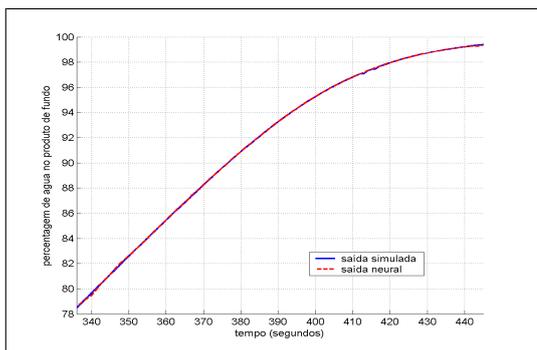
principais), (b) modelo PCR10 (modelo gerado com a *PCR* utilizando-se 10 componentes principais), (c) modelo *fuzzy* e (d) modelo neural,. A Tabela 5.15 apresenta os índices de desempenho dos quatro modelos. Novamente, observa-se pela Figura 5.16(c) e pela Tabela 5.15 que o desempenho da rede neural foi superior ao desempenho apresentado pelos demais modelos, e o modelo *fuzzy* apresentou o pior resultado.



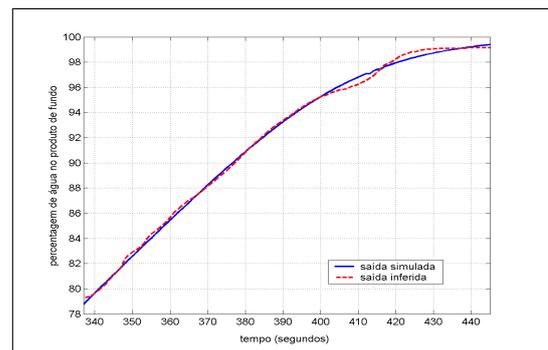
(a) Comparação do modelo PCR7.



(b) Comparação do modelo PCR10.



(c) Comparação do modelo neural.



(d) Comparação do modelo fuzzy.

Figura 5.16: Comparações de modelos para a inferência da composição do produto de topo.

Tabela 5.15: Comparação do desempenho dos modelos PCR7, PCR10, neural e fuzzy.

	PCR7	PCR10	neural	fuzzy
<i>mse</i>	0.0027	0.0022	0.0015	0.0684
<i>mape</i>	0.0449%	0.0379%	0.0229%	0.2297%

Capítulo 6

Coluna Fracionadora de Óleo Cru

Nesta seção será considerado um modelo de coluna fracionadora de óleo cru, conhecido na literatura como *Processo Shell*, inicialmente introduzido por Pretti & Morari (1987). Como explicado em Pretti & Morari (1987), esta coluna não representa um processo real, mas foi projetada para conter todos os elementos importantes de problemas de fracionadoras reais.

6.1 Descrição da Coluna Fracionadora de Óleo Cru

A Figura 6.1¹ ilustra um esquema simplificado da coluna fracionadora. Como pode ser observado, a coluna possui três vazões de saída (produto de topo, produto lateral, e produto de fundo) e três malhas de circulação lateral para a remoção de calor. O calor necessário entra pelo fundo da coluna, juntamente com a alimentação que é gasosa. As especificações dos produtos de topo e lateral são determinadas por necessidades econômicas e os valores dessas composições são medidos pelos analisadores localizados nos pontos de remoção dos produtos de topo e lateral, respectivamente. Para o produto de fundo não existem especificações, porém há uma restrição de operação sobre a temperatura na parte inferior da coluna. As três malhas de circulação removem o calor a fim de alcançar a separação apropriada do produto. O calor removido nestas malhas aquece outras partes da coluna, as quais têm necessidades de taxas de calor diferentes. Assim, os fluxos de calor nessas malhas são considerados variantes no tempo. A malha de fundo possui um controlador de entalpia que regula a remoção de calor através do controle de refluxo. A remoção de calor nas outras duas malhas atua como perturbação na coluna (Ansari & Tadé, 2000).

¹Figura obtida de (Tyagunov, 2004).

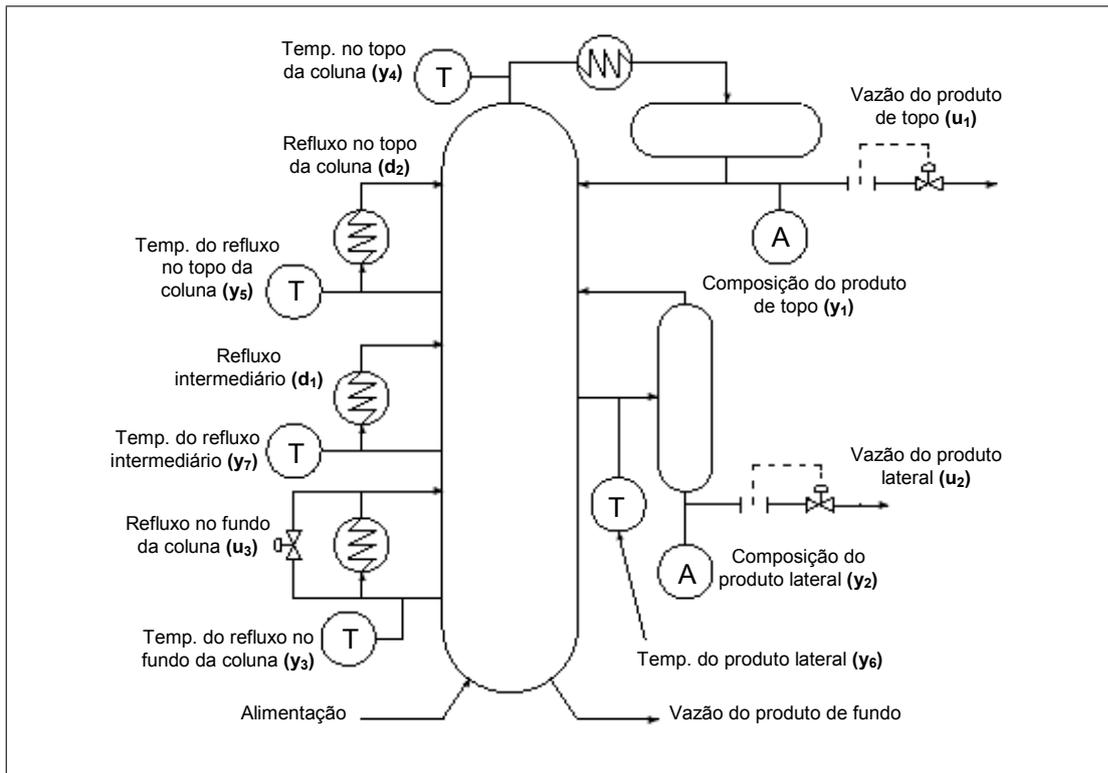


Figura 6.1: Modelo da coluna fracionadora de óleo cru.

De forma resumida, o modelo da coluna fracionadora de óleo cru possui sete saídas ($y_1, y_2, y_3, y_4, y_5, y_6, y_7$) e cinco entradas (u_1, u_2, u_3, d_1, d_2). Três dessas entradas (u_1, u_2, u_3) podem ser utilizadas para fins de controle e duas (d_1, d_2) são consideradas perturbações não mensuráveis. Das sete saídas, três (y_1, y_2, y_3) estão envolvidas nos objetivos de controle e as outras quatro (y_4, y_5, y_6, y_7) são consideradas auxiliares (Ansari & Tadé, 2000). A Tabela 6.1 apresenta as variáveis do modelo da coluna fracionadora.

O modelo discreto usado para descrever o comportamento da fracionadora para fins de controle é um modelo multivariável representado por uma matriz de funções de transferência de primeira ordem com atraso, como descrito na equação (6.1).

$$G(z) = \frac{Kz^{-d}(b_0z + b_1)}{z + a_1}. \quad (6.1)$$

Os valores para K, d, a_1, b_0 e b_1 são apresentados na Tabela 6.2. Estas funções de transferência descrevem o comportamento da planta nas condições nominais de operação, para um período de amostragem de 4 minutos.

Uma das principais características deste modelo é apresentar ganhos incertos em regime. Estes ganhos são correlatos como mostrados na Tabela 6.3, onde a correlação

Tabela 6.1: Descrição das variáveis da coluna fracionadora.

Variável	Descrição
u_1	vazão do produto de topo
u_2	vazão do produto lateral
u_3	refluxo no fundo da coluna
d_1	refluxo intermediário
d_2	refluxo no topo da coluna
y_1	composição do produto de topo
y_2	composição do produto na lateral
y_3	temperatura do refluxo no fundo da coluna
y_4	temperatura no topo da coluna
y_5	temperatura do refluxo no topo da coluna
y_6	temperatura do produto lateral
y_7	temperatura do refluxo intermediário

é parametrizada através de variáveis ϵ_i , $i = 1, \dots, 5$ no intervalo de $[-1, 1]$. Para cada valor de $\Delta\epsilon_i$ neste intervalo, o processo é diferente do modelo nominal.

Para os exemplos simulados neste trabalho, esses valores foram gerados por uma função aleatória, cujo instante de transição entre os níveis não é periódico, mas sim definido como na equação 6.2, obtida de Nørgard et al. (2000), onde $e(t)$ é uma função aleatória. A Figura 6.2 apresenta os valores obtidos para ϵ_i , $i = 1, \dots, 5$, com $\alpha = 0.99$ na primeira metade dos dados e $\alpha = 0.95$ na segunda metade, com os valores variando dentro do intervalo $[-1, 1]$

$$u(t) = \begin{cases} u(t-1), & \text{com probabilidade } \alpha \\ e(t), & \text{com probabilidade } (1 - \alpha) \end{cases} \quad (6.2)$$

A Figura 6.3 apresenta os gráficos com as variáveis de entrada do modelo da coluna fracionadora (u_1 , u_2 , u_3 , d_1 , d_2) também geradas conforme a equação (6.2), dentro do intervalo $[-0.5, 0.5]$, com probabilidade $\alpha = 0.95$ na primeira metade dos dados e $\alpha = 0.99$ na segunda metade dos dados.

A Figura 6.4 ilustra as saídas obtidas com a aplicação das variáveis u_1 , u_2 , u_3 , d_1 , d_2 no modelo discreto descrito na Tabela 6.2.

Foram realizadas simulações considerando-se dois conjuntos diferentes de variáveis secundárias com o objetivo de inferir a composição do produto de topo da coluna fracionadora (variável y_1 da Tabela 6.1).

Tabela 6.2: Modelo discreto da coluna fracionadora.

	(u_1) vazão do produto de topo	(u_2) vazão do produto lateral	(u_3) refluxo no fundo da coluna	(d_1) refluxo intermediário	(d_2) refluxo no topo da coluna
(y_1) composição do produto de topo	$K = 4.05$ $d = 7$ $a_1 = -0.9231$ $b_0 = 0.0198$ $b_1 = 0.0571$	$K = 1.77$ $d = 7$ $a_1 = -0.9355$ $b_0 = 0.0$ $b_1 = 0.0654$	$K = 5.88$ $d = 7$ $a_1 = -0.9231$ $b_0 = 0.0198$ $b_1 = 0.0571$	$K = 1.20$ $d = 7$ $a_1 = -0.9150$ $b_0 = 0.0220$ $b_1 = 0.0630$	$K = 1.44$ $d = 7$ $a_1 = -0.9048$ $b_0 = 0.0247$ $b_1 = 0.0705$
(y_2) composição do produto lateral	$K = 5.39$ $d = 5$ $a_1 = -0.9231$ $b_0 = 0.0392$ $b_1 = 0.0377$	$K = 5.72$ $d = 4$ $a_1 = -0.9355$ $b_0 = 0.0328$ $b_1 = 0.0317$	$K = 6.90$ $d = 4$ $a_1 = -0.9048$ $b_0 = 0.0247$ $b_1 = 0.0705$	$K = 1.52$ $d = 4$ $a_1 = -0.8521$ $b_0 = 0.0392$ $b_1 = 0.1087$	$K = 1.83$ $d = 4$ $a_1 = -0.8187$ $b_0 = 0.0488$ $b_1 = 0.1325$
(y_3) temperatura do refluxo no fundo da coluna	$K = 4.38$ $d = 5$ $a_1 = -0.8859$ $b_0 = 0.0$ $b_1 = 0.1141$	$K = 4.42$ $d = 6$ $a_1 = -0.9131$ $b_0 = 0.0444$ $b_1 = 0.0425$	$K = 7.20$ $d = 0$ $a_1 = -0.8102$ $b_0 = 0.0$ $b_1 = 0.1898$	$K = 1.14$ $d = 0$ $a_1 = -0.8623$ $b_0 = 0.0$ $b_1 = 0.1377$	$K = 1.26$ $d = 0$ $a_1 = -0.8825$ $b_0 = 0.0$ $b_1 = 0.1175$
(y_4) temperatura no topo da coluna	$K = 3.66$ $d = 1$ $a_1 = -0.6412$ $b_0 = 0.1993$ $b_1 = 0.1595$	$K = 1.65$ $d = 5$ $a_1 = -0.8752$ $b_0 = 0.0$ $b_1 = 0.1248$	$K = 5.53$ $d = 1$ $a_1 = -0.9048$ $b_0 = 0.0488$ $b_1 = 0.0464$	$K = 1.16$ $d = 0$ $a_1 = -0.6951$ $b_0 = 0.0$ $b_1 = 0.3049$	$K = 1.27$ $d = 0$ $a_1 = -0.5134$ $b_0 = 0.0$ $b_1 = 0.4866$
(y_5) temperatura do refluxo no fundo da coluna	$K = 5.92$ $d = 3$ $a_1 = -0.7165$ $b_0 = 0.0800$ $b_1 = 0.2035$	$K = 2.54$ $d = 3$ $a_1 = -0.8623$ $b_0 = 0.0$ $b_1 = 0.1377$	$K = 8.10$ $d = 0$ $a_1 = -0.8187$ $b_0 = 0.0952$ $b_1 = 0.0861$	$K = 1.73$ $d = 0$ $a_1 = -0.4493$ $b_0 = 0.0$ $b_1 = 0.5507$	$K = 1.79$ $d = 0$ $a_1 = -0.8102$ $b_0 = 0.0$ $b_1 = 0.1898$
(y_6) temperatura do produto lateral	$K = 4.13$ $d = 2$ $a_1 = -0.6065$ $b_0 = 0.3127$ $b_1 = 0.0808$	$K = 2.38$ $d = 2$ $a_1 = -0.8102$ $b_0 = 0.0513$ $b_1 = 0.1385$	$K = 6.23$ $d = 1$ $a_1 = -0.6703$ $b_0 = 0.1813$ $b_1 = 0.1484$	$K = 1.31$ $d = 0$ $a_1 = -0.1353$ $b_0 = 0.0$ $b_1 = 0.8647$	$K = 1.26$ $d = 0$ $a_1 = -0.8338$ $b_0 = 0.0$ $b_1 = 0.1662$
(y_7) temperatura do refluxo intermediário	$K = 4.06$ $d = 2$ $a_1 = -0.7351$ $b_0 = 0.0$ $b_1 = 0.2649$	$K = 4.18$ $d = 1$ $a_1 = -0.8859$ $b_0 = 0.0$ $b_1 = 0.1141$	$K = 6.53$ $d = 1$ $a_1 = -0.6412$ $b_0 = 0.2835$ $b_1 = 0.0753$	$K = 1.19$ $d = 0$ $a_1 = -0.8102$ $b_0 = 0.0$ $b_1 = 0.1898$	$K = 1.17$ $d = 0$ $a_1 = -0.8465$ $b_0 = 0.0$ $b_1 = 0.1535$

Tabela 6.3: Descrição das incertezas nos ganhos em regime da coluna fracionadora.

	u_1	u_2	u_3	d_1	d_2
y_1	$4.05 + 2.11\epsilon_1$	$1.77 + 0.39\epsilon_2$	$5.88 + 0.59\epsilon_3$	$1.20 + 0.12\epsilon_4$	$1.44 + 0.16\epsilon_5$
y_2	$5.39 + 3.29\epsilon_1$	$5.72 + 0.57\epsilon_2$	$6.90 + 0.59\epsilon_3$	$1.52 + 0.13\epsilon_4$	$1.83 + 0.13\epsilon_5$
y_3	$4.38 + 3.11\epsilon_1$	$4.42 + 0.73\epsilon_2$	$7.20 + 1.33\epsilon_3$	$1.14 + 0.18\epsilon_4$	$1.26 + 0.18\epsilon_5$
y_4	$3.66 + 2.29\epsilon_1$	$1.65 + 0.35\epsilon_2$	$5.53 + 0.67\epsilon_3$	$1.16 + 0.08\epsilon_4$	$1.27 + 0.08\epsilon_5$
y_5	$5.92 + 2.34\epsilon_1$	$2.54 + 0.24\epsilon_2$	$8.10 + 0.32\epsilon_3$	$1.73 + 0.02\epsilon_4$	$1.79 + 0.04\epsilon_5$
y_6	$4.13 + 1.71\epsilon_1$	$2.83 + 0.93\epsilon_2$	$6.23 + 0.30\epsilon_3$	$1.31 + 0.03\epsilon_4$	$1.26 + 0.02\epsilon_5$
y_7	$4.06 + 2.39\epsilon_1$	$4.18 + 0.35\epsilon_2$	$6.53 + 0.72\epsilon_3$	$1.19 + 0.08\epsilon_4$	$1.17 + 0.01\epsilon_5$

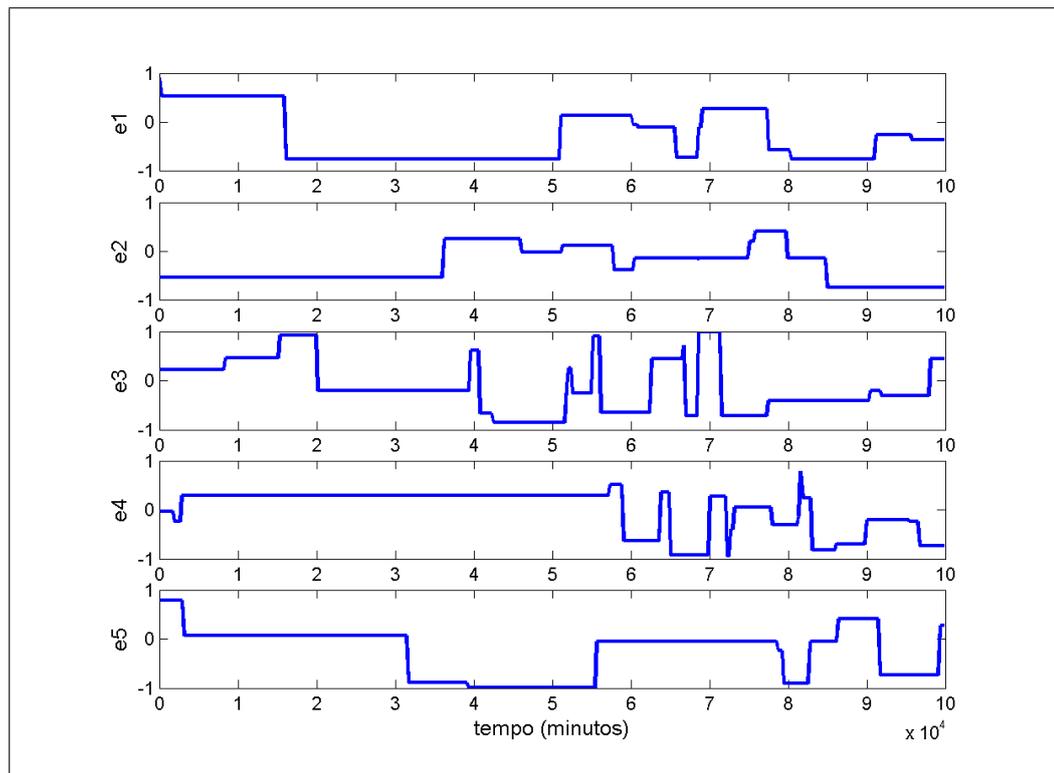


Figura 6.2: Incertezas nos ganhos ϵ_i , $i = 1, \dots, 5$.

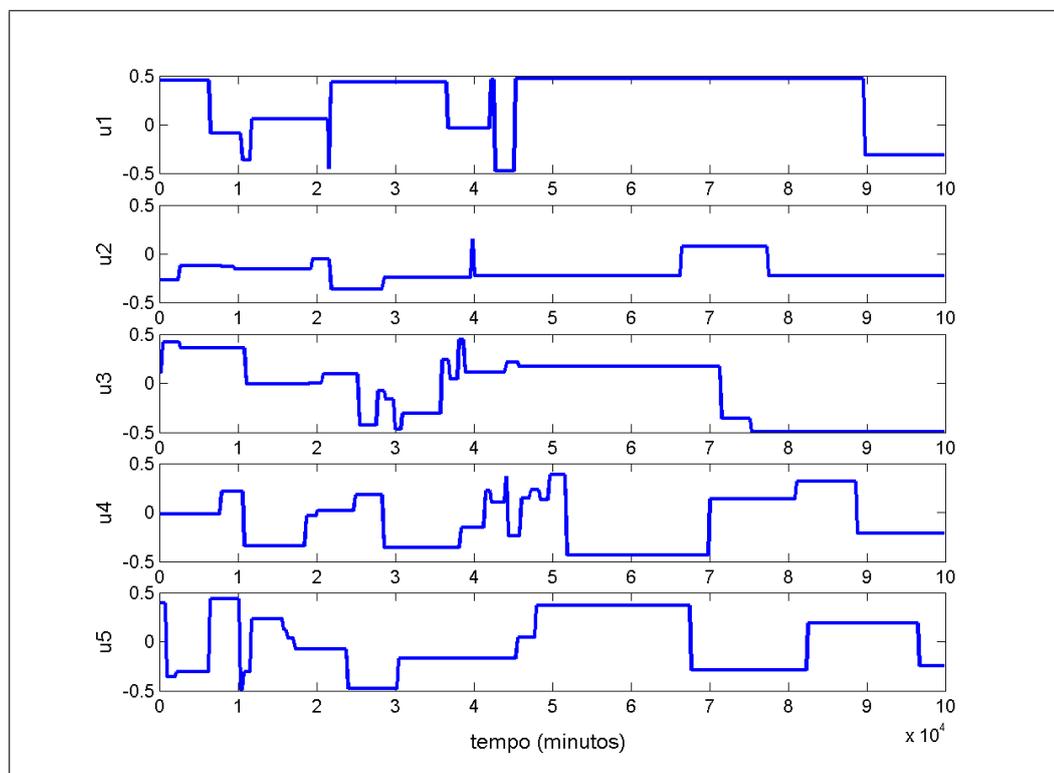


Figura 6.3: Variáveis de entrada do modelo da coluna fracionadora $\{ u_1, u_2, u_3, d_1, d_2 \}$.

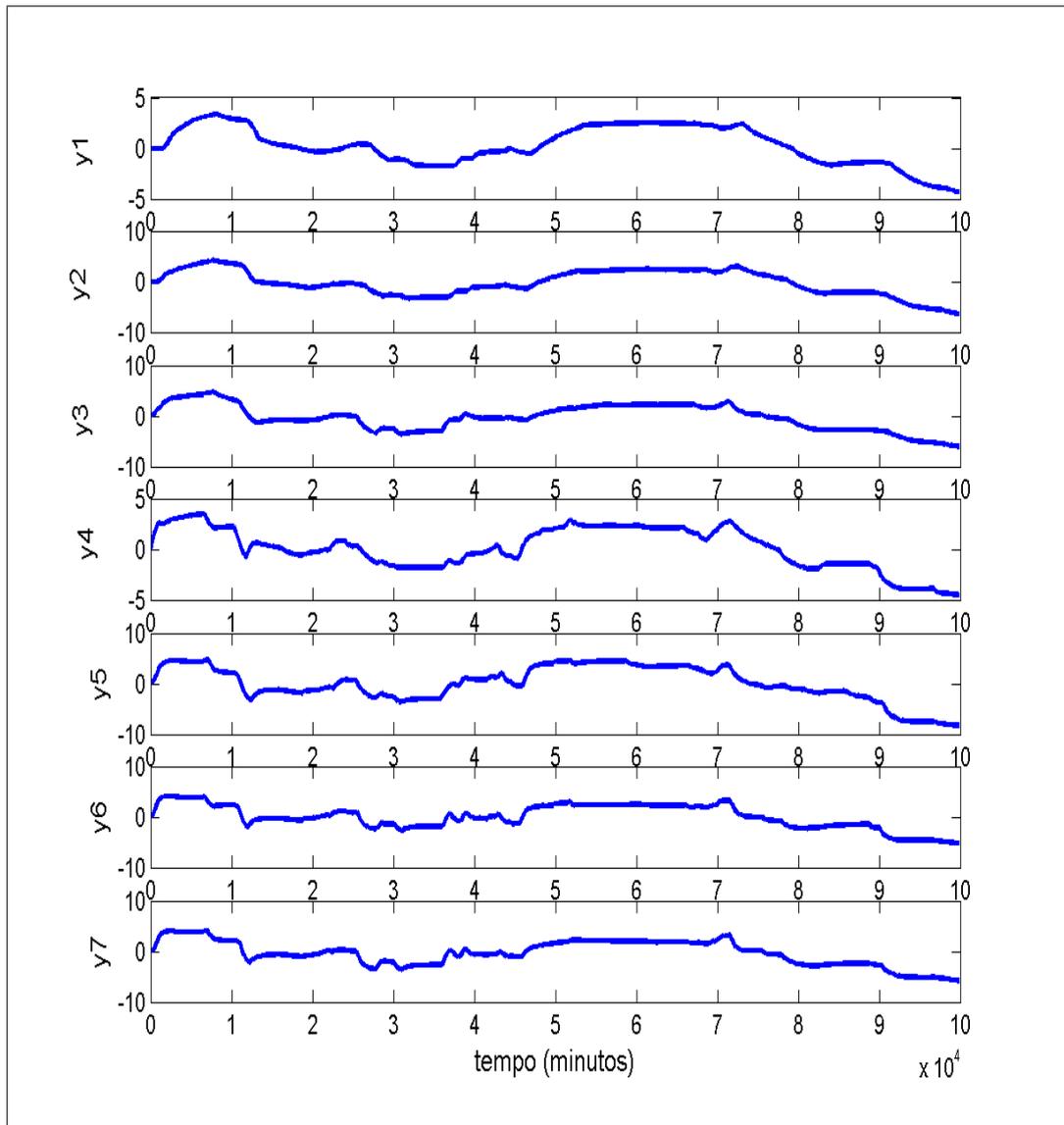


Figura 6.4: Variáveis de saída do modelo da fracionadora $\{y_1, y_2, y_3, y_4, y_5, y_6, y_7\}$.

O primeiro conjunto variáveis secundárias, CVS01, é composto pelas variáveis descrita na Tabela 6.1, exceto pelas variáveis relacionadas ao produto lateral (composição do produto lateral (y_2) e temperatura do produto lateral (y_6)). Este conjunto foi utilizado por conter todas as variáveis disponibilizadas pelo modelo da coluna fracionadora.

O segundo conjunto, denominado CVS02 é composto pelas variáveis $\{u_1, d_2, y_4\}$, numa aplicação semelhante à apresentada em Ansari & Tadé (2000), onde também foi construído um modelo inferencial para a estimação do produto de topo.

Nas próximas seções deste capítulo serão apresentados os resultados de simulação obtidos com esses dois conjuntos de variáveis secundárias.

6.2 Caso 01: Simulação com o Conjunto CVS01

Esta seção apresenta os resultados obtidos com o conjunto de variáveis secundárias CVS01 para a inferência do produto de topo da coluna fracionadora de óleo cru. As variáveis que compõem este conjunto estão descritas na Tabela 6.4.

Tabela 6.4: Descrição das variáveis secundárias que compõem o conjunto CVS01 (coluna fracionadora).

Variável	Descrição
u_1	vazão do produto de topo
u_2	vazão do produto na lateral
u_3	refluxo no fundo da coluna
d_1	refluxo intermediário
d_2	refluxo no topo da coluna
y_3	temperatura do refluxo no fundo da coluna
y_4	temperatura no topo da coluna
y_5	temperatura do refluxo no topo da coluna
y_7	temperatura do refluxo intermediário

O procedimento de seleção de variáveis de entrada foi aplicado sobre esse conjunto de variáveis secundárias. Os valores, em módulo, do coeficiente de correlação entre cada variável secundária e a variável de saída (composição do produto de topo (y_1)) são ilustrados pela Figura 6.5 e Tabela 6.5.

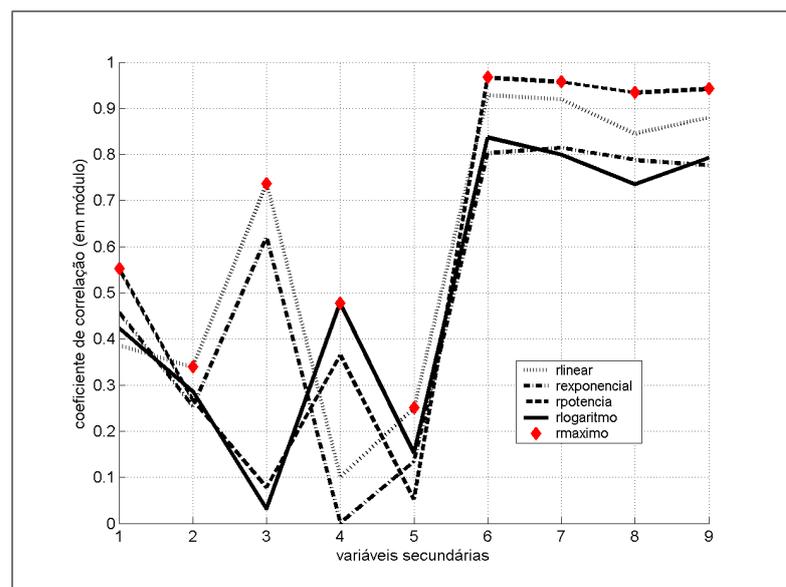


Figura 6.5: Cálculo da correlação para o Caso 01 (coluna fracionadora).

Tabela 6.5: Classificação das variáveis de acordo com o valor absoluto da correlação para o Caso 01 (coluna fracionadora).

posição	1º	2º	3º	4º	5º	6º	7º	8º	9º
variável	y_3	y_4	y_7	y_5	u_3	u_1	d_1	u_2	d_2
valor absoluto de correlação	0.9662	0.9567	0.9417	0.9332	0.7523	0.5543	0.4799	0.3298	0.2514

Conforme esses valores de coeficiente de correlação, os subconjuntos de variáveis secundárias são criados, como pode ser visto na Tabela 6.6.

Tabela 6.6: Especificação dos subconjuntos de variáveis secundárias para o Caso 01 (coluna fracionadora).

Subconjuntos	Variáveis
c_1	y_3, y_1
c_2	y_3, y_4, y_1
c_3	y_3, y_4, y_7, y_1
c_4	y_3, y_4, y_7, y_5, y_1
c_5	$y_3, y_4, y_7, y_5, u_3, y_1$
c_6	$y_3, y_4, y_7, y_5, u_3, u_1, y_1$
c_7	$y_3, y_4, y_7, y_5, u_3, u_1, d_1, y_1$
c_8	$y_3, y_4, y_7, y_5, u_3, u_1, d_1, u_2, y_1$
c_9	$y_3, y_4, y_7, y_5, u_3, u_1, d_1, u_2, d_2, y_1$

Esses subconjuntos foram avaliados e o resultado da seleção do subconjunto mais apropriado é ilustrado pela Figura 6.6, onde é indicado que o subconjunto selecionado foi o c_4 (composto pelas quatro variáveis secundárias referentes às medidas de temperatura).

A partir do conjunto de variáveis de entrada selecionado, a estrutura do modelo *fuzzy* é determinada, ou seja, o conjunto de regressores que será utilizado para identificar o modelo *fuzzy*. Consideraram-se atrasos de zero a cinco ($lag = 0, \dots, lag = 5$) e o resultado dessa seleção é ilustrado pela Figura 6.7. Novamente, para reduzir o esforço computacional, considerou-se o atraso de y_1 igual ao atraso das variáveis de entrada e o atraso selecionado foi igual a 1.

Seja $y_1(t)$ a composição do produto de topo no instante t , x_1, \dots, x_4 as variáveis de entrada selecionadas anteriormente, e x_5 a composição do produto de topo utilizada como realimentação do modelo; o modelo *fuzzy* pode ser representado por uma função $g(\cdot)$, como na equação (6.3), considerando-se o atraso igual a 1:

$$\hat{y}_1(t) = g(x_1(t-1), \dots, x_4(t-1), x_5(t-1)). \quad (6.3)$$

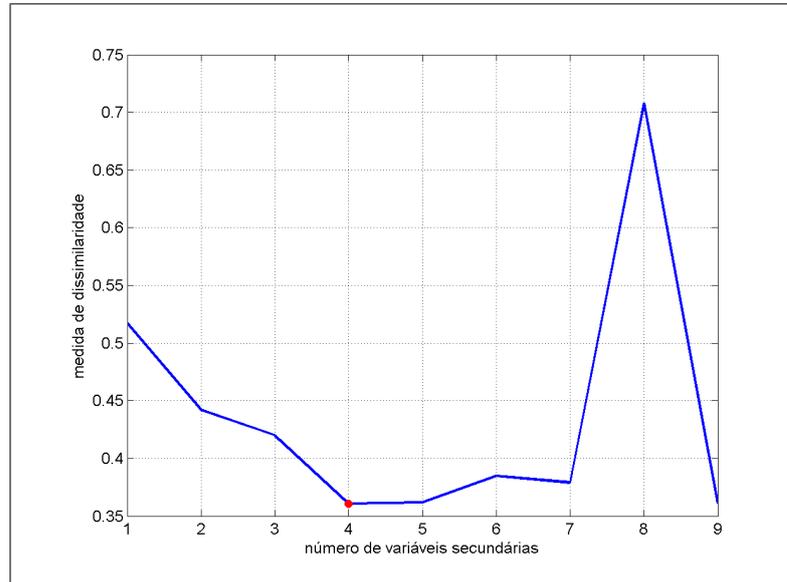


Figura 6.6: Seleção de variáveis para o Caso 01 (coluna fracionadora).

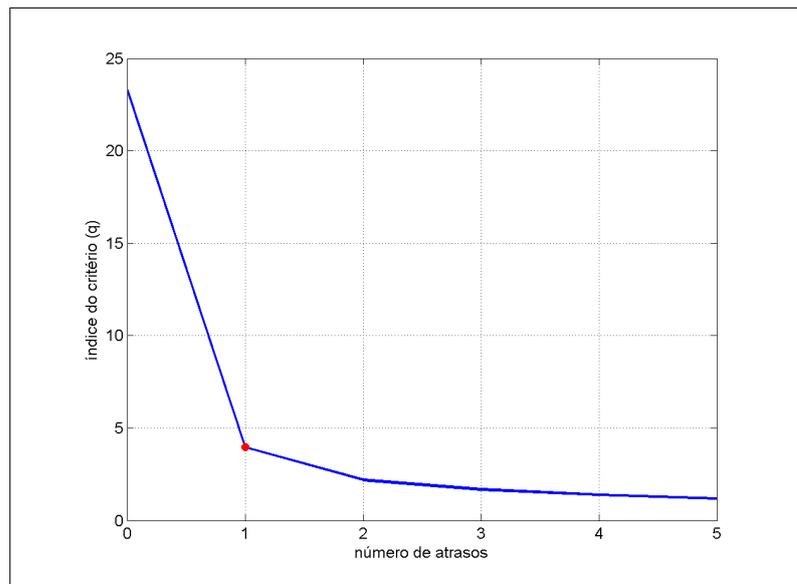


Figura 6.7: Seleção do atraso para o Caso 01 (coluna fracionadora).

Definidas as variáveis de entrada e a estrutura do modelo *fuzzy*, a base de regras inicial foi gerada de acordo com o algoritmo descrito pela Figura 4.9, onde o algoritmo *FCM* (Figura 2.12) foi aplicado ao espaço de entrada e saída.

O conjunto de dados amostrados é composto por 410 pares de amostras de entrada e saída. Levando-se em consideração o atraso selecionado ($lag=1$), o conjunto disponível foi reduzido para 409. Assim como no caso da coluna binária, este conjunto foi dividido em três partes iguais: conjunto de treinamento, conjunto de validação e conjunto de teste. O tamanho da base de regras inicial foi estipulado como sendo a metade do tamanho do conjunto de treinamento (base de regras inicial = 68 regras).

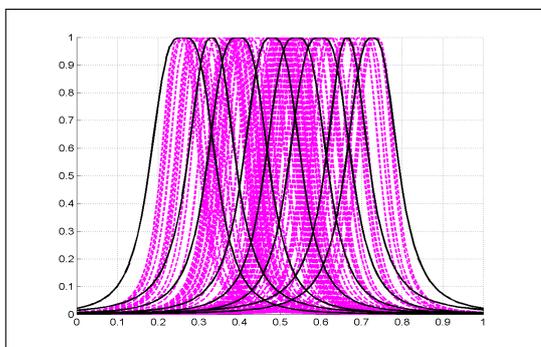
Após a aplicação do algoritmo *FCM*, os *clusters* foram projetados no domínio de cada variável e as funções paramétricas dos conjuntos *fuzzy* foram estabelecidas. Em cada domínio, foi realizada a simplificação agrupando-se as funções de pertinência similares.

A Tabela 6.7 apresenta as variáveis de entrada selecionadas e o número de funções de pertinência de descreve cada uma delas. Novamente, o tamanho da base de regras não sofreu grandes alterações contudo, o número de funções de pertinência para cada variável reduziu drasticamente.

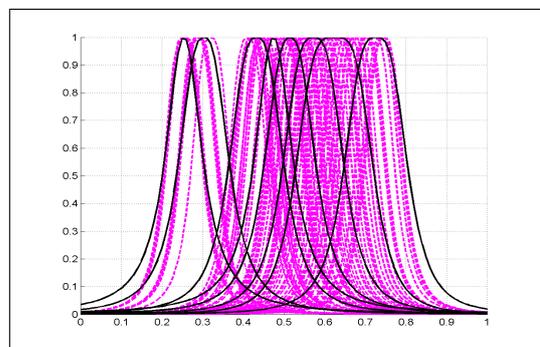
Tabela 6.7: Variáveis selecionadas para o Caso 01 (coluna fracionadora).

#	Variável	Nº de funções de pertinência
x_1	temp. do refluxo no fundo da coluna (y_3)	8
x_2	temp. no topo da coluna (y_4)	8
x_3	temp. do refluxo intermediário (y_7)	9
x_4	temp. do produto lateral (y_5)	6
x_5	composição do produto de topo (realimentação) (y_1)	8
y	composição do produto de topo (y_1)	8

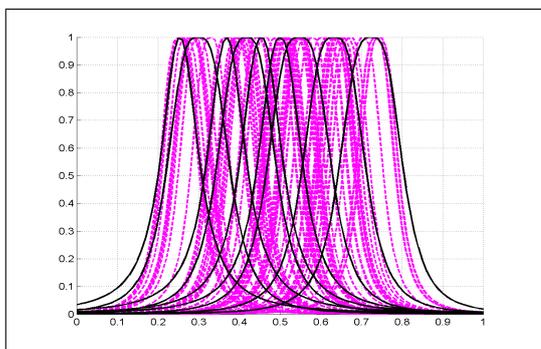
A Figura 6.8 contém a partição do domínio de todas as variáveis do modelo *fuzzy*. Nesta figura é possível observar a partição do domínio antes (linhas tracejadas) e após (linhas contínuas) da fase de simplificação da base de regras.



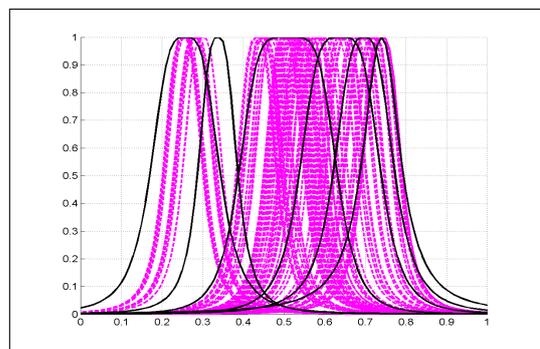
(a) Temp. do refluxo no fundo da coluna (y_3).



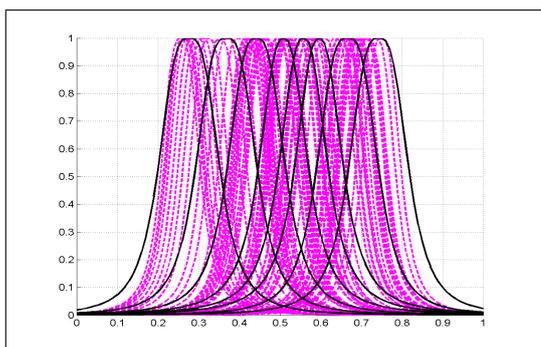
(b) Temp. no topo da coluna (y_4).



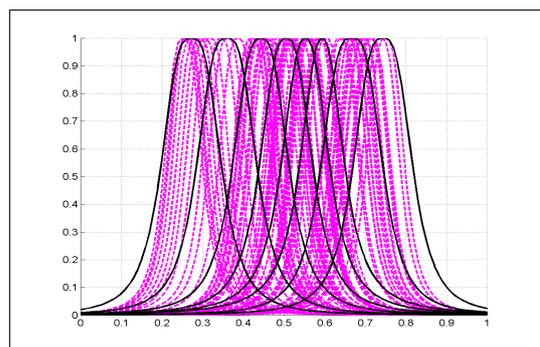
(c) Temp. do refluxo intermediário (y_7).



(d) Temp. do produto lateral (y_5).



(e) Composição do produto de topo (y_1 - realimentação).



(f) Composição do produto de topo (y_1)

Figura 6.8: Funções de pertinência das variáveis do modelo fuzzy inferencial para o Caso 01 (coluna fracionadora).

A Figura 6.9 ilustra a comparação entre a saída do modelo do processo (saída simulada) e a saída do modelo inferencial. A Tabela 6.8 apresenta as informações relevantes sobre o modelo *fuzzy* inferencial obtido para estimar a composição do produto de topo da coluna fracionadora, tais como o tamanho da base de regras, o número de variáveis do modelo e os operadores utilizados no módulo de inferência *fuzzy*.

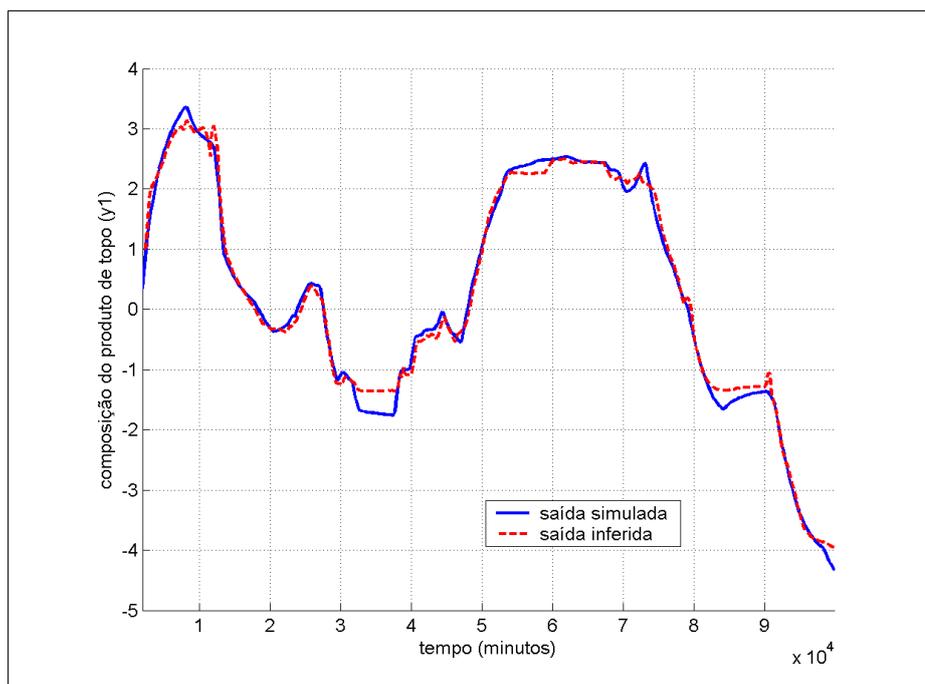


Figura 6.9: Inferência da composição do produto de topo para o Caso 01 (coluna fracionadora).

Tabela 6.8: Parâmetros do modelo *fuzzy* inferencial obtido para o Caso 01 (coluna fracionadora).

Descrição	valor
tamanho da base de regras inicial (<i>ncr</i>)	68
tamanho da base de regras simplificada	47
número de regras inconsistentes encontradas	0
número de variáveis de entrada	5
número de variáveis de saída	1
<i>lag</i> selecionado	1
operador de agregação dos antecedentes	t-norma mínimo
operador de implicação	t-norma produto
operador de agregação das regras	s-norma máximo
operador de defuzificação	<i>BOA</i>

A média dos dados referentes à composição do produto de topo (y_1) é próxima de zero (média(y_1) = 0.2740), fazendo com que a métrica *mape* apresente valores elevados, apesar da saída estimada ser visivelmente próxima à saída simulada. Neste caso, mais uma métrica será considerada a fim de verificar o desempenho do modelo *fuzzy* inferencial. Esta métrica adicional é a média do erro absoluto (*mae* - *mean absolute error*), calculado em módulo, como na equação (6.4).

$$mae = \frac{1}{N} \sum_{t=1}^N |y(t) - y_f(t)|; \quad (6.4)$$

onde N é o número de amostras consideradas, $y(t)$ é a saída simulada no instante t e $y_f(t)$ é a saída inferida (saída do modelo inferencial *fuzzy*) no instante t .

A Tabela 6.9 apresenta os índices de desempenho do modelo. A proximidade dos valores dos erros obtidos com os dados de treinamento e os dados de teste indicam que o modelo obtido apresenta uma boa capacidade de generalização.

Tabela 6.9: Índices de desempenho do modelo inferencial obtido para o Caso 01 (coluna fracionadora).

	treinamento	validação	teste
<i>mse</i>	0.0231	0.0279	0.0264
<i>mape</i>	12.5571%	37.7716%	26.9242%
<i>mae</i>	0.1200	0.1310	0.1248

A Figura 6.10 apresenta o erro absoluto, em cada instante de tempo, entre a saída simulada e a saída inferida. Nota-se que os valores de erro absoluto são baixos e ficam entre -0.5 e 0.5 . A Figura 6.11 apresenta a comparação entre a saída simulada e a saída inferida com o erro absoluto ocorrido em cada instante de tempo considerado.

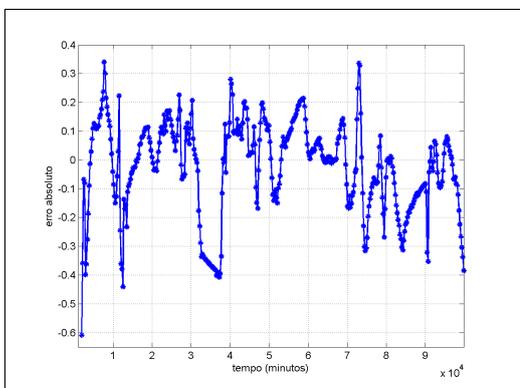


Figura 6.10: Erro absoluto entre a saída simulada e a saída inferida.

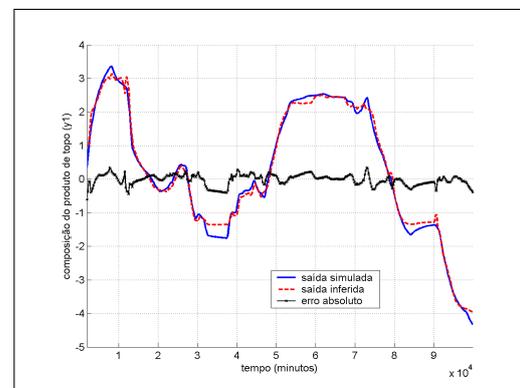


Figura 6.11: Comparação das saídas com o erro absoluto.

É interessante notar que o conjunto selecionado de variáveis de entrada é composto por todas as variáveis de temperatura. A maioria das aplicações, onde o objetivo é inferir a composição de algum produto de coluna de destilação, utiliza medidas de temperatura (Mejdell & Skogestad, 1991b; Kano et al., 2000; Barolo et al., 2002). Muitas vezes estipula-se como variáveis de entrada a medida de temperatura de determinados pratos, ou empregam-se todas as medidas de temperatura aplicando-se algum método de regressão, tal como *PLS*. Segundo Mejdell & Skogestad (1991b), este fato deve-se à semelhança nas respostas dinâmicas das temperaturas e da composição do produto da coluna.

6.3 Caso 02: Simulação com o conjunto CVS02

Em Ansari & Tadé (2000) é construído um modelo inferencial para prever os valores de do produto de topo baseado nas medidas disponíveis (refluxo no topo da coluna (d_2), temperatura no topo da coluna (y_4), vazão do produto de topo (u_1), composição do produto de topo obtido de um analisador e pressão). Foi assumido que as perturbações podem ser estimadas perfeitamente no estado estacionário.

A predição do produto de topo é baseada na seguinte correlação que pode ser derivada do modelo no estado estacionário da destilação. Essa correlação é dada pela equação (6.5):

$$CPT = [P, T, \Lambda_1, R/D] + \Lambda_2, \quad (6.5)$$

onde CPT é a composição do produto de topo, P é a pressão na coluna, T é a temperatura no topo da coluna (y_4), Λ_1 é um filtro de primeira ordem para reduzir o ruído de medida, Λ_2 é o *bias* para atualizar o modelo a partir dos resultados do analisador e R/D é a razão entre o refluxo no topo da coluna (d_2) e a vazão do produto de topo (u_1).

É importante notar que a equação (6.5) é não-linear nos termos da razão entre o refluxo no topo da coluna e a vazão do produto de topo (d_2/u_1). Segundo os autores, a vantagem de se utilizar um modelo não-linear é que este é válido para um amplo intervalo de operação e reduz o esforço gasto com a re-sintonia.

Baseando-se nesta idéia, um modelo foi gerado utilizando-se essas mesmas variáveis de entrada, exceto pela variável de pressão, cujos valores não são disponíveis. Desta forma, para este caso o procedimento de seleção de variáveis de entrada não é utilizado, sendo o procedimento de seleção da estrutura do modelo empregado dire-

tamente com as variáveis y_4 (temperatura no topo da coluna) e d_2/u_1 (razão entre o refluxo no topo da coluna e a vazão do produto de topo).

Repetindo a seqüência de procedimentos da metodologia proposta, a estrutura do modelo foi selecionada. Foram considerados atrasos de zero a cinco ($lag = 0, \dots, lag = 5$) e o resultado da seleção da estrutura do modelo é ilustrado pela Figura 6.12. Novamente, para reduzir o esforço computacional, considerou-se o atraso de y_1 igual ao atraso das variáveis de entrada. O atraso selecionado foi igual a 1.

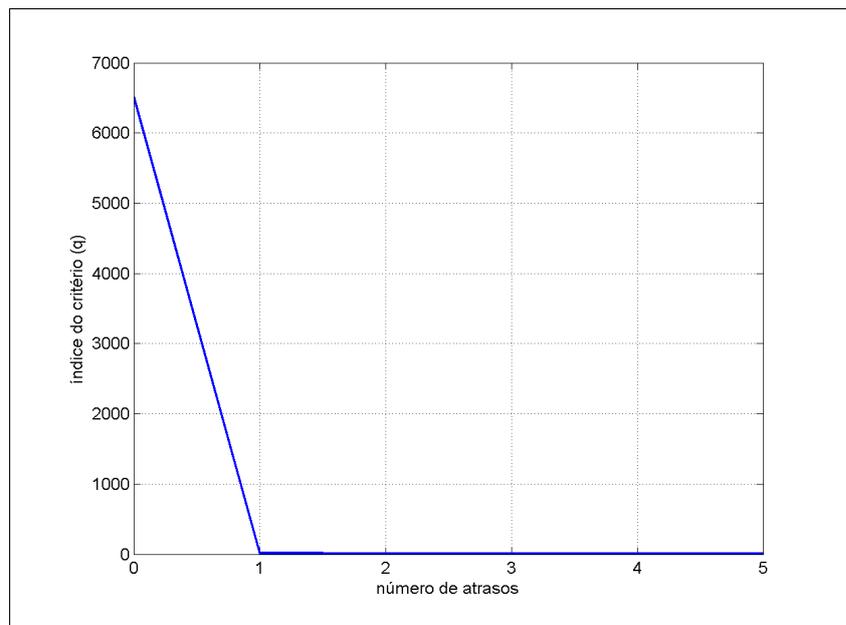


Figura 6.12: Seleção do atraso para o Caso 02 (coluna fracionadora).

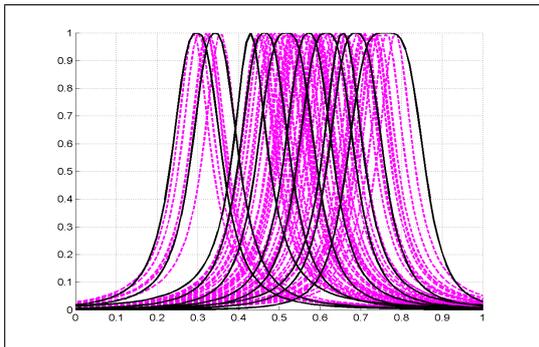
Seja $y_1(t)$ a composição do produto de topo no instante t , x_1 , x_2 as variáveis de entrada, e x_3 a composição do produto de topo utilizada como realimentação do modelo, o modelo *fuzzy* pode ser representado por uma função $g(\cdot)$, como na equação (6.6), considerando-se o atraso igual a 1:

$$\hat{y}_1(t) = g(x_1(t-1), x_2(t-1), x_3(t-1)). \quad (6.6)$$

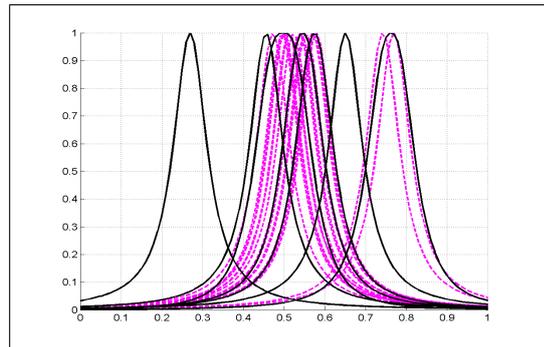
Definidas as variáveis de entrada e a estrutura do modelo *fuzzy*, a base de regras inicial foi gerada de acordo com o algoritmo descrito pela Figura 4.9, onde o algoritmo *FCM* foi aplicado ao espaço de entrada e saída. Assim como no primeiro caso deste capítulo, o conjunto de dados amostrados é composto por 410 pares de amostras de entrada e saída. Levando-se em consideração o atraso selecionado ($lag=1$), o conjunto disponível foi reduzido para 409. Novamente, este conjunto foi dividido em três partes iguais: conjunto de treinamento, conjunto de validação e conjunto de teste. O tamanho da base de regras inicial foi estipulado como sendo a metade do

tamanho do conjunto de treinamento (base de regras inicial = 68 regras).

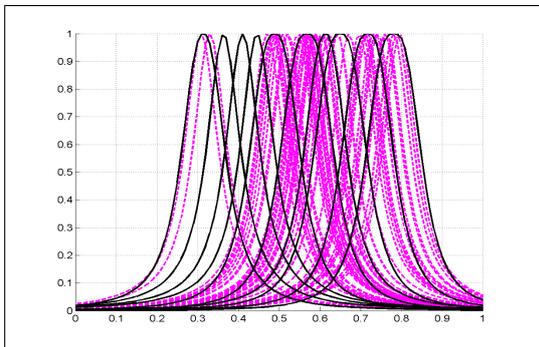
Após a aplicação do algoritmo *FCM*, os *clusters* foram projetados no domínio de cada variável e as funções paramétricas dos conjuntos *fuzzy* foram estabelecidas. Em cada domínio, foi realizada a simplificação agrupando-se as funções de pertinência similares. A Figura 6.13 contém a partição do domínio de todas as variáveis do modelo *fuzzy*. Nesta figura é possível observar a partição do domínio antes (linhas tracejadas) e após (linhas contínuas) da fase de simplificação da base de regras.



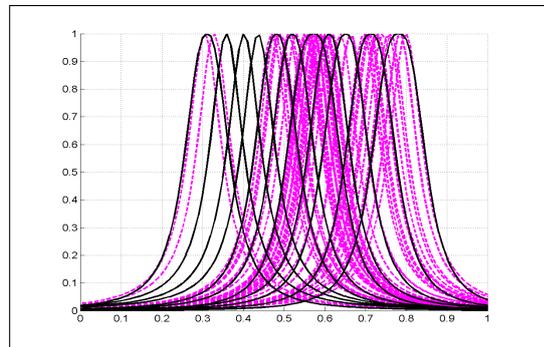
(a) Temp. no topo da coluna (y_4).



(b) Razão entre refluxo no topo da coluna e a vazão do produto de topo (d_2/u_1).



(c) Composição do produto de topo (y_1 - realimentação).



(d) Composição do produto de topo (y_1)

Figura 6.13: Funções de pertinência das variáveis do modelo *fuzzy* inferencial para o Caso 02 (coluna fracionadora).

A Tabela 6.10 apresenta as variáveis de entrada e o número de funções de pertinência que descreve cada uma delas. Novamente, o tamanho da base de regras não sofreu grandes alterações contudo, o número de funções de pertinência para cada variável foi reduzido drasticamente.

As informações relevantes sobre o modelo *fuzzy* inferencial obtido para estimar a composição do produto de topo da coluna fracionadora são apresentados na Tabela 6.11, tais como o tamanho da base de regras, o número de variáveis do modelo e os operadores utilizados no módulo de inferência *fuzzy*.

Tabela 6.10: Variáveis selecionadas para o Caso 02 (coluna fracionadora).

#	Variável	Nº de funções de pertinência
x_1	temperatura no topo da coluna (y_4)	10
x_2	razão entre o refluxo no topo da coluna e a vazão do produto de topo (d_2/u_1)	7
x_3	composição do produto de topo (realimentação) (y_1)	10
y	composição do produto de topo (y_1)	11

Tabela 6.11: Parâmetros do modelo fuzzy inferencial obtido para o Caso 02 (coluna fracionadora).

Descrição	valor
tamanho da base de regras inicial (ncr)	68
tamanho da base de regras simplificada	40
número de regras inconsistentes encontradas	0
número de variáveis de entrada	3
número de variáveis de saída	1
<i>lag</i> selecionado	1
operador de agregação dos antecedentes	t-norma min
operador de projeção	t-norma <i>prod</i>
operador de agregação das regras	s-norma max
operador de defuzificação	<i>boa</i>

Os índices de desempenho do modelo obtido para o caso 02 são apresentados pela Tabela 6.12. A proximidade dos valores dos erros obtidos com os dados de treinamento e os dados de teste indicam que o modelo obtido apresenta uma boa capacidade de generalização. Neste caso também foi calculado a média do erro absoluto (*mae*), tendo em vista que ocorreu o mesmo problema com a métrica *mape*, que gerou valores altos para esta métrica.

Tabela 6.12: Índices de desempenho do modelo inferencial obtido para o Caso 01 (coluna fracionadora).

	treinamento	validação	teste
<i>mse</i>	0.0231	0.0279	0.0264
<i>mape</i>	12.5571%	37.7716%	26.9242%
<i>mae</i>	0.1200	0.1310	0.1248

A Figura 6.14 ilustra a comparação entre a saída do modelo do processo (saída simulada) e a saída do modelo inferencial.

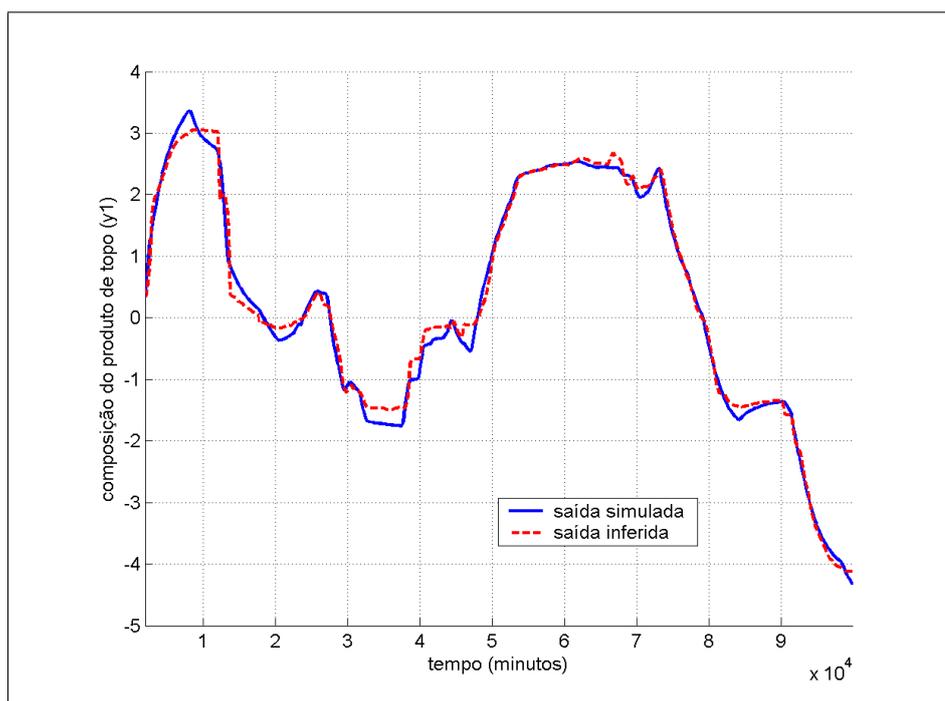


Figura 6.14: Comparação entre a saída do modelo fuzzy inferencial e a saída desejada para o Caso 02 (coluna fracionadora).

A Figura 6.15 apresenta o erro absoluto, em cada instante de tempo, entre a saída simulada e a saída inferida. Nota-se que os valores de erro absoluto são baixos e ficam entre -0.65 e 0.40 . A Figura 6.16 apresenta a comparação entre a saída simulada e a saída inferida com o erro absoluto ocorrido em cada instante de tempo considerado.

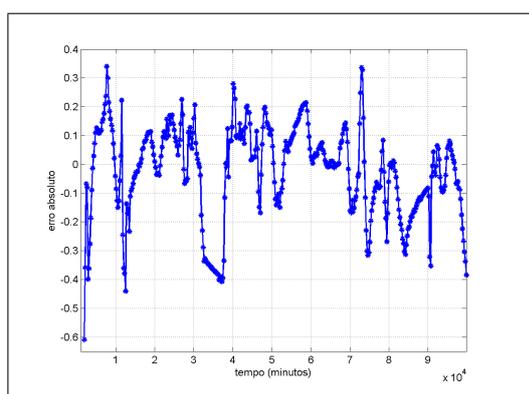


Figura 6.15: Erro absoluto entre a saída simulada e a saída inferida.

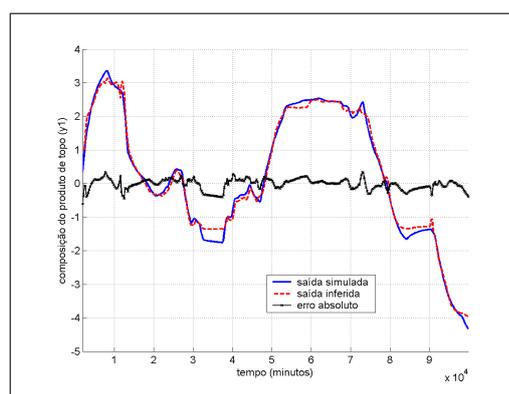


Figura 6.16: Comparação das saídas com o erro absoluto.

Em comparação aos resultados obtidos com as variáveis secundárias empregadas no caso 01 (seção 6.2), o modelo identificado para o caso 02 apresentou resultados similares, com a vantagem de utilizar um conjunto menor de variáveis de entrada,

pois o caso 01 utiliza um conjunto de entrada com cinco variáveis mais a saída atrasada, e o caso 02 utiliza somente duas variáveis de entrada mais a saída atrasada.

É possível notar que os modelos identificados com a metodologia proposta nos dois casos de simulação apresentaram resultados satisfatórios, principalmente se for considerado que, usualmente, essas medidas de composição são obtidas através de analisadores, que disponibilizam esses valores com um certo tempo de atraso.

Capítulo 7

Considerações Finais

O emprego de modelos inferenciais no controle de processos industriais, principalmente no controle preditivo, mostra-se cada vez mais indispensável, sendo necessário, portanto, o desenvolvimento de técnicas robustas e eficientes para o desenvolvimento destes modelos. Como descrito no capítulo 2, os principais tópicos que devem ser abordados durante o desenvolvimento de modelos inferenciais são: (i) aquisição dos dados; (ii) pré-processamento dos dados; (iii) seleção das variáveis secundárias; e (iv) implementação do método de inferência.

A fase de aquisição de dados pode considerar somente dados em bateladas, dados disponibilizados *on-line* ou até mesmo uma combinação dos dois tipos de dados e, de acordo com o emprego desses dados e do método de inferência, os modelos inferidos podem ser dinâmicos ou não. Uma breve discussão sobre as vantagens e desvantagens do uso de modelos dinâmicos pode ser encontrado em Kano et al. (2003).

A fase de pré-processamento de dados visa normalizar e reduzir a dimensionalidade do conjunto de dados sem que estes percam as suas características e informações importantes sobre o processo, assim como tratar *ouliers* e dados faltantes. Métodos estatísticos são usualmente empregados para reduzir a alta dimensionalidade através da análise de correlação das variáveis.

A seleção das variáveis de entrada, a partir das variáveis secundárias, pode ser feita através de conhecimento *a priori* sobre o processo, quando este for disponível, ou através de métodos sistemáticos, tais como análise de sensibilidade. Como discutido em Hocking (1976), uma análise para verificar quais variáveis secundárias são realmente relevantes na inferência das variáveis principais é de grande importância durante a identificação do modelo, pois o emprego de uma variável desnecessária além de não contribuir na inferência, pode aumentar a variância associada, prejudi-

cando o modelo gerado.

Finalmente, na fase de implementação é necessário selecionar o método de inferência que melhor se adapta às características do processo, i.e., nos casos onde o processo é linear, a maioria dos métodos existentes apresentam resultados satisfatórios. Porém, processos complexos, dinâmicos, não-lineares e com alta dimensionalidade, dificultam o emprego de técnicas lineares, sendo necessário que tais características do processo sejam levadas em consideração. Nestes casos a gama de alternativas torna-se escassa. Esta dificuldade apresenta-se como uma motivação para o desenvolvimento de técnicas não-lineares e dinâmicas para a implementação de modelos inferenciais. Tais modelos poderiam ser empregados em conjunto com um esquema de controle, gerando, desta forma, uma estratégia de controle inferencial.

Neste contexto, este trabalho propõe uma metodologia para a identificação de modelos *fuzzy* inferenciais para a construção de *soft sensors*. Esta metodologia é composta por quatro módulos.

O módulo um determina a estrutura do modelo *fuzzy* inferencial e é composto por dois procedimentos. O primeiro procedimento deste módulo seleciona o conjunto de variáveis de entrada do modelo, dentre um conjunto de variáveis secundárias disponíveis do processo a ser modelado.

Inicialmente os subconjuntos de variáveis secundárias são criados conforme os respectivos valores de coeficiente de correlação (em módulo) com relação a variável de saída. Para cada subconjunto é criado um mapa de Kohonen e após o treinamento, a qualidade desses mapas é verificada através de uma medida de dissimilaridade. A medida de dissimilaridade verifica a diferença entre dois mapas distintos e o mapa com o menor valor de dissimilaridade média, ou seja, o mapa mais similar a todos os outros mapas, contém o subconjunto mais representativo de variáveis secundárias.

O segundo procedimento do módulo um preocupa-se com a seleção dos regressores que serão utilizados na identificação do modelo *fuzzy*. O conjunto de regressores de $y(t)$ é composto pelas amostras atrasadas das variáveis de entrada e saída utilizados na identificação do modelo. Este procedimento utiliza os quocientes de Lipschitz para realizar a seleção. Através desses quocientes é possível verificar se determinado conjunto de regressores é o suficiente para a identificação do modelo *fuzzy*. A seleção dos regressores é uma fase importante e deve ser criteriosa devido à sua influência direta no tamanho da base de regras e, conseqüentemente, na complexidade do modelo *fuzzy*.

No módulo dois a base de regras inicial do modelo *fuzzy* é gerada aplicando-se um algoritmo de agrupamento *fuzzy FCM* ao espaço de entrada e saída. O objetivo

é particionar esse espaço em *ncr clusters* e, a partir desses *clusters*, obter as funções de pertinência que descrevem o universo de discurso de cada variável e a base de regras inicial do modelo *fuzzy*. Neste procedimento, destaca-se o fato de que não é estipulado o tamanho da base de regras ou o número de funções de pertinência para cada variável. Em geral, emprega-se um valor alto para *ncr* para que os *clusters* cubram todas as regiões possíveis do espaço de entrada e saída. Entretanto, esta fase é extremamente dependente da qualidade dos dados amostrados, pois a distribuição dos *clusters* é determinada pela localização dos dados no espaço de entrada e saída. Se os dados amostrados não forem suficientemente representativos do processo a ser modelado, a qualidade do modelo final fica comprometida.

A completude do modelo *fuzzy* é garantida pela constante de ponderação do algoritmo *FCM*, que determina a sobreposição dos *clusters*, porém a propriedade de completude do modelo, no caso da metodologia proposta neste trabalho, também é dependente da qualidade dos dados amostrados. Novamente, se existir alguma região no espaço de entrada e saída que não é coberta por esses dados amostrados, então o modelo *fuzzy* não será capaz de inferir uma saída apropriada para as entradas pertencentes a tal região.

A base de regras inicial é simplificada no módulo três. Neste módulo foi implementado um procedimento para simplificar a base de regras através da substituição de eventuais funções de pertinência similares por uma nova função que as represente adequadamente. No procedimento de simplificação da base de regras não é estipulado o número de funções de pertinência para cada variável, pois a quantidade de funções de pertinência necessária para descrever cada variável do modelo *fuzzy* varia conforme a distribuição dos dados amostrados. Estipula-se, neste caso, uma distância mínima (*threshold*) entre as funções de pertinência, em cada domínio, e as funções consideradas próximas (ou similares) são substituídas por uma nova função cujos parâmetros são determinados de acordo com a localização das funções similares que serão substituídas. Este procedimento destaca-se pela simplicidade, ao realizar a simplificação através da verificação dos parâmetros das funções de pertinência, e também por não pré-estabelecer o número de funções de pertinência para cada variável. Esta etapa é essencial para a manutenção da interpretabilidade do modelo *fuzzy*.

E finalizando a metodologia, o módulo quatro verifica e trata das regras inconsistentes, caso existam. Entende-se por regras inconsistentes aquelas que possuem o mesmo antecedente e conseqüentes distintos. A inconsistência de regras pode surgir devido a vários fatores, tais como: a união de funções de pertinência similares e a

constante de ponderação que determina a sobreposição dos *clusters* no algoritmo *FCM*.

É importante ressaltar que cada uma das etapas da metodologia proposta constitui um procedimento independente, que pode ser utilizado em combinação com outras técnicas para gerar novas metodologias de modelagem, identificação e controle *fuzzy*, ou não.

Na maioria das aplicações, onde o processo apresenta alta dimensionalidade, empregam-se métodos estatísticos, tais como *PLS* e *PCA* para diminuir a dimensão do espaço de entrada, onde as variáveis de entrada (variáveis independentes) são transformadas de modo a formar as variáveis latentes, que são uma combinação linear das variáveis de entrada. Uma desvantagem dessa abordagem é que esta combinação de variáveis prejudica a interpretabilidade do modelo gerado, pois torna-se difícil a análise do comportamento do processo através do modelo gerado. Além disso, esses métodos não são dinâmicos, pois não consideram as amostras atrasadas das variáveis de entrada e saída do processo. Portanto, ao selecionar somente as variáveis secundárias mais adequadas para integrar o conjunto de variáveis de entrada contribui-se não somente para gerar um modelo *fuzzy* dinâmico e mais simples, pois o número de regras necessárias para descrever o modelo cresce exponencialmente com o número de entradas (Drogies, 1996), como também na interpretabilidade do modelo *fuzzy*.

A geração da base de regras inicial, descrita pelo algoritmo da Figura 4.9, pode ser usada para a obtenção de um modelo *fuzzy* lingüístico a partir de um conjunto de dados qualquer, numa abordagem de identificação do tipo caixa-preta.

A simplificação da base de regras inicial é uma fase importante durante o processo de identificação de modelos *fuzzy* inferenciais, pois o compromisso entre a precisão e a generalização do modelo são considerados durante esta fase. O parâmetro ζ_ω (inequação (4.20)), que define a diferença mínima entre os centros dos *clusters*, controla, indiretamente, o número de funções de pertinência das variáveis de modelo *fuzzy*. Desta forma, é possível aumentar a precisão do modelo *fuzzy* ao diminuir o valor de ζ_ω . A diminuição do valor de ζ_ω provoca o aumento do número de funções de pertinência das variáveis do modelo e, provavelmente, provoca também o aumento do tamanho da base de regras. Portanto, o compromisso entre a precisão e a transparência do modelo depende da escolha do valor do parâmetro ζ_ω . Em Wang & Mendel (1991), foram realizadas simulações mostrando que à medida que se aumentava o número de funções de pertinência das variáveis, a precisão do modelo *fuzzy* tornava-se cada vez maior, a um custo de uma base de regras muito grande.

Grande parte das propostas para a simplificação da base de regras, através da redução da redundância na descrição do universo de discurso das variáveis, baseia-se na medida de similaridade, como descrita em Babůska et al. (1998). Não obstante a efetividade dessa abordagem, uma de suas desvantagens reside no fato de que, para verificar a similiaridade entre duas funções de pertinência, é necessário calcular o grau de pertinência de todos os pontos do universo de discurso em relação a essas funções de pertinência. Uma vez que duas funções são consideradas similares, elas são substituídas por uma nova função capaz de representá-las adequadamente. Então o procedimento se repete até que não haja mais similaridade de funções na base de regras. A carga computacional, e a provável lentidão desse procedimento, aumenta de acordo com o número de variáveis do modelo *fuzzy* e também da quantidade de funções de pertinências que descreve o domínio de cada variável.

Observando essa dificuldade, este trabalho apresenta um algoritmo alternativo para verificar a similaridade entre funções de pertinência de forma simplificada, realizada diretamente através da comparação dos parâmetros das funções em questão. Os procedimentos propostos para a simplificação da base de regras e para o tratamento de regras inconsistentes podem ser aplicados a qualquer modelo *fuzzy* lingüístico.

Durante o procedimento de identificação verificou-se que, devido às características inerentes aos sistemas *fuzzy* (que tendem a suavizar as saídas) e também ao algoritmo de agrupamento *FCM*, os dados próximos aos limites inferior ou superior não eram devidamente representados pelo modelo *fuzzy* final. Numa tentativa de superar essa deficiência, o algoritmo de agrupamento Gustafson-Kessel (Gustafson & Kessel, 1979) foi utilizado em substituição ao algoritmo *FCM*, pois o formato elipsoidal dos *clusters* gerados pelo algoritmo Gustafson-Kessel poderia gerar representações mais adequadas da localização dos pares de dados. Porém, a projeção dos *clusters* no domínio de cada variável, utilizando-se o método de projeção e obtenção dos parâmetros das funções de pertinência proposto neste trabalho, gerou funções de pertinência que cobriam todo o universo de discurso, muitas vezes uma variável era representada por somente uma função de pertinência. Este tipo de problema ocorreu justamente devido ao formato desses *clusters*. Desta forma, optou-se por manter o uso do algoritmo *FCM*.

Ainda sobre a questão da deficiência na representação dos dados localizados próximos aos limites inferior e superior, para gerar o modelo, o algoritmo *FCM* é aplicado ao conjunto de treinamento com um determinado número de *clusters*. Como resultado, obtém-se a localização dos *clusters* no espaço de entrada e saída. Esta localização depende da distribuição dos dados no espaço de entrada e saída,

e o centro desses *clusters* localizam-se onde há maior concentração de dados. Este fato prejudica a representação dos dados localizados nos extremos do universo de discurso. Para resolver esse problema, o universo de discurso de cada variável deve ser estendido de forma que os dados fiquem mais centralizados. Este artifício melhorou consideravelmente o desempenho dos modelos obtidos.

A metodologia proposta foi utilizada na construção de modelos inferenciais para a estimação de medidas de composição de produtos em processos petroquímicos simulados. Os resultados obtidos mostraram que esta metodologia é capaz de gerar estimativas aproximadas das saídas simuladas. Porém, observou-se que os modelos gerados através do algoritmo de agrupamento *FCM* apresentam deficiências para inferir as saídas próximas aos limites do universo de discurso, como discutido anteriormente, devido ao fato do algoritmo *FCM* enfatizar os pares de dados próximos aos centros dos *clusters* e, portanto, os dados longe desses centros não são tratados de forma adequada.

No caso da coluna de destilação binária, descrita no capítulo 5, foram realizados estudos comparativos, um para o caso do produto de topo e outro para o caso do produto de fundo. Nesses dois casos foram implementados dois modelos estatísticos, baseados nos resultados da *PCA*, através da *PCR*, e um modelo neural. Para o caso do produto de topo, o estimador neural proposto em Fabro (2003) apresentou as estimativas mais precisas, seguido do modelo *fuzzy* gerado com a metodologia proposta neste trabalho. Os modelos estatísticos apresentaram os piores resultados. Mesmo que o número de funções de pertinência das variáveis do modelo seja relativamente alto, assim como o tamanho da base de regras, o modelo *fuzzy* ainda apresenta a vantagem da interpretabilidade, onde o comportamento do processo pode ser compreendido através das regras *fuzzy*, o que não ocorre com o estimador neural.

Para o caso da composição do produto de fundo, o modelo *fuzzy* apresentou resultados inferiores aos modelos estatísticos e ao modelo neural. Este fraco desempenho do modelo *fuzzy* pode ser resultante da baixa quantidade de dados amostrados disponíveis para realizar a identificação do modelo *fuzzy*. Como dito anteriormente, a identificação, realizada através de um algoritmo de agrupamento, é fortemente dependente da qualidade e quantidade dos dados amostrados, que neste caso, não foram suficientes para gerar um modelo *fuzzy* adequado.

7.1 Contribuições

As contribuições resultantes do desenvolvimento deste trabalho podem ser enumeradas como segue:

7.1.1 Investigação do uso de modelos *fuzzy* lingüísticos como modelos inferenciais, pois na maioria das aplicações utilizam-se modelos *fuzzy* Takagi-Sugeno ou redes neurais para este tipo de aplicação;

7.1.2 Desenvolvimento de um método baseado em mapas de Kohonen para a seleção de variáveis de entrada, onde empregaram-se coeficientes de correlação para modelos exponencial, potência e logaritmo, ao invés de considerar somente o modelo linear (coeficiente de correlação linear);

7.1.3 Emprego do algoritmo de agrupamento *FCM* para gerar modelos *fuzzy* lingüísticos, pois em geral este algoritmo é utilizado para gerar modelos Takagi-Sugeno;

7.1.4 Método de projeção dos *clusters* e obtenção dos parâmetros das funções de pertinência de modo simplificado;

7.1.5 Implementação de um método simples e intuitivo para a simplificação da base de regras inicial.

7.2 Trabalhos Futuros

Alguns pontos que continuaram em aberto poderiam ser investigados, tais como:

7.2.1 Implementação de um método de ajuste dos parâmetros das funções de pertinência (σ, γ, ω da função *gbell*), a fim de melhorar a qualidade das estimativas geradas pelo modelo *fuzzy* inferencial, e também para tratar possíveis casos de incompletude do modelo;

7.2.2 Implementação de um método para a avaliação da relevância de cada regra, onde regras irrelevantes poderiam ser removidas da base e regras similares poderiam ser combinadas;

7.2.3 Realização de testes onde a metodologia fosse utilizada para a identificação de outros processos complexos, para verificar a sua aplicabilidade em processos diversos, inclusive em esquemas de controle baseados em modelo, onde o horizonte de predição também poderia ser testado.

Capítulo 8

Publicações

1. Nagai, E.Y., Arruda, L.V.R., Neves-Jr, F. (2006) Metodologia neuro-fuzzy de construção de modelos inferenciais para a estimação de medidas de concentração em processos de refino, *Rio Oil & Gas Expo Conference*, Rio de Janeiro - RJ.
2. Nagai, E.Y. & Arruda, L.V.R. (2005) Identificação de modelos inferenciais fuzzy para a estimação de medidas de concentração em processos de refino, *3o. Congresso Brasileiro de P & D em Petróleo e Gás (3o. PDPetro)*, Salvador - BA.
3. Arruda, L.V.R. & Nagai, E.Y. (2005) Um modelo neuro-fuzzy para construção de modelos inferenciais, *VII Congresso Brasileiro de Redes Neurais (VII CBRN)*, Natal - RN.
4. Nagai, E.Y. & Arruda, L.V.R. (2005) Soft sensor based on fuzzy model identification, *16th International Federation of Automatic Control World Congress (16th IFAC)*, Praga - República Tcheca.

Referências Bibliográficas

- Abdul-Wahab, S., Elkamel, A., Madhuranthakam, C. R. & Al-Otaibi, M. B. (2006). Building inferential estimators for modeling product quality in a crude oil desalting and dehydration process, *Chemical Engineering and Processing* **45**: 568–577.
- Abe, S. & Lan, M. S. (1995). A method for fuzzy rules extraction directly from numerical data and its application to pattern classification, *IEEE Transactions on Fuzzy Systems* **03**: 18–28.
- Abonyi, J., Roubos, J., Babůska, R. & Szeifert, F. (2003). Interpretable semi-mechanistic fuzzy models by clustering, OLS and FIS model reduction, in J. Casillas, O. Cordon, F. Herrera, & L. Magdalena (eds), *Interpretability Issues in Fuzzy Modeling*, Springer-Verlag, Heidelberg, pp. 221–248.
- Ansari, R. M. & Tadé, M. O. (2000). *Non-Linear Model-Based Process Control - Applications in Petroleum Refining*, Springer-Verlag, London.
- Babůska, R. (1999). An overview of fuzzy modeling and model-based fuzzy control, in H. B. Verbruggen & R. Babůska (eds), *Fuzzy Logic Control: Advances in Applications*, World Scientific, pp. 3–35.
- Babůska, R. (2002). Accuracy and transparency of fuzzy systems, in H.-J. Zimmerman, G. Tselentis, M. van Someren, & G. Dounias (eds), *Advances in computational intelligence and learning: Methods and applications*, Kluwer Academic Publishers, Boston-USA, pp. 3–16.
- Babůska, R., Bovarie, S., Bross, M., Passaquay, D. & Titli, A. (2000). Performance evaluation of a fuzzy rule base for control purpose, *Proceedings of 9th IEEE Conference on Fuzzy Systems*, San Antonio - USA, pp. 423–428.
- Babůska, R., Kaymak, U., van Nauta, H. R. & Setnes, M. (1998). Similarity measures in fuzzy rule base simplification, *IEEE Transactions on Systems, Man, and Cybernetics - Part B* **28**(03): 376–386.

- Babüska, R. & Vebruggen, H. B. (1996). An overview of fuzzy modeling for control, *Control Eng. Practice* **04**(11): 1593–1606.
- Balakrishnan, A. V. (1987). *Kalman Filtering Theory*, Optimization Software Inc., Publication Division.
- Barolo, M., Seborg, D. E. & Zamprogna, E. (2002). Development of a soft sensor for a batch distillation column using linear and nonlinear PLS regression techniques, *15th Triennial World Congress of the International Federation of Automatic Control*, Barcelona.
- Besley, D., Kuh, E. & Welsh, R. (1980). *Regression Diagnostic: Identifying Influential Data and Sources of Collinearity*, Wiley, New York.
- Bezdek, J. C. (1987). *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, NY.
- Bezdek, J. C. & Pal, N. R. (1995). On cluster validity for the fuzzy c-means model, *IEEE Transactions on Fuzzy Systems* **03**(03): 370–378.
- Bhartiya, S. & Whiteley, J. R. (2001). Development of inferential measurements using neural networks, *ISA Transactions* **40**: 307–323.
- Carpenter, G. A., Grossberg, S. & Rosen, D. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, *Neural Networks* **3**.
- Chen, C. T. (1984). *Linear Systems Theory and Design*, Holt-Sauders Interneticon Editions, New York, USA.
- Cleveland, W. S. & Devlin, S. J. (1988). Locally weighted regression: an approach to regression analysis by local fitting, *Journal of American Statistics Association* **83**: 596–610.
- Dam, M. & Saraf, D. N. (2006). Design of neural network using genetic algorithm for on-line property estimation of crude fractionator products, *Computers and Chemical Engineering* **30**: 722–729.
- Dayal, B., MacGregor, J. F., Taylor, P., Kildaw, R. & Marcikic, S. (1992). Application of artificial neural networks and partial least squares regression for modeling kappa number in a continuous Kamyr digester, *Proc. of Control Systems - Dreams vs. Reality*, pp. 191–196.

- Dayal, B. S. & MacGregort, J. F. (1997). Recursive exponentially weighted PLS and its applications to adaptive control and prediction, *J. Proc. Cont.* **07**(03): 169–179.
- Delgado, M. R., Zuben, F. V. & Gomide, F. (2004). Coevolutionary genetic fuzzy systems: a hierarquical collaborative approach, *Fuzzy Sets and Systems* **141**: 89–106.
- Doebelin, E. O. (1990). *Measurement Systems: Application and Design*, 4ed edn, McGraw-Hill.
- Dorsey, A. W. & Lee, J. H. (2003). Building inferential models of batch processes using subspace identification, *Journal of Process Control* **13**(5): 397–406.
- Downs, J. J. & Vogel, E. F. (1993). A plantwide industrial process control problem, *Computers and Chemical Engineering* **17**(3): 245–255.
- Drogies, S. (1996). Model predictive control with fuzzy models: studies in simulation, *Technical report*, Centre National de la Recherche Scientifique - Laboratoire d'Analyse et d'Architecture des Système (CNRS/LAAS), Toulouse-France.
- Fabro, J. A. (2003). *Uma abordagem neuro-nebulosa para controle preditivo de processos multi-estágios*, PhD thesis, Programa de Pós-Graduação em Informática Industrial e Eng. Elétrica - CEFET-PR, Curitiba-PR-Brazil.
- Fabro, J. A., Arruda, L. V. R. & Neves-Jr, F. (2005). Startup of a distillation column using intelligent control techniques, *Computers and Chemical Engineering* **30**: 309–320.
- Fahlmann, S. (1988). An empirical study of learning speed in backpropagation networks, *Technical report*, Carnegie Mellon University. Technical Report CMU-CS-88-162.
- Fausett, L. V. (1994). *Fundamentals of Neural Networks - Architecture, Algorithms and Applications*, Prentice-Hall Inc., New Jersey.
- Fertig, C. S., Freitas, A. A., Arruda, L. V. R. & Kaestner, C. A. (1999). Fuzzy beam-search rule induction algorithm, *3rd European Conference on Principles and Practice of Knowledge Discovery in Database*, pp. 341–347.
- Figueiredo, M. F. (1997). *Redes Neurais Nebulosas Aplicadas a Problemas de Modelagem e Controle Autônomo*, PhD thesis, Universidade Estadual de Campinas (UNICAMP), Campinas, SP.

- Figueiredo, M. F. & Gomide, F. (1999). Design of fuzzy systems using neurofuzzy networks, *IEEE Transactions on Neural Networks* **10**(04): 815–827.
- Fortuna, L., Graziani, S. & Xibilia, M. G. (2005). Soft sensor for product quality monitoring in debutanizer distillation columns, *Control Engineering Practice* **13**: 499–508.
- Fortuna, L., Rizzo, A., Sinatra, M. & Xibilia, M. G. (2003). Soft analyzers for a sulfur recovery unit, *Control Engineering Practice* **11**: 1491–1500.
- Frank, I. E. & Lanteri, S. (1988). A nonlinear regression model, *Chemometrics and Intelligent Laboratory Systems* **03**(04): 301–313.
- Gao, Y., Joo, M. & Wu, S. (2001). A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks, *IEEE Transactions on Fuzzy Systems* **09**(04): 578–594.
- Geladi, P. & Kowalski, B. R. (1986). Partial least-squares regression: a tutorial, *Analytica Chimica Acta* **185**: 1–17.
- Grewal, M. & Andrews, A. (1993). *Kalman Filtering: Theory and Practice*, Prentice Hall Information, New Jersey, USA.
- Guillaume, S. (2001). Designing fuzzy inference systems from data: an interpretability-oriented review, *IEEE Transactions on Fuzzy Systems* **09**(03): 426–443.
- Guimarães, I. A. (2005). Estatística - Notas de Aulas, *Technical report*, Fundação de Estudos Sociais do Paraná - FESP. Instituto de Ciências Sociais do Paraná - ICSP.
- Gustafson, D. E. & Kessel, W. C. (1979). Fuzzy clustering with a fuzzy covariance matrix, *Proceedings of IEEE International Conference on Fuzzy Systems*, San Diego-CA, pp. 761–766.
- Hagan, M. T. & Menhaj, M. (1994). Training feedforward networks with the marquardt algorithm, *IEEE Transactions on Neural Networks* **5**(6): 989–993.
- Haykin, S. (1999). *Neural Networks*, 2 edn, Prentice-Hall Inc., New Jersey.
- He, X. & Asada, H. (1993). A new method for identifying orders of input-output models for nonlinear dynamical systems, *Proc. of the American Control Conference*, San Francisco, California, pp. 2520–2523.

- Hecht-Nielsen, R. (1987). Kolmogorov's mapping neural network existence theorem, *Proceedings of IEEE First Annual International Conference on Neural Networks*, Vol. 3, San Diego, CA, pp. 11–13.
- Herbst, J. A. & Pate, W. T. (1999). Object components for comminution system soft sensor design, *Powder Technology* **105**: 424–429.
- Hocking, R. R. (1976). The analysis and selection of variables in linear regression, *Biometrics* **32**: 1–49.
- Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities, *Bulletin of Mathematical Biophysics* **5**: 115–133.
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components, *Journal of Educational Psychology* **24**: 417–520.
- Hwang, H. S., Kang, S. J., Woo, C. H. & Woo, K. B. (2000). Evolutionary design of fuzzy rule base for nonlinear system modeling and control, *IEEE Transactions on Fuzzy Systems* **08**(01): 37–15.
- HYSYS (1989). HYSYS 3.0. HYPROTECH Ltd.
- Jang, J. S. (1993). ANFIS: Adaptive-Neuro-based Fuzzy Inference System, *IEEE Transactions on Systems, Man, and Cybernetics* **23**(3): 665–685.
- Jang, J. S. R., Sun, C. T. & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, Upper Saddle River, NJ.
- Jiménez, F., Gómez-Skarmeta, A., Sánchez, G., Roubos, H. & Babüska, R. (2003). Accurate, transparent and compact fuzzy models by multi-objective evolutionary algorithms, in J. Casillas, O. Cordon, F. Herrera & L. Magdalena (eds), *Interpretability Issues in Fuzzy Modeling*, Springer-Verlag, Heidelberg, pp. 431–451.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems, *Transactions of the ASME—Journal of Basic Engineering* **82**(Series D): 35–45.
- Kano, M., Miyazaki, K., Hasebe, S. & Hashimoto, I. (2000). Inferential control system of distillation compositions using dynamic partial least squares regression, *Journal of Process Control* **10**(2-3): 157–166.
- Kano, M., Showchaiya, N., Hasebe, S. & Hashimoto, I. (2003). Inferential control of distillation compositions: selection of model and control configuration, *Control Engineering Practice* **11**: 927–933.

- Kaski, S. & Lagus, K. (1996). Comparing self-organizing maps, *Proceedings of ICANN'96*.
- Kaspar, M. H. & Ray, H. (1992). Chemometrics methods for process monitoring and high performance controller design, *AlChe Journal* **38**(10).
- Kiparissides, C., Verros, G. & MacGregor, J. F. (1993). Mathematical modeling, optimization, and quality control of high-pressure polymerization reactors, *Journal of Macromolecular Science - Reviews in Macromolecular Chemistry and Physics* **4**: 437–527.
- Klir, G. J. & Folger, T. A. (1988). *Fuzzy Sets, Uncertainty, and Information*, Prentice Hall, Englewood Cliffs, NJ.
- Kohonen, T. (1984). *Self-organization and associative memory*, Springer-Verlag, New York.
- Kolmogorov, A. N. (1957). On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition, *American Mathematical Society Transactions* **2**(55): 55–59.
- Kramer, M. A. (1992). Autoassociative neural networks, *Computers and Chemical Engineering* **16**(04): 313–328.
- Kresta, J. V., Marlin, T. E. & MacGregor, J. (1994). Development of inferential process models using PLS, *Computers and Chemical Engineering* **18**(7): 597–611.
- Kurkova, V. (1992). Kolmogorov's theorem and multilayer neural networks, *Neural Networks* **5**: 501–506.
- Lee, C. C. (1990). Fuzzy logic in control systems: Fuzzy logic controller-Part I and II, *IEEE Transactions on Systems, Man and Cybernetics* **20**(2): 404–418.
- Lines, B., Hartlen, D., Paquin, F. D., Treiber, S., Tremblay, M. & Bell, M. (1993). Polyethylene reactor modelling and control design, *Hydrocarbon Processing*.
- Liu, J. (2006). On-line soft sensor for polyethylene process with multiple productions grades, *Control Engineering Practice* **0**: 0.
- Ljung, L. (2000). *System identification: theory for the user*, 2nd edn, Prentice Hall International.

- Lu, N., Yang, Y., Gao, F. & Wang, F. (2004). Multirate dynamic inferential modeling for multivariable processes, *Chemical Engineering Science* **59**(4): 855–864.
- Lu, W. & Fisher, D. G. (1992). Multirate adaptive inferential estimation, *IEEE Proceedings-D Control Theory Application* **139**: 181.
- MacGregor, J. F. (1986). Control of polymerization reactors, *Proc. IFAC* **1**.
- Mallow, C. L. (1973). Some comments on Cp, *Technometrics* pp. 661–675.
- Malthouse, E. C., Tamhane, A. C. & Mah, R. S. H. (1997). Nonlinear partial squares, *Computers and Chemical Engineering* **21**(08): 875–890.
- Mamdani, E. H. (1974). Application of fuzzy algorithms for control of a simple dynamic plant, *Proceedings of the Institute of Electrical Engineers* **121**(12): 1585–1588.
- Mardia, K., Kent, J. & Bibby, J. (1980). *Multivariate Analysis*, Academic Press, London.
- Marlin, T. E. (1995). *Process Control – Designing Processes and Control Systems for Dynamic Performance*, McGraw Hill, New York, USA.
- Mejdell, T. & Skogestad, S. (1991a). Estimation of distillation composition from multiple temperature measurements using partial least-squares regression, *Industrial and Engineering Chemistry Research* **30**: 2543–2555.
- Mejdell, T. & Skogestad, S. (1991b). Estimators for ill-conditioned plants: High-purity distillation, *Preprints IFAC Symposium ADCHEM'91*, Toulouse - France, pp. 2543–2555.
- Meleiro, L. A. C. & Maciel-Filho, R. (2000). A self-tuning adaptive control applied to an industrial large scale ethanol production, *Computers and Chemical Engineering* **24**: 925–930.
- Mendel, J. M. (1995). Fuzzy logic systems for engineering: A tutorial, *Proceedings of the IEEE* **83**(3): 345–377.
- Nagai, E. Y. (2002). *Métodos de identificação de modelos fuzzy a partir de dados numéricos*, Master's thesis, Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial - CEFET-PR, Curitiba - Paraná.
- Neves-Jr, F. & Aguilar-Martin, J. (1999). Qualitative event-based expert supervision for transient condition monitoring, *Applied Intelligence* **10**: 197–210.

- Nie, J. (1995). Constructing fuzzy models by self-organizing counterpropagation network, *IEEE Transaction on Systems, Man, and Cybernetics* **25**: 963–970.
- Nørgard, M., Ravn, O., Poulsen, N. K. & Hansen, L. K. (2000). *Neural Networks for Modeling and Control of Dynamic Systems*, Springer-Verlag, London.
- Ogunnaike, A. B. (1995). A contemporary industrial perspective on process control theory and practice, *Proc. DYCORD* **1**.
- Ohtake, H., Tanaka, K., Taniguchi, T. & Wang, H. O. (2001). Model construction, rule reduction and robust compensation for generalized form of Takagi-Sugeno fuzzy systems, *IEEE Transaction on Fuzzy Systems* **09**(04): 525–538.
- Oisioviçi, R. M. & Cruz, S. L. (2000). State estimation of batch distillation columns using an extended Kalman filter, *Chemical Engineering Science* **55**: 4667–4680.
- Park, S. & Han, C. (2000). A nonlinear soft sensor based on multivariate smoothing procedure for quality estimation in distillation column, *Computers and Chemical Engineering* **24**: 871–877.
- Park, Y. M., Moon, U. C. & Lee, K. Y. (1995). A self-organizing fuzzy logic controller for dynamic systems using fuzzy auto-regressive moving average (FARMA) model, *IEEE Transactions on Fuzzy Systems* **03**(01): 75–82.
- Pearson, K. (1901). On lines and planes of closest fit to systems of points in space, *Philosophical Magazine* **2**: 559–572.
- Pedrycz, W. (1984). An identification algorithm in fuzzy relational systems, *Fuzzy Sets and Systems* **13**: 153–167.
- Pedrycz, W. (1993). *Fuzzy Control and Fuzzy Systems*, John Wiley and Sons, New York, USA.
- Pollard, J. F., Broussard, M. R., Garrison, D. B. & San, K. Y. (1992). Process identification using neural networks, *Computers and Chemical Engineering* **16**(4): 253–270.
- Ponton, J. W. & Klemes, J. (1993). Alternatives to neural networks for inferential measurement, *Computers and Chemical Engineering* **17**(10): 991–1000.
- Prett, D. M. & García, C. E. (1988). *Fundamental Process Control*, Butterworth Publisher, Boston.

- Prett, D. M. & Morari, M. (1987). *The Shell Process Control Workshop*, Butterworth Publisher, Boston.
- Qin, S. J. (1996). Neural networks for intelligent sensor and control - Practical issues and some solutions, *Neural networks for control*, Academic Press, chapter 8, pp. 215–236.
- Qin, S. J. & McAvoy, T. J. (1992). Nonlinear PLS modeling using neural networks, *Computers and Chemical Engineering* **16**(04): 379–391.
- Quinlan, J. R. (1987). Generating production rules from decision trees, *Proc. Int. Joint Conf. AI (IJCAI-87)*, pp. 304–307.
- Rallo, R., Ferre-Gine, J., Arenas, A. & Giralt, F. (2002). Neural virtual sensor for the inferential prediction of product quality from process variables, *Computers and Chemical Engineering* **26**: 1735–1754.
- Rizzo, A. & Xibilia, M. G. (2002). An innovative intelligent system for sensor validation in Tokamak machines, *IEEE Transactions on Control System Technology* **10**(03).
- Rosenblatt, F. (1962). *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*, Spartan.
- Roubos, J. A. & Setnes, M. (2001). Compact and transparent fuzzy models and classifier through iterative complexity reduction, *IEEE Transaction on Fuzzy Systems* **09**(04): 516–524.
- Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning internal representations by error propagations, in D. E. Rumelhart & J. L. McClelland (eds), *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge, MA, chapter 8, pp. 318–362.
- Setnes, M. (1999). *Complexity reduction methods for fuzzy systems design*, Fuzzy Logic Control - Advances in Applications, World Scientific Publishing.
- Setnes, M. (2000). Supervised fuzzy clustering for rule extraction, *IEEE Transactions on Fuzzy Systems* **08**(04): 416–424.
- Singh, V., Gupta, I. & Gupta, H. O. (2005). ANN based estimator for distillation - inferential control, *Chemical Engineering and Processing* **44**: 785–795.

- Skagerberg, B., MacGregor, J. F. & Kiparissides, C. (1992). Multivariate data analysis applied to low-density polyethylene reactors, *Chemometrics and Intelligent Laboratory Systems* **14**(1-3): 341–356.
- Soderström, T. & Stoica, P. (1989). *System Identification*, 1st edn, Prentice Hall Int. Ltd., U.K.
- Spiegel, M. R. (2004). *Estatística*, Makron Books, São Paulo.
- Takagi, T. & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions on Systems, Man, and Cybernetics* **15**(1): 116–132.
- Tham, M. T., Montague, G. A., Morris, A. J. & Lant, P. A. (1991). Soft-sensors for process estimation and inferential control, *J. Proc. Cont.* **01**: 3–14.
- Tyagunov, A. A. (2004). *High-performance model predictive control for process industry*, PhD thesis, Eindhoven University of Technology, Eindhoven - The Netherlands.
- Wang, L. X. & Mendel, J. M. (1991). Generating fuzzy rules from numerical data with application, *IEEE Transactions on Systems, Man, and Cybernetics* **22**(06): 1414–1427.
- Wang, Y. & Rong, G. (1999). A self-organizing neural-network-based fuzzy system, *Fuzzy Sets and Systems* **103**: 1–11.
- Wangen, L. E. & Kowalshi, B. R. (1988). A multiblock partial least squares algorithm for investigating complex chemical systems, *Journal of Chemometrics* **3**: 3.
- Welch, G. & Bishop, G. (1998). An introduction to the Kalman filter, *Technical report*, University of North Carolina at Chapel Hill.
- Willis, M. J., Montague, G. A., Di-Massimo, C., Tham, M. T. & Morris, J. (1992). Artificial neural network in process estimation and control, *Automatica* **28**(6): 1181–1187.
- Wilson, D. I., Agarwal, M. & Rippin, D. W. T. (1998). Experiences implementing the extended Kalman filter on an industrial batch reactor, *Computers Chem. Engng.* **22**(11): 1653–1672.
- Wold, H. (1966). Estimation of principal components and related models by iterative least squares, in P. R. Krishnaiah (ed.), *Multivariate Analysis*, Academic Press, pp. 391–420.

- Wold, S., Kettaneh, N. & Tjessem, K. (1996). Hierarchical multi-block PLS and PC models for easier model interpretation and as an alternative to variable selection, *Journal of Chemometrics* **10**(5-6): 463–482.
- Wold, S., Kettaneh-Wold, N. & Skagerberg, B. (1989). Nonlinear PLS modeling, *Chemometrics and Intelligent Laboratory Systems* **07**(1-2): 53–65.
- Xie, X. L. & Beni, G. (1991). A validity measure for fuzzy clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**: 841–847.
- Yeh, T.-M., Huang, M.-C. & Huang, C.-T. (2003). Estimate of process compositions and plantwide control from multiple secondary measurements using artificial neural networks, *Computers and Chemical Engineering* **27**(1): 55–72.
- Zadeh, L. A. (1965). Fuzzy sets, *Information and Control* **8**(3): 338–353.
- Zadeh, L. A. (1971). Similarity relations and fuzzy ordering, *Information Sciences* **3**: 177–206.
- Zadeh, L. A. (1973). Outline of a new approach to decision processes, *IEEE Transactions on Systems, Man, and Cybernetics* **3**: 28–44.
- Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning, *Information Sciences* **8**: 199–249.
- Zambrogna, E., Barolo, M. & Seborg, D. E. (2005). Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis, *Journal of Process Control* **15**: 39–52.
- Zimmermann, H. J. (1991). *Fuzzy Set Theory and its Applications*, Kluwer Academic Publishers.
- Zuchini, M. H. (2003). *Aplicações de mapas auto-organizáveis em mineração de dados e recuperação de informação*, Master's thesis, Faculdade de Engenharia Elétrica e de Computação - UNICAMP, Campinas - SP - Brazil.