

**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

**PROGRAMA DE PÓS-GRADUAÇÃO EM  
ENGENHARIA ELÉTRICA**

**SISTEMA DE CONTROLE PREDITIVO MULTIMODELOS  
FUZZY TS-BFO EMBARCADO EM UM CONTROLADOR  
LÓGICO PROGRAMÁVEL**

**João Paulo Carvalho Henriques**

**Itajubá, Agosto de 2014**

**UNIVERSIDADE FEDERAL DE ITAJUBÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
ENGENHARIA ELÉTRICA**

**João Paulo Carvalho Henriques**

**SISTEMA DE CONTROLE PREDITIVO MULTIMODELOS  
FUZZY TS-BFO EMBARCADO EM UM CONTROLADOR  
LÓGICO PROGRAMÁVEL**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de Mestre em Ciências em Engenharia Elétrica.

Área de Concentração: Automação e Sistemas Elétricos Industriais

Orientador: Prof. Dr. Luís Henrique de Carvalho Ferreira

Co-orientador: Prof. Dr. Jeremias Barbosa Machado

**Agosto de 2014**

**Itajubá - MG**

**UNIVERSIDADE FEDERAL DE ITAJUBÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
ENGENHARIA ELÉTRICA**

**João Paulo Carvalho Henriques**

**SISTEMA DE CONTROLE PREDITIVO MULTIMODELOS  
FUZZY TS-BFO EMBARCADO EM UM CONTROLADOR  
LÓGICO PROGRAMÁVEL**

Dissertação aprovada por banca examinadora em 13 de Agosto de 2014, conferindo ao autor o título de **Mestre em Ciências em Engenharia Elétrica**

**Banca Examinadora:**

Prof. Dr. Luís Henrique de Carvalho Ferreira - UNIFEI

Prof. Dr. Jeremias Barbosa Machado - UNIFEI

Prof. Dr. Carlos Alberto Murari Pinheiro - UNIFEI

Prof. Dr. Ronaldo Rossi - UNESP-Feg

**Agosto de 2014**

**Itajubá - MG**

# *Agradecimentos*

Agradeço primeiramente a Deus por me dar saúde e disposição para a realização deste sonho.

Aos meus orientadores Prof. Dr. Jeremias Barbosa Machado e Prof. Dr. Luís Henrique de Carvalho Ferreira pela orientação e amizade durante este período de mestrado.

À minha querida esposa Janaina pelo apoio incondicional nos momentos difíceis.

Aos meus pais Flavio e Suely pelos inúmeros ensinamentos durante minha formação.

Aos amigos do INATEL, em especial ao Prof. Dr. Alexandre Baratella Lugli, pelo apoio e incentivo na realização deste trabalho.

Por fim, agradeço à B&R Automação Industrial, em nome do Sr. Evandro Rogério Mazo Avancicni, pela doação do controlador lógico programável utilizado nesta dissertação.



# Resumo

Este trabalho aborda o problema da identificação e controle de sistemas industriais não-lineares através de um algoritmo de controle preditivo que utiliza multimodelos lineares. Algoritmos de controle preditivo baseados em modelos (MBPC - *Model Based Predictive Controller*) utilizam o modelo do processo para a determinação do conjunto de previsões de saída e desta forma determinar qual a ação de controle ótima a ser adotada. Neste contexto, a proposta deste trabalho é implementar um sistema de controle preditivo em um controlador lógico programável (CLP), utilizando para a representação dos sistemas não-lineares modelos *fuzzy* Takagi-Sugeno (TS) com base de funções ortonormais nos consequentes das regras.

As bases de funções ortonormais apresentam características estruturais interessantes para representação de sistemas dinâmicos, com destaque para a ausência de realimentação de saída, característica de suma importância em algoritmos de controle preditivo. Dentre as bases de funções ortonormais utilizadas na modelagem de sistemas dinâmicos, destacam-se as bases de funções ortonormais generalizadas (GOBF) com funções internas em estrutura *Ladder*. Com a utilização de tais funções o sistema dinâmico sob análise é parametrizado utilizando somente valores reais, independente da natureza de seus polos.

Os modelos *fuzzy* TS-GOBF neste trabalho são obtidos através de amostras da entrada e saída do sistema. Os antecedentes das regras *fuzzy* são determinados através da técnica de agrupamento *fuzzy* (*fuzzy clustering*), sendo o número ideal de grupos obtido através de critérios de avaliação de agrupamento *fuzzy*. Os parâmetros dos consequentes das regras, formados por GOBFs, são inicialmente obtidos utilizando-se o método dos mínimos quadrados locais. Determinados os modelos *fuzzy* TS-GOBF inicial, são utilizadas técnicas de simplificação da base de regras *fuzzy* e um algoritmo para a otimização dos parâmetros do modelo TS-GOBF, como as funções de pertinência nos antecedentes das regras e os parâmetros nos consequentes.

Obtido o modelo *fuzzy* TS-GOBF otimizado, os controladores preditivos lineares que atuarão nos modelos locais são embarcados no CLP, juntamente com a base de regras *fuzzy* e com os parâmetros das GOBFs. A ação de controle global é obtida através da combinação ponderada das ações dos controladores locais. A cada ciclo do CLP a ação de controle global é atualizada e aplicada no processo sob controle.

A abordagem proposta neste trabalho apresenta vantagens com relação a outras metodologias de controle não-linear utilizadas na indústria, uma vez que o sistema de controle em questão pode ser implementado em CLPs comerciais de baixo custo utilizando a linguagem Texto Estruturado. Para ilustrar a proposta dessa dissertação, são apresentados, no final deste trabalho, exemplos de modelagem e controle de processos reais.

**Palavras-Chave:** Controle Preditivo Não-Linear, Controlador Lógico Programável, Base de Funções Ortonormais, Modelos *Fuzzy* TS, Identificação de Sistemas Dinâmicos.

# *Abstract*

This work involves around the identification and control of nonlinear systems using a model based predictive controller with multiple linear models. The Model Based Predictive Controller (MBPC) uses a model of the process to obtain a set of output predictions and determine the optimal control action to be applied to the system. The propose of this works is to implement a MBPC in a Programmable Logical Controller (PLC), using a fuzzy Takagi-Sugeno (TS) model with the framework of orthonormal basis function (OBF) in the fuzzy consequents, to represent the nonlinear process.

The orthonormal basis functions presents interesting structural characteristics to represent dynamic systems, for instance the absence of output feedback, a very important characteristic for predictive control algorithm. Among the orthonormal basis functions used to modeling dynamic systems, the ladder-structures generalized orthonormal basis functions (GOBF) stands out. The Ladder-structured GOBFs allows to model dynamics systems through a real values parameters, independent of the system poles.

The fuzzy TS-GOBF model used in this work is obtained using input/outputs samples of the system. The fuzzy antecedents are determined by the fuzzy clustering technique and the optimal number of fuzzy groups are determined by a fuzzy clustering evaluation criteria. The parameters of fuzzy consequents, made up by GOBFs, are initially obtained using local least square method. Once the initial fuzzy TS-GOBF models is determined, techniques to simplify the fuzzy based rules are used and an algorithm optimizes the parameters of the TS-GOBF model, such as fuzzy pertinence functions in the antecedents and the parameters of the consequents.

Once the optimized fuzzy TS-GOBF model is obtained, the linear predictive controllers that will act in the local models are embedded in PLC, along with the fuzzy rules and the GOBFs parameters. The global control action is obtained with a weighted combination of local controllers. The global control action is updated and applied in the process at every PLC Cycle.

The approach proposed in this work presents advantages over the other nonlinear control techniques used in the industry, once the proposed control system could be implemented in low coast commercial PLCs using a Structured Text language. To illustrate the purpose of this dissertation examples of modeling and control of real process are presented.

**Keywords:** Nonlinear Model Predictive Control, Programmable Logic Controller, Orthonormal Basis Function, Fuzzy TS models, Identification of dynamics systems.

# *Sumário*

**Lista de Figuras**

**Lista de Tabelas**

**Lista de Siglas e Abreviações**

**Lista de Símbolos**

<b>1</b>	<b>Introdução</b>	p. 14
<b>2</b>	<b>Modelagem de Sistemas Dinâmicos</b>	p. 20
2.1	Modelagem de Sistemas Lineares . . . . .	p. 20
2.1.1	Equações Diferenciais . . . . .	p. 21
2.1.2	Funções de Transferência . . . . .	p. 22
2.1.3	Espaço de Estados . . . . .	p. 23
2.1.4	Base de Funções Ortonormais . . . . .	p. 24
2.1.5	Base de Funções Ortonormais Generalizadas . . . . .	p. 28
2.1.6	Funções Internas . . . . .	p. 29
2.2	Modelagem de Sistemas Não-Lineares . . . . .	p. 31
2.2.1	Série de Volterra . . . . .	p. 32
2.2.2	Modelos de Hammersrtein e de Wiener . . . . .	p. 32
2.2.3	Modelo NOBF - <i>Nonlinear Orthonormal Basis Function</i> . . . . .	p. 33
2.2.4	Identificação de Sistemas Dinâmicos . . . . .	p. 34
2.2.5	Resumo . . . . .	p. 36

<b>3</b>	<b>Modelagem de Sistemas Não-Lineares por <i>Fuzzy-GOBF</i></b>	p. 37
3.1	Introdução aos Conjuntos <i>Fuzzy</i> . . . . .	p. 37
3.2	Modelos <i>Fuzzy</i> TS-BFO . . . . .	p. 39
3.3	Projeto de Modelos <i>fuzzy</i> -TS BFO . . . . .	p. 42
3.3.1	Agrupamento . . . . .	p. 42
3.3.2	Algoritmo de Gustafson-Kessel . . . . .	p. 43
3.3.3	Determinação do Número de Regras . . . . .	p. 44
3.3.4	Determinação do Número de Grupos . . . . .	p. 47
3.4	Construção dos Modelos <i>Fuzzy</i> . . . . .	p. 48
3.4.1	Determinação dos Antecedentes das Regras . . . . .	p. 48
3.4.2	Determinação dos Consequentes das Regras . . . . .	p. 49
3.4.3	Simplificação dos Modelos <i>Fuzzy</i> . . . . .	p. 51
3.5	Otimização do modelo <i>fuzzy</i> TS-GOBF . . . . .	p. 54
3.6	Resumo . . . . .	p. 56
<b>4</b>	<b>Controle Preditivo Não-Linear</b>	p. 58
4.1	Controle Preditivo . . . . .	p. 58
4.1.1	Modelagem do Processo e Previsão da Saída $j$ Passos a Frente . . . . .	p. 61
4.1.2	Definição da Lei de Controle Preditivo Linear . . . . .	p. 63
4.1.3	Controle Preditivo Linear com BFO e Critério de Custo Quadrático . . . . .	p. 65
4.2	Controlador Preditivo Não-Linear . . . . .	p. 66
4.3	Resumo . . . . .	p. 67
<b>5</b>	<b>Sistema de Controle Preditivo Embarcado</b>	p. 69
5.1	Introdução a Controladores Lógicos Programáveis . . . . .	p. 69
5.1.1	Arquitetura dos CLPs e Princípio de Funcionamento . . . . .	p. 70
5.1.2	Linguagens de Programação . . . . .	p. 73

5.2	Implementação do Controle Preditivo Não-Linear em um Controlador Lógico Programável . . . . .	p. 75
5.2.1	Identificação . . . . .	p. 75
5.2.2	Cálculo das Ações de Controle . . . . .	p. 77
5.3	Resumo . . . . .	p. 84
<b>6</b>	<b>Aplicações e Resultados</b>	p. 86
6.1	Descrição Sistema de Nível . . . . .	p. 86
6.2	Modelagem e controle de nível em um tanque - Abordagem Linear . . . . .	p. 87
6.3	Modelagem e controle de nível em um tanque - Abordagem Não-Linear . . . . .	p. 93
6.4	Resumo . . . . .	p. 100
<b>7</b>	<b>Conclusão</b>	p. 102
7.1	Contribuições . . . . .	p. 104
7.2	Trabalhos Futuros . . . . .	p. 104
7.3	Trabalhos publicados e em desenvolvimento . . . . .	p. 104
	<b>Referências</b>	p. 106
	<b>Apêndice A – Método de Mínimos Quadrados</b>	p. 112
	<b>Apêndice B – Código para Implementação do Controlador Preditivo Fuzzy TS-GOBF</b>	p. 114

## *Lista de Figuras*

1	Representação Gráfica das Funções de Laguerre. . . . .	p. 26
2	Representação Gráfica de um modelo GOBF. . . . .	p. 28
3	Representação em blocos da estrutura interna parametrizada por $\gamma$ . . .	p. 30
4	Representação em blocos das funções da base de uma GOBF. . . . .	p. 30
5	Modelo de Hammerstein. . . . .	p. 32
6	Modelo de Wiener. . . . .	p. 33
7	Funções de Pertinência Trapezoidal. . . . .	p. 38
8	Funções de Pertinência Trapezoidal. . . . .	p. 49
9	Fusão de grupos similares. . . . .	p. 52
10	Diagrama de Blocos do Controlador Preditivo. . . . .	p. 60
11	Controlador Preditivo Proposto. . . . .	p. 67
12	Diagrama em Blocos do CLP. . . . .	p. 71
13	<i>Scan</i> do CLP. . . . .	p. 72
14	Tempo de Ciclo do CLP. . . . .	p. 72
15	Tempo de Ciclo do CLP <i>Fast Reaction</i> . . . . .	p. 73
16	Fluxograma dos passos para obtenção do modelos TS-GOBF. . . . .	p. 76
17	Fluxograma dos passos para execução do controle TS-GOBF. . . . .	p. 78
18	Cálculo da Resposta Livre do Modelo Local. . . . .	p. 79
19	Cálculo do vetor $f$ do modelo local. . . . .	p. 80
20	Cálculo da ação de controle local. . . . .	p. 80
21	Cálculo da ação de controle global. . . . .	p. 81
22	Cálculo da ação de controle global. . . . .	p. 82

23	Cálculo da saída das funções GOBF. . . . .	p. 83
24	Saída dos modelos locais. . . . .	p. 83
25	Saída global do sistema. . . . .	p. 84
26	Sistema de Controle de Nível. . . . .	p. 86
27	Sistema de Nível Didático - Datapool. . . . .	p. 87
28	CLP utilizado na identificação e controle dos sistemas de nível. . . . .	p. 88
29	Dados utilizados na identificação do sistema - Abordagem Linear. . . . .	p. 89
30	Comparação entre o sistema real e o modelo obtido. . . . .	p. 90
31	Resposta do sistema de nível - Controlador Preditivo Linear. . . . .	p. 91
32	Resposta do sistema de nível - Controlador PID. . . . .	p. 92
33	Comparação entre as respostas do Controlador PID e o Controlador Preditivo Linear. . . . .	p. 92
34	Planta Didática de Nível - Festo Didatic. . . . .	p. 94
35	Dados utilizados na estimação do modelo. . . . .	p. 94
36	Funções de pertinência presentes nos antecedentes das regras do modelo TS-GOBF inicial. . . . .	p. 95
37	Comparação entre o sistema real o modelo <i>fuzzy</i> TS-GOBF inicial. . . . .	p. 95
38	Funções de pertinência presentes nos antecedentes das regras do modelo TS-GOBF otimizado. . . . .	p. 96
39	Comparação entre o sistema real o modelo <i>fuzzy</i> TS-GOBF final. . . . .	p. 97
40	Resposta do Sistema de Nível - Controlador Preditivo Não-Linear. . . . .	p. 97
41	Resposta do Sistema de Nível - Controlador PID. . . . .	p. 98
42	Comparação entre as respostas do Controlador PID e o Controlador Preditivo Não-Linear. . . . .	p. 99

## *Lista de Tabelas*

1	Aplicações Comercial de MBPC Linear. . . . .	p. 59
2	Custo Computacional dos Controladores - Abordagem Linear. . . . .	p. 93
3	Custo Computacional dos Controladores - Abordagem Não-Linear . . . .	p. 99



# *Lista de Siglas e Abreviações*

APD	<i>Average Partition Density</i>
ARMAX	<i>AutoRegressive Moving Average with Exogenous inputs</i>
ARX	<i>AutoRegressive with Exogenous Inputs</i>
AWCD	<i>Average Within-Cluster Distance</i>
ASWC	<i>Average Silhouette Width Criterion</i>
BLUE	<i>Best Linear Unbiased Estimator</i>
CARIMA	<i>Controlled Auto Regressive Integrated Moving Average</i>
CCM	<i>Compatible Cluster Merging</i>
CLP	Controlador Lógico Programável
CPU	<i>Central Processing Unit</i>
DMC	<i>Dynamic Matrix Controller</i>
EQM	Erro Quadrático Médio
BFO	Base de Função Ortonormal
FBD	<i>Function Block Diagram</i>
FHV	<i>Fuzzy HyperVolume</i>
FIR	<i>Finite Impulse Response</i>
FS	<i>Fuzzy Silhouette</i>
FT	Função de Transferência
GK	Gustafson-Kessel
GOBF	<i>Generalized Orthonormal Basis Function</i>
GPC	<i>Generalized Predictive Control</i>
MAP	<i>Manufacturing Automation Protocol</i>
LD	<i>Ladder Diagram</i>
IEC	<i>International Electrotechnical Commission</i>
IHM	Interface Homem Máquina

IIR	<i>Infinite Impulse Response</i>
IL	<i>Instruction List</i>
MBPC	<i>Model Based Predictive Control</i>
MMAC	<i>Multiple Model Adaptative Control</i>
MODICON	<i>Modular Digital Controller</i>
MQ	Mínimos Cuadrados
NOBF	<i>Nonlinear Orthonormal Basis Function</i>
PID	Proporcional+Integral+Derivativo
SFC	<i>Sequential Function Chart</i>
ST	<i>Structured Text</i>
TS	Takagi-Sugeno

# *Lista de Símbolos*

$\psi(k)$	Resposta de uma função ortonormal a uma entrada $u(k)$
$\gamma$	Parâmetro da GOBF com funções internas em estrutura <i>ladder</i>
$\Upsilon$	Conjunto de parâmetros $\gamma$
$n_b$	Número de polos da GOBF
$n_b$	Número de polos da GOBF
$K$	Número de regras <i>fuzzy</i> - modelos locais
$\varsigma$	Número de variáveis nos antecedentes das regras <i>fuzzy</i>
$\mu_i$	Valor de ativação de uma regra <i>fuzzy</i> $i$
$F_i$	Matriz de covariância do grupo $i$
$S(A, B)$	Valor de similaridade entre duas funções de pertinência $A$ e $B$
$L_{i,j}$	Função de pertinência dos modelos <i>fuzzy</i> TS-BFO
$v_{i,j}$	Protótipo dos grupos de particionamento
$\hat{y}(k)$	Saída estimada
$y(k)$	Saída medida ou simulada
$N_1$	Horizonte inicial de previsão de saída
$N_y$	Horizonte final de predição de saída
$N_u$	Horizonte de controle
$u(k)$	Sinal de controle
$\Delta u(k)$	Variação do sinal de controle
$\omega$	Sinal de referência

# 1 *Introdução*

Atualmente no meio industrial a maior parte dos processos a serem controlados são bem atendido por sistemas de controle clássico como o PID (Proporcional + Integral + Derivativo) de ganhos fixos e controladores clássicos com realimentação de estados. No entanto, alguns processos específicos, como os sistemas multivariáveis, não-lineares e invariantes no tempo necessitam de sistemas de controle complexos para atender de forma eficiente as rígidas especificações de projeto. Dentre as técnicas de controle avançado utilizadas no meio industrial para controle deste tipo de processo, o controle preditivo baseado em modelos têm emergido com uma técnica promissora (CAMACHO; BORDONS, 2004).

O conceito de controle preditivo foi introduzido nos trabalhos de Richalet et al. (1978) com o *Identification Command* (IDCOM) e de Cutler e Ramaker (1980) com o *Dynamic Matrix Controller* (DMC). A partir destes trabalhos surgiu a metodologia dos MBPC<sup>1</sup> utilizando representação a partir de modelos de resposta ao impulso. Algoritmos de controle preditivo podem ser utilizados para controle de diferentes tipos de sistemas, incluindo processos instáveis, de fase não-mínima, não-lineares, com grande atraso de transporte e etc, sem grandes mudanças do ponto de vista conceitual (CAMPELLO; OLIVEIRA; AMARAL, 2007). Por incorporar nas leis de controle as restrições relacionadas com a entrada/saída do processo, os algoritmos de controle preditivo estão cada vez mais presentes na indústria, em especial no controle de processos químicos e petroquímicos.

Nos algoritmos de controle preditivo, a lei de controle e, conseqüentemente, o desempenho do sistema em malha fechada, está diretamente relacionado com o modelo do processo a ser controlado. Desta forma, a modelagem do processo é uma etapa vital para o bom desempenho de um sistema de controle. Diferentes tipos de modelos podem ser utilizados na representação do processo, dentre eles destacam-se: os sistemas de controle preditivo utilizando modelo CARIMA (*Controlled Auto Regressive Integrated Moving Average*) (CLARKE; MOHTADI, 1989); modelo de resposta ao impulso e de res-

---

<sup>1</sup> *Model Based Predictive Control*

posta ao degrau (MORARI; GARCIA; PRETT, 1989). Além destes modelos de destaque na literatura podem se obter sistemas de controle baseados em modelos obtidos através de base de funções ortonormais (BFO) (OLIVEIRA, 1997).

A utilização de sistemas de controle preditivo baseados em modelos BFO se mostra vantajosa sobre demais abordagens, pois tais modelos não apresentam realimentação de saída. Desta forma, os modelos BFO não propagam erro na predição de saída, característica de suma importância em sistemas de controle baseados em modelos (CAMPELLO; OLIVEIRA; AMARAL, 2007). Além disso, a representação de sistemas dinâmicos através de modelos BFO apresenta vantagens quando comparada com outros modelos normalmente utilizados para representação dos mesmos sistemas dinâmicos, como: a não necessidade de se conhecer previamente a ordem do modelo nem o atraso de transporte, eliminando-se assim a etapa de determinação dos regressores; se necessário é possível aumentar a capacidade de representação do modelo aumentando-se o número de funções ortonormais utilizadas; garantia de que a representação de um sistema estável é também estável; desacoplamento natural das saídas em modelos multivariáveis; tolerância a dinâmicas não modeladas. Estas propriedades fazem dos modelos BFO muito atraentes para serem utilizados na identificação e projeto de sistemas de controle (WAHLBERG, 1994; HOF; HEUBERGER; BOKOR, 1995; OLIVEIRA, 1997; CAMPELLO; AMARAL, 2002; CAMPELLO; MELEIRO; AMARAL, 2004; MACHADO, 2007; HENRIQUES; MACHADO; FERREIRA, 2013).

As bases de funções ortonormais mais utilizadas na representação de sistemas dinâmicos lineares e não-lineares são as bases de Laguerre, de Kautz e a base de funções ortonormais generalizadas (HEUBERGER; Van den Hof; WAHLBERG, 2005). A base de Laguerre é mais indicada para representar sistemas com polos puramente reais ou com parte imaginária de valor reduzido. Por ser parametrizado por polos complexos conjugados, as bases de Kautz necessitam de um número menor de funções para representar sistemas com dinâmica oscilatória. Quando se deseja representar sistemas com múltiplos modos dominantes (amortecidos ou não), utilizam-se as bases de funções ortonormais generalizadas, em que as funções ortonormais podem apresentar vários polos distintos.

A classe das bases de funções ortonormais generalizadas (GOBF - *Generalized Orthogonal Basis Functions*), é obtida pela conexão de  $n$  filtros *all-pass* de  $n_n$ -ésima ordem, no qual cada filtro terá  $n_b$  polos distintos ou não, podendo ser de qualquer natureza, reais ou complexos. Um caso particular das GOBFs, denominada funções de Takenaka-Malmquist, consiste em conectar diferentes filtros *all-pass* de primeira e segunda ordem,

sendo que cada filtro pode apresentar polo(s) distinto(s). Entretanto, as funções de Takenaka-Malmquist apresentam algumas desvantagens quando não se tem conhecimento a priori da ordem e do tipo dos polos do sistema, pois modelos com polos complexos geram saídas complexas. Uma alternativa para tal desvantagem é a utilização de GOBF com funções internas de estrutura *ladder*, na qual os modelos são parametrizados por parâmetros reais, independente da natureza dos polos da funções internas (HEUBERGER; Van den Hof; WAHLBERG, 2005). Tais características fazem os modelos GOBF com funções internas de estrutura *ladder* adequados para a utilização em sistemas de controle baseado em modelos, em especial para sistemas de controle preditivo.

Nos processos industriais em geral, a maioria das plantas a serem controladas contém algum grau de não-linearidade. Com esta motivação, alguns algoritmos de MBPC baseados em modelos não-lineares foram propostos na literatura (OLIVEIRA; LEMOS, 1995; III; OGUNNAIKE; PEARSON, 1995; ALLGOWER; ZHENG, 2000; OLIVEIRA; AMARAL; LATAWIEC, 2003). Neste contexto, este trabalho propõe o desenvolvimento de um sistema de controle preditivo não-linear utilizando uma abordagem multimodelos, no qual o sistema a ser controlado será representado por modelos *fuzzy* Takagi-Sugeno (TS) com estruturas GOBF nos consequentes das regras.

O conceito de conjuntos *fuzzy* foi originalmente proposto por Zadeh (1965) e com os trabalhos de Takagi e Sugeno (1985) foi introduzida uma nova abordagem de modelagem baseada em regras compostas de conjuntos *fuzzy*, cujos modelos são capazes de representar por exemplo, o comportamento de sistemas não-lineares. Essa representação é possível, pois os modelos *fuzzy* TS possuem a propriedade de serem aproximadores universais de funções em um espaço compacto (WANG; MENDEL, 1992b; KOSKO, 1994a). Estes modelos, denominados *fuzzy* Takagi-Sugeno (TS), consistem de interpolações de múltiplos modelos locais (usualmente lineares e afins), o que torna tais modelos interessantes para o projeto de sistemas de controle, já que localmente se trabalha com sistemas lineares (MACHADO, 2007).

No presente trabalho utiliza-se para a modelagem dos sistemas não-lineares a serem controlados modelos *fuzzy* TS que utilizam GOBF com estrutura interna tipo *ladder* nos consequentes das regras (CAMPELLO; AMARAL, 2002; MEDEIROS; AMARAL; CAMPELLO, 2006; MACHADO, 2007). O modelo *fuzzy* TS-GOBF apresenta todas as características já descritas para as estruturas dinâmicas que utilizam BFO. Outras abordagens de aproximadores de funções não-lineares utilizando BFO (redes neurais artificiais, funções de base radial e etc.) são encontradas na literatura, porém não são constituídas

por modelos locais e não são matematicamente interpretáveis (CAMPELLO; MELEIRO; AMARAL, 2004; MACHADO, 2011) o que dificulta sua utilização em sistemas de controle baseado em modelo.

A obtenção do modelo *fuzzy* TS-GOBF nesta dissertação se dá através da metodologia proposta por Machado (2011), na qual o modelo *fuzzy* TS-GOBF é obtido utilizando técnicas de otimização não-linear sem nenhum conhecimento a priori das características dinâmicas do sistema. O processo consiste em primeiramente em obter-se um modelo *fuzzy* TS-GOBF inicial através de técnicas de agrupamento *fuzzy* e critérios de avaliação dos agrupamentos (MACHADO, 2007). Em seguida, aplicam-se técnicas para simplificação da base de regras do modelo *fuzzy*, e por fim, através de técnicas de otimização não-lineares, otimizam-se os parâmetros do modelo *fuzzy* TS-GOBF.

Neste trabalho, os antecedentes são obtidos através do algoritmo agrupamento *fuzzy*, proposto por Gustafson e Kessel (1979), aplicados as amostras de entrada e saída do sistema. A determinação do número de grupos que melhor agrupa os dados do sistema se dá através de critérios de avaliação de agrupamentos *fuzzy* (GATH; GEVA, 1989; CAMPELLO; HRUSCHKA, 2006). Obtidos os antecedentes das regras, determinam-se os parâmetros dos modelos locais através de estimação por mínimos quadrados locais, que levam em consideração a ativação das regras de acordo com os dados de entrada do sistema.

Obtido o modelo *fuzzy* TS-GOBF inicial, aplica-se um algoritmo para a análise e simplificação do modelo *fuzzy* utilizando critérios de similaridade entre regras e funções de pertinência *fuzzy* (BABUSKA, 1998). Finalizada a simplificação, utiliza-se, sobre o modelo inicialmente obtido, um algoritmo de otimização não-linear, proposto por Machado (2011), para o ajuste dos polos e dos coeficientes da expansão das funções ortonormais presentes nos consequentes das regras e dos parâmetros das funções de pertinência nos antecedentes das regras.

Uma vez modelado o sistema dinâmico sob estudo, define-se os parâmetros  $N_y$ ,  $N_u$  e  $\alpha$  dos sistemas de controle preditivo lineares, que atuarão localmente sobre os modelos locais. As ações dos controladores locais são combinadas de acordo com a ativação de cada regra dos respectivos modelos locais e a ação de controle global é aplicada ao processo.

A utilização de técnicas avançadas de controle industriais normalmente demanda um grande esforço computacional ou requer um hardware dedicado para sua aplicação. Desta forma, é comum encontrar em plantas industriais, para controle de sistemas dinâmicos complexos, computadores industriais de alto desempenho (PC Industrial *ADVANTECH*

- UNO) ou ainda sistemas dedicados de controle (*CompactRIO* da *National Instruments*). Neste contexto, propõe-se nesta dissertação o desenvolvimento e implementação do sistema de controle preditivo multimodelos *fuzzy* TS-GOBF em um Controlado Lógico Programável comercial.

O Controlador Lógico Programável (CLP) surgiu no final da década de 60 com o objetivo de automatizar e tornar mais flexíveis as linhas de produção em série, especialmente a plantas automobilísticas. Os primeiros CLPs eram dispositivos caros e realizavam somente operações digitais, como leitura de sensores digitais, acionamento de reles e contadores, temporização, contagem de eventos e etc. A partir de meados dos anos 80, com a evolução da eletrônica e dos sistemas computacionais, surgiram novos CLPs que executavam a leitura e a interpretação de grandezas analógicas, possibilitando assim, implementar algoritmos de controle clássicos como o PID (Proporcional+Integral+Derivativo) (MORAES; CASTRUCCI, 2007).

Desenvolvido com intuito de automatizar os processos industriais, até o final dos anos 80, os CLPs eram programados utilizando uma linguagem de programação específica para automação denominada *Ladder*, sendo que cada fabricante implementava em seu CLP sua própria versão da linguagem, uma vez que esta não era padronizada. Com o objetivo de solucionar os problemas causados pela falta de padronização, foi publicada no início dos anos 90 a primeira norma relacionada a CLPs, na qual eram definidas as linguagens de programação, ciclo de vida do CLP, estrutura de software, etc. Assim, a partir da publicação da norma e de suas revisões relativas a linguagem de programação, no início do ano 2000, surgiram os primeiros CLPs que utilizavam linguagens de programação padronizadas que permitiam implementar funções matemáticas complexas e conseqüentemente, algoritmos avançados de controle.

O conteúdo dessa dissertação está apresentado da seguinte forma:

O Capítulo 2 apresenta os conceitos básicos utilizados para a modelagem de sistemas dinâmicos lineares e não-lineares apresentando com maiores detalhes, a representação de sistemas dinâmicos utilizando Base de Funções Ortonormais Generalizadas com funções internas de estrutura *ladder*. Em seguida, são apresentadas algumas metodologias para representação de sistemas não-lineares, e por fim, é feita uma breve introdução de identificação de sistemas dinâmicos, dando-se maior ênfase no método dos Mínimos Quadrados.

No Capítulo 3 é descrita uma breve introdução da teoria de conjuntos *fuzzy* apresentado os modelos mais utilizados, com especial destaque para o modelo *fuzzy* Takagi-Sugeno. Em seguida, é detalhada a metodologia para o projeto de modelos *fuzzy* TS



com BFO nos consequentes das regras utilizada para representação de sistemas dinâmicos. Para a etapa do projeto, inicialmente é apresentada a abordagem para obtenção das regras *fuzzy*, utilizando o algoritmo de agrupamento *fuzzy* proposto por Gustafson e Kessel (1979), e alguns critérios de avaliação do melhor número de grupos. Posteriormente, são descritas duas variações do método dos mínimos quadrados, estimação local e global, utilizados para obtenção dos consequentes das regras *fuzzy*. Por fim, é apresentada uma metodologia para simplificação da base de regras do modelo *fuzzy*, bem como a técnica de otimização dos parâmetros das GOBF utilizadas nos consequentes das regras.

O Capítulo 4 apresenta uma introdução aos sistemas de controle preditivo baseado em modelos com suas principais características e alguns exemplos de aplicação. Em seguida, é detalhado o algoritmo de controle preditivo linear baseado em modelos GOBF, que neste trabalho irão atuar de forma local nos modelos de cada regra. Por fim, é descrito o algoritmo de controle preditivo não-linear que atuará no controle de sistemas dinâmicos através de uma ação de controle global obtida pela combinação ponderada das ações dos controladores locais.

O Capítulo 5 apresenta um breve histórico sobre Controladores Lógicos Programáveis, descrevendo com detalhes seu princípio de funcionamento. Em seguida, é detalhado o algoritmo utilizado para identificação do sistema a ser controlado, utilizando a metodologia já descrita anteriormente. Por fim, será apresentada a metodologia para a implementação do sistema de controle preditivo proposto neste trabalho, em um controlador lógico programável detalhando-se os passos necessários para o cálculo das ações dos controladores locais e o cálculo da ação de controle global.

No Capítulo 6 serão apresentadas duas aplicações do sistema de controle proposto, utilizando um CLP comercial e duas plantas didáticas de nível. Inicialmente, apresenta-se a modelagem e o controle de uma planta de nível didática, utilizando uma abordagem linear do sistema de controle em questão. Em seguida, é apresentada a modelagem e o controle de uma segunda planta didática utilizando a abordagem não-linear do sistema de controle preditivo proposto. Para comprovar a eficiência da metodologia de controle proposta, comparou-se o desempenho dos dois sistemas de nível controlados por um controlador PID e pelo controlador preditivo desenvolvido, comparando-se também a taxa de ocupação da CPU para as duas abordagens de controle.

O Capítulo 7 traz as conclusões finais, uma revisão sobre as contribuições do trabalho e a perspectiva de trabalhos futuros como continuação desta proposta.

## 2 *Modelagem de Sistemas Dinâmicos*

A modelagem é uma etapa vital para entendimento e estudo de sistemas dinâmicos, especialmente na área de controle. Neste capítulo serão apresentadas algumas formas de representar matematicamente sistemas dinâmicos, com especial interesse na modelagem por base de funções ortonormais generalizadas com funções internas em estrutura *Ladder*. Também serão apresentadas técnicas de modelagem para sistemas não-lineares presentes na literatura e o estimador de mínimos quadrados utilizado na identificação de sistemas dinâmicos (AGUIRRE, 2004; MACHADO, 2007; CAMPELLO; OLIVEIRA; AMARAL, 2007).

### 2.1 Modelagem de Sistemas Lineares

Modelar um sistema dinâmico significa obter uma representação matemática capaz de descrever as principais características de sistemas físicos reais. Existem muitas maneiras de representar sistemas dinâmicos utilizando equações ou relações matemáticas, podendo haver algum modelo que se adapte melhor ao sistema a ser estudado (SOUZA; PINHEIRO, 2008). Em engenharia, os modelos matemáticos que representam sistemas dinâmicos são utilizados para análise, otimização, simulação e controle de processos, sendo possível modelar um sistema linear simples ou uma máquina complexa. Neste trabalho, o objetivo é modelar sistemas dinâmicos com a finalidade de analisar seu comportamento, bem como utilizar tais modelos para projeto de sistemas de controle preditivo.

Obter modelos que descrevem exatamente o comportamento de um sistema dinâmico é algo muito difícil. O que ocorre na prática, é uma representação aproximada do sistema real através de um modelo, ou de um conjunto de modelos, que devem representar o sistema em estudo de maneira mais fidedigna possível (LJUNG, 1999). Muitas vezes a complexidade que os sistemas reais podem apresentar levam à necessidade de utilizar

aproximações simplificadoras das características desses sistemas.

Um sistema dinâmico é dito linear quando este obedece o princípio da superposição (AGUIRRE, 2004). Sistemas que não obedecem tal princípio são conhecidos como não-lineares. Para a representação de sistemas dinâmicos, sejam eles lineares ou não-lineares, existem uma grande variedade de modelos. Tais modelos em geral são mais adequados dependendo da natureza (dinâmica) do sistema ou de sua aplicação. A seguir, serão apresentadas algumas técnicas de modelagem de sistemas dinâmicos, descrevendo com maiores detalhes, a modelagem de sistemas dinâmicos por Base de Funções Ortonormais que é de especial interesse neste trabalho.

### 2.1.1 Equações Diferenciais

O estudo do cálculo diferencial e integral teve início em 1666 com Isaac Newton e Gottfried Wilhelm Leibniz e até hoje atrai a atenção de muitos estudiosos. Sistemas físicos reais frequentemente envolvem em sua dinâmica uma taxa de variação de uma ou mais variáveis, resultando em uma ou mais equações utilizando derivadas. Para entender e estudar processos físicos que envolvam fluxo de corrente elétrica, movimento de fluidos e transferência de calor, é necessário conhecer as leis da física que regem o comportamento de tais processos, sendo essas equações diferenciais, os modelos dos processos sob estudo (SOUZA; PINHEIRO, 2008).

Uma equação diferencial linear de  $n$ -ésima ordem pode ser descrita da seguinte forma:

$$P_0(t) \frac{d^n y}{dt^n} + P_1(t) \frac{d^{n-1} y}{dt^{n-1}} + \dots + P_{n-1}(t) \frac{dy}{dt} + P_n y = G(t) \quad (2.1)$$

Admitindo que  $P_0, \dots, P_n$  e  $G$  são as funções contínuas que assumem o valor real no intervalo  $I : \alpha < t < \beta$ , e que  $P_0$  é diferente de zero no intervalo, pode-se então, dividindo a equação (2.1) por  $P_0$  e utilizando a notação  $dy/dt = y'$ , reescrever a equação (2.1) como:

$$y^n + p_1(t)y^{n-1} + \dots + p_{n-1}(t)y' + p_n(t)y = g(t) \quad (2.2)$$

Caso os parâmetros  $p_i$  com  $i = 1, 2, \dots, n$  apresentem valores constantes para todo  $t$ , pode-se afirmar que a equação (2.2) é uma equação diferencial linear e invariante no tempo (MACHADO, 2007).

## 2.1.2 Funções de Transferência

Em modelagem de sistemas dinâmicos, as funções de transferência são amplamente utilizadas para representar a relação entrada/saída de componentes e sistemas que podem ser modelados por equações diferenciais lineares e invariantes no tempo.

Por definição, Função de Transferência é a relação entre a transformada de Laplace de saída e a transformada de Laplace de entrada de um sistema, admitindo-se todas as condições iniciais nulas (OGATA, 2003). Considere a equação diferencial (2.3) que representa um sistema linear e invariante no tempo.

$$a_0 y^n + a_1 y^{n-1} + \dots + a_{n-1} \dot{y} + a_n y = b_0 x^m + b_1 x^{m-1} + \dots + b_{m-1} \dot{x} + b_m x \quad (2.3)$$

na qual  $y$  é a saída do sistema,  $x$  a entrada e ( $n \geq m$ ). A função de transferência desse sistema é obtida dividindo a transformada de Laplace da saída pela transformada de Laplace da entrada, com as condições iniciais nulas como mostrado na equação (2.4).

$$G(s) = \frac{Y(s)}{X(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_0} \quad (2.4)$$

Os valores de  $s$  que anulam a função  $G(s)$  são conhecidos como zeros do sistema e os valores de  $s$  que anulam o polinômio  $X(s)$  são conhecidos como polos do sistema. Para o sistema ser considerado estável, os polos devem apresentar a parte real com valores negativos. Sistemas contendo polos com parte real nula são marginalmente estáveis (OGATA, 2003).

Muitas vezes, sistemas de controle são implementados em sistemas computadorizados ou microprocessados, sendo necessário que se utilize modelos discretizados com tempo de amostragem definido. Utilizando, por exemplo, a transformada Bilinear ou Tustin apresentados em (PHILLIPS; NAGLE, 1995) é possível obter um modelo discreto (em  $z$ ) a partir de um modelo contínuo (em  $s$ ), como apresentado na equação (2.5).

$$s = \frac{2}{T_s} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (2.5)$$

sendo  $T_s$  o tempo de amostragem, que deve obedecer o critério de Nyquist (PHILLIPS; NAGLE, 1995).

Isolando  $z$  na equação (2.5) tem-se:

$$z = \frac{(2/T_s) + s}{(2/T_s) - s} \quad (2.6)$$

Desta forma, uma função de transferência discretizada apresentará o formato da equação (2.7).

$$G(s) = \frac{Y(s)}{X(s)} = \frac{r_0 + r_1z + \dots + r_{0n}z^{0n}}{d_0 + d_1z + \dots + d_{0a}z^{0a}} \quad (2.7)$$

Assim como para os sistemas contínuos, no qual a estabilidade se dá quando os polos da função apresentam parte real negativa, para sistemas discretizados a estabilidade se dá quando os polos da função estão dentro do círculo central unitário  $\|z\| \leq 1$ .

### 2.1.3 Espaço de Estados

Com o aumento da complexidade dos sistemas dinâmicos reais, novas abordagens de sistemas de controle surgiram ao longo das últimas décadas. Em muitos algoritmos de controle, especialmente no controle moderno, a representação matemática de um sistema dinâmico deve conter além de informações referentes a sua dinâmica, informações de suas condições internas. Neste contexto, a representação em espaço de estados se faz muito importante pois esta traz informações da dinâmica entre a entrada e a saída e informações intrínsecas do processo.

O conhecimento dos valores das variáveis de estado e das funções de entrada no sistema, com as equações que descrevem a dinâmica entre a entrada e a saída, fornece a possibilidade de se conhecer os estados futuros e a saída futura do sistema (DORF; BISHOP, 2009). Desta forma, as variáveis de estado são as grandezas que podem determinar o comportamento futuro de um sistema quando são conhecidos os sinais de entrada e os estados do sistema.

Admitindo um sistema dinâmico linear e invariante no tempo a ser modelado, que possui  $q$  entradas  $u_1, u_2, \dots, u_q$  e  $n$  estados do sistema representados por  $x_1, x_2, \dots, x_n$ . O conjunto de equações apresentados em (2.8) representa a relação dinâmica entre as entradas e os estados do sistema (OGATA, 2003).

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + b_{11}u_1 + b_{12}u_2 + \dots + b_{1q}u_q \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + b_{21}u_1 + b_{22}u_2 + \dots + b_{2q}u_q \\ &\vdots \\ \dot{x}_n &= a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n + b_{n1}u_1 + b_{n2}u_2 + \dots + b_{nq}u_q \end{aligned} \quad (2.8)$$

As relações entre as saídas, os estados e as entradas é dada na equação (2.9):

$$\begin{aligned}
 \dot{y}_1 &= c_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + d_{11}u_1 + d_{12}u_2 + \dots + d_{1q}u_q \\
 \dot{y}_2 &= c_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + d_{21}u_1 + d_{22}u_2 + \dots + d_{2q}u_q \\
 &\vdots \\
 \dot{y}_n &= c_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n + d_{n1}u_1 + d_{n2}u_2 + \dots + d_{nq}u_q
 \end{aligned} \tag{2.9}$$

Em sistemas dinâmicos lineares e invariantes no tempo os termos  $a_{ij}, b_{ij}, c_{ij}, d_{ij}$  são constantes e dependem do sistema a ser modelado. Os termos  $a_{ij}, b_{ij}$  modelam a parte dinâmica do sistema e os termos  $c_{ij}, d_{ij}$  modelam a parte estática.

$$\begin{aligned}
 \dot{x} &= Ax + Bu \\
 y &= Cx + Du
 \end{aligned} \tag{2.10}$$

Assim como na representação por Funções de Transferência, a representação em espaço de estados possui um modelo discretizado, na qual para a equação (2.11) deve-se substituir  $x$  por  $x(k)$ ,  $u$  por  $u(k)$ ,  $y$  por  $y(k)$  e a derivada do estado  $\dot{x}$  por  $x(k+1)$  (PHILLIPS; NAGLE, 1995).

$$\begin{aligned}
 x(k+1) &= A_d x(k) + B_d u(k) \\
 y(k) &= C_d x(k) + D_d u(k)
 \end{aligned} \tag{2.11}$$

A partir das equações no espaço de estado de um sistema dinâmico com entrada e saída única, é possível encontrar sua Função de Transferência utilizando a relação matemática (2.12) como apresentado em (DORF; BISHOP, 2009).

$$G(s) = C(sI - A)^{-1}B + D \tag{2.12}$$

No mesmo contexto, é possível encontrar as equações no espaço de estado de um sistema dinâmico, utilizando sua representação através de Funções de Transferência (OGATA, 2003).

#### 2.1.4 Base de Funções Ortonormais

Um processo linear estável pode ser caracterizado por sua resposta ao impulso  $h(k)$ , com  $k = 0, 1 \dots \infty$ . Supondo que o processo seja causal e que  $h(k)$  pertença ao espaço das funções quadraticamente somáveis (Espaço de Lebesgue  $L_2[0, \infty)$  no caso contínuo),

então a resposta ao impulso  $h(k)$  pode ser modelada através de uma série de funções ortonormais (OLIVEIRA; CAMPELLO; AMARAL, 1999):

$$h(k) = \sum_{i=1}^{\infty} c_i \phi_i(k) \quad (2.13)$$

com

$$c_i(k) = \sum_{k=0}^{\infty} h(k) \phi_i(k) \quad (2.14)$$

na qual  $\phi_i(k), i = 1, 2, \dots, \infty$  é uma base de funções ortonormais e  $c_i$  são os coeficientes associados ao desenvolvimento em série da resposta ao impulso  $h(k)$ .

A utilização das bases de funções ortonormais vem sendo aplicadas na identificação de sistemas dinâmicos há algumas décadas, como por exemplo em (WIENER, 1949) e nos últimos anos com (MACHADO, 2007; CAMPELLO; OLIVEIRA; AMARAL, 2007; OLIVEIRA, 1997; ROSA; CAMPELLO; AMARAL, 2009). A representação utilizando BFOs apresenta vantagens quando comparados com outros modelos normalmente utilizado para representação dos mesmos sistemas (HENRIQUES; MACHADO; FERREIRA, 2013):

- A utilização BFOs com polo(s) próximo(s) à dinâmica dominante do sistema requer uma menor quantidade de termos para representar o sistema com a mesma precisão de um modelo FIR;
- Não há a necessidade de determinar previamente a ordem do modelo nem o atraso de transporte, eliminando-se assim a etapa (geralmente árdua) de determinação dos regressores;
- É possível aumentar a capacidade de aproximação dos modelos simplesmente aumentando-se o número de funções, o que pode ser feito de forma adaptativa;
- Os modelos BFO apresentam grande tolerância às dinâmicas não modeladas e sensibilidade reduzida aos parâmetros estimados (dinâmicas não dominantes);
- Capacidade de lidar de forma robusta com atrasos de transporte;
- Desacoplamento natural das saídas em sistemas multivariáveis;
- É garantido que a representação de um sistema estável é também estável;
- Ausência de realimentação de saída, o que, conseqüentemente, leva a modelos mais precisos e evita a propagação de erro na predição de sinais.

Duas BFOs amplamente utilizadas para modelagem de sistemas dinâmicos são as bases de Kautz e Laguerre (WAHLBERG, 1994). A base de funções ortonormais de Laguerre é caracterizada pela utilização de funções de transferência com apenas um polo real (MACHADO, 2007):

$$\Phi(z) = \frac{\sqrt{1-p^2}}{z-p} \left( \frac{1-pz}{z-p} \right)^{i-1} \quad (2.15)$$

na qual  $p = \{p : p \in R \text{ e } |p| < 1\}$  é um polo estável que parametriza as funções ortonormais,  $i$  a ordem da função e  $z^{-1}$  o operador de atraso no tempo. Pode-se observar que os modelos com funções de Laguerre podem ser caracterizados como modelos equivalentes FIR com polos fora da origem (CAMPELLO; OLIVEIRA; AMARAL, 2007; OLIVEIRA, 1997). É possível representar um modelo de Laguerre de ordem  $i$  através de modelos em cascata  $i - 1$ , como mostra a Figura 1.

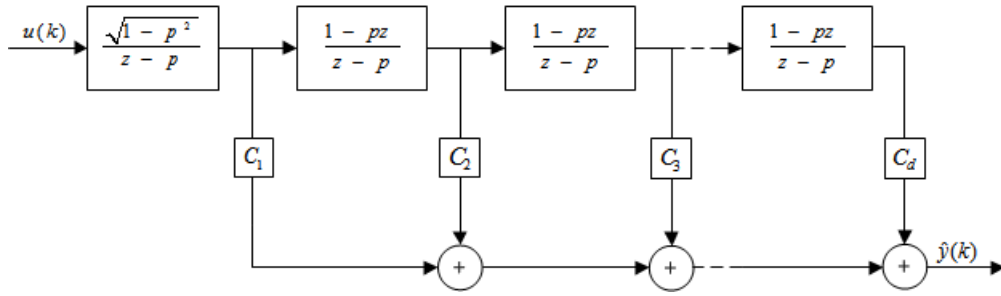


Figura 1: Representação Gráfica das Funções de Laguerre.

A base de funções de Kautz é mais indicada para representar funções de transferência parametrizadas por um par de polos complexos conjugados  $\beta = \alpha \pm j\omega$ , segundo as equações (2.16) e (2.17):

$$\Phi_{2i-1}(z) = \frac{\sqrt{(1-c^2)(z-b)}}{z^2 + b(c-1)z - c} \left( \frac{-cz^2 + b(c-1)z + 1}{z^2 + b(c-1)z - c} \right)^{i-1} \quad (2.16)$$

$$\Phi_{2i}(z) = \frac{\sqrt{(1-b^2)(1-c^2)}}{z^2 + b(c-1)z - c} \left( \frac{-cz^2 + b(c-1)z + 1}{z^2 + b(c-1)z - c} \right)^{i-1} \quad (2.17)$$

nas quais,

$$b = (\beta + \beta^*) / (1 + \beta\beta^*) \text{ e } c = -\beta\beta^* \quad (2.18)$$

sendo  $0 < b < 1$ ,  $-1 < c < 0$  e  $\beta\beta^*$  o par de polos complexos conjugados. Por apresentarem a propriedade de recursividade, ou seja, a  $i - \text{ésima}$  amostra pode ser escrita em função da  $(i - 1) - \text{ésima}$  amostra, as funções de base ortonormal podem ser representadas



através de equações de estado (MACHADO, 2007), como apresentado na equação (2.19).

$$\begin{aligned}\Psi(k+1) &= A_f \Psi(k) + B_f u(k) \\ \hat{y}(k) &= C_f^T \Psi(k)\end{aligned}\quad (2.19)$$

na qual  $\Psi(k) = [\Psi_1(k) \dots \Psi_d(k)]^T$  é o vetor de estados ortonormais<sup>1</sup> de ordem  $d$ , sendo esses estados coincidentes com as saídas dos filtros correspondentes às funções ortonormais de ordem equivalente. O modelo em questão, pode incluir uma parcela estática da representação que modela o nível contínuo do sistema sob estudo (MACHADO, 2011). É de especial interesse neste trabalho a modelagem de sistemas para controle preditivo utilizando a representação variacional em espaço de estados como apresentado na equação (2.20).

$$\begin{aligned}\Delta \Psi(k+1) &= A_f \Delta \Psi(k) + B_f \Delta u(k) \\ \hat{y}(k) &= \hat{y}(k-1) + C_f^T \Delta \Psi(k)\end{aligned}\quad (2.20)$$

As matrizes  $A_f$  e  $B_f$  da representação em espaço de estados mostradas na equação (2.20), dependem somente das bases de funções ortonormais. Para as bases de funções de Laguerre, tem-se (OLIVEIRA, 1997):

$$A_f = \begin{bmatrix} p & 0 & 0 & \cdots & 0 \\ 1-p^2 & p & 0 & \cdots & 0 \\ (-p)(1-p^2) & 1-p^2 & p & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (-p)^{d-2}(1-p^2) & (-p)^{d-3}(1-p^2) & \cdots & p \end{bmatrix}$$

$$B_f = \sqrt{1-p^2} \begin{bmatrix} 1 & -p & (-p)^2 & \cdots & (-p)^{d-1} \end{bmatrix}$$

sendo  $p$  o polo das bases de Laguerre,  $d$  o número de filtros de Laguerre e os elementos da matriz  $C$  os coeficientes da expansão da série, que podem ser calculados utilizando métodos clássicos como o método dos mínimos quadrados (MACHADO, 2007). A Base de Kautz também possui uma representação matricial como a base de Laguerre.

---

<sup>1</sup>"Estados ortonormais" é uma denominação usada por conveniência, já que a ortonormalidade é uma propriedade da função e não dos estados

Já para representar sistemas dinâmicos com mais de dois polos distintos, recomenda-se utilizar modelos formados por base de funções ortonormais generalizadas, na qual as funções podem apresentar  $n_b$  polos distintos (HEUBERGER; Van den Hof; WAHLBERG, 2005; MACHADO, 2011) como apresentado na Seção a seguir.

### 2.1.5 Base de Funções Ortonormais Generalizadas

Como apresentado nos trabalhos de Ninness, Hjalmarsson e Gustafsson (1999) e Heuberger, Van den Hof e Wahlberg (2005), a classe das funções ortonormais generalizadas (GOBF - *Generalized Orthonormal Basis Functions*) é obtida pela conexão de  $n$  filtros *all-pass* de  $n_b$ -ésima ordem, ou seja, cada filtro terá  $n_b$  polos distintos ou não, podendo ser de qualquer natureza, reais ou complexos (MACHADO, 2011). A Figura (2) apresenta a estrutura em blocos de um modelo de bases de funções ortonormais generalizadas.

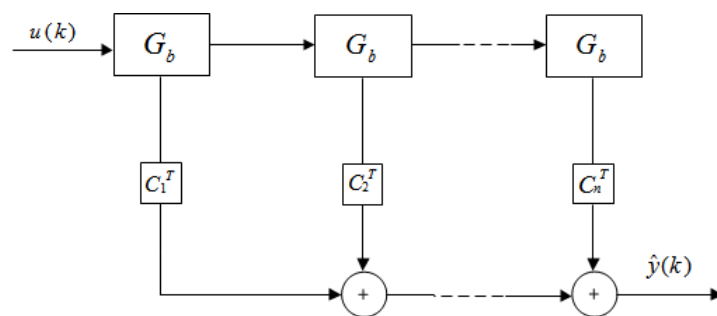


Figura 2: Representação Gráfica de um modelo GOBF.

Os blocos representados por  $G_b$  são as funções ortonormais que compõem base e  $C_i^T$ , com  $i = 1, \dots, n$ , são os coeficientes que multiplicam as saídas das funções para formar a saída  $\hat{y}(k)$ .

Um caso especial de GOBF, consiste em conectar diferentes filtros *all-pass* de primeira e segunda ordem, podendo cada filtro apresentar polo(s) distintos(s). Essa base de funções é conhecida por função de Takenaka-Malmquist (HEUBERGER; Van den Hof; WAHLBERG, 2005):

$$F_k(z) = \frac{\sqrt{(1 - |\xi_k|^2)^{k-1}}}{z - \xi_k} \prod_{i=1}^{k-1} \left[ \frac{1 - \xi_i z}{z - \xi_i} \right] \quad (2.21)$$

na qual  $\xi_i, i = 1, \dots, k$  são polos do modelo GOBF. Modelos GOBF com polos complexos conjugados podem ser combinados, correspondendo as funções de Kautz, porém este tipo de abordagem apresenta algumas desvantagens quando as características da dinâmica do sistema são desconhecidas (MACHADO, 2011):

- É necessário se especificar a priori o número de polos reais e polos complexos conjugados do sistema;
- A parametrização não é única, uma vez que qualquer permutação de polos levará a modelos idênticos;
- Para determinação dos parâmetros no caso de par de polos complexos conjugados (partes reais e imaginárias), há uma interdependência dos parâmetros pois os polos devem estar dentro do círculo unitário, ou seja, ter polos com módulo menor que a unidade (HEUBERGER; Van den Hof; WAHLBERG, 2005);

Uma outra representação GOBF que não apresenta os problemas descritos acima, foi proposta por Gray e Markel (1975) e é conhecida como base de funções ortonormais generalizadas com funções internas de estrutura *ladder*, sendo os modelos de Kautz e Laguerre casos especiais das Bases de Funções Ortonormais Generalizadas com funções internas.

### 2.1.6 Funções Internas

Uma função de transferência racional  $G(z)$  estável e que satisfaz a equação (2.22) é considerada interna, ou seja, um filtro *all-pass*.

$$G(z)G\left(\frac{1}{z}\right) = 1 \quad (2.22)$$

Um exemplo claro de função interna  $G(z)$  com polo real  $\xi$  que satisfaz a equação (2.22) é dado pela equação (2.23):

$$G(z) = \frac{1 - \xi_m z}{z - \xi_m} \quad (2.23)$$

A equação (2.23) representa um modelo com polo real e é equivalente à representação com funções de Laguerre (HEUBERGER; Van den Hof; WAHLBERG, 2005). Sistemas dinâmicos com par de polos complexos conjugados, modelo equivalente à Kautz, também possuem representação interna como apresentado na equação (2.24), sendo os polos complexos conjugados  $\xi_m$  e  $\xi_m^*$ .

$$G_1(z) = \frac{1 - \xi_m^* z}{z - \xi_m} \quad (2.24)$$

$$G_2(z) = \frac{1 - \xi_m z}{z - \xi_m^*} \quad (2.25)$$

A conexão de  $n_b$  funções internas gera uma função com ordem  $n_b$ , porém se as funções envolverem polos complexos conjugados os parâmetros normalmente se tornam complexos. Desta forma, Gray e Markel (1975) propuseram uma representação para modelos GOBF através de funções internas que sempre apresentam parâmetros reais  $\gamma$ , independente da natureza dos polos. Esta representação em espaço de estados para uma função interna com somente um polo real, duas entradas e duas saídas, conhecida como estrutura *ladder* é dado por (HEUBERGER; Van den Hof; WAHLBERG, 2005):

$$\begin{bmatrix} \varphi_i(k+1) \\ y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \sqrt{1-\gamma^2} & -\gamma & 0 \\ \gamma & \sqrt{1-\gamma^2} & 0 \end{bmatrix} \begin{bmatrix} \varphi_i(k) \\ u_1(k) \\ u_2(k) \end{bmatrix} \quad (2.26)$$

sendo  $-1 < \gamma < 1$ . O modelo da equação (2.26) pode ser representado através do diagrama em blocos como apresentado na Figura (3).

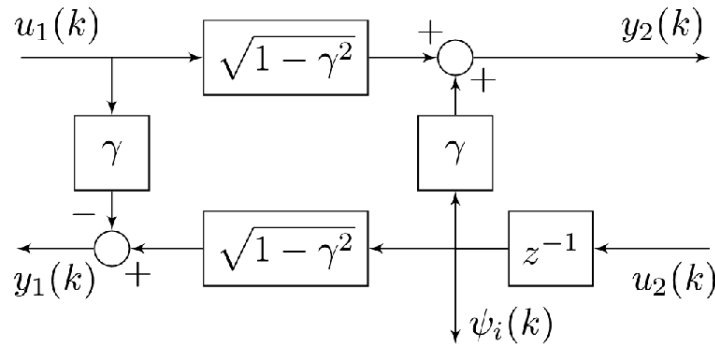


Figura 3: Representação em blocos da estrutura interna parametrizada por  $\gamma$ .

A ligação de  $n_b$  blocos distintos formam uma base ortonormal generalizada com  $n_b$  polos, que podem ser distintos ou não, é obtida através das ligações apresentadas na Figura (4).

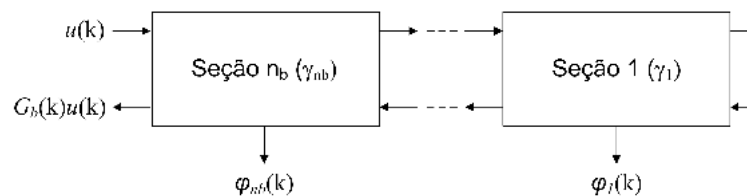


Figura 4: Representação em blocos das funções da base de uma GOBF.

Para representar a estrutura GOBF com funções internas foi desenvolvido no trabalho de Machado (2011) um modelo em espaço de estados para obter as saídas das funções

na base, como apresentado na equação (2.27), considerando um modelo com somente um bloco  $G_b$  com  $n_b$  funções internas parametrizadas pelos parâmetros  $\gamma_i$  com  $i = 1, \dots, n_b$ .

$$\begin{aligned}\Psi_b(k+1) &= A_b \Psi_b(k) + B_b u(k) \\ \hat{y}(k) &= C_b \Psi_b(k)\end{aligned}\tag{2.27}$$

na qual  $\Psi_b(k) = [\psi_1(k) \dots \psi_{n_b}(k)]$  são as saídas dos blocos que compõe o bloco  $G_b$  da GOBF. As matrizes  $A_b$  e  $B_b$  da equação (2.27) são obtidas diretamente a partir dos parâmetros  $\xi_i$  com  $i = 1, \dots, n_b$  como apresentado em (MACHADO, 2011). Os parâmetros da matriz  $C$  podem ser obtidos pelo método dos mínimos quadrados levando-se em consideração as saídas  $y(k)$  medidas do sistema a ser modelado e as saídas das funções ortonormais  $\Psi(k)$  geradas pela entrada  $u(k)$ .

Para a modelagem de sistemas dinâmicos utilizando base de funções ortonormais, os parâmetros a serem determinados são os polos das funções ortonormais e os coeficientes da expansão da série de funções. Uma vez conhecido os polos do sistema, é possível determinar os termos da expansão da série de funções utilizando o Método do Mínimos Quadrado (MACHADO, 2007; CAMPELLO; OLIVEIRA; AMARAL, 2007; CAMPELLO; AMARAL, 2002).

Para a seleção adequada do(s) polo(s) da base de funções ortonormais é necessário um conhecimento a priori do sistema ou a utilização de métodos de otimização que minimizem o erro do modelo estimado, como apresentado em (HEUBERGER; Van den Hof; WAHLBERG, 2005; NELLES, 2001; ROSA; CAMPELLO; AMARAL, 2009). É de especial interesse neste trabalho, a otimização dos parâmetros do modelo BFO, utilizando métodos de otimização não-linear apresentado no trabalho de Machado (2011).

## 2.2 Modelagem de Sistemas Não-Lineares

Quando analisados com rigor, a maioria dos sistemas dinâmicos encontrados na indústria são sistemas não-lineares. Em muitos casos, a representação linear é suficiente para representar tais sistemas, porém quando é necessária uma representação mais realista do processo sob estudo, é necessário utilizar técnicas não-lineares de representação (AGUIRRE, 2004). A seguir, serão apresentadas algumas técnicas de representação não-linear de sistemas dinâmicos.

### 2.2.1 Série de Volterra

Uma maneira de representar o modelo de um sistema dinâmico não-linear é através das séries de Volterra. No trabalho de Boyd e Chua (1985) foi demonstrado que o modelo de Volterra com resposta ao impulso pode aproximar, com precisão arbitrária, qualquer sistema a ser modelado que admita descrição entrada/saída através do operador  $H$  contínua e com *fading memory*, desde que a entrada  $u$  tenha limites superiores e inferiores (MACHADO, 2011) como apresentado na equação (2.28) (AGUIRRE, 2004).

$$y(t) = \sum_{j=1}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_j(\tau_1, \dots, \tau_j) \prod_{i=1}^j u(t - \tau_i) d\tau_i \quad (2.28)$$

sendo que as funções  $h_j$  são os *kernels* (núcleos) de Volterra que são generalizações não-lineares da resposta ao impulso  $h(t)$ .

Para sistemas lineares, a representação de Volterra é uma generalização da representação da resposta ao impulso. O modelo de Volterra discreto é apresentado na equação (2.29).

$$y(t) = \sum_{m=1}^M \sum_{k_1=0}^{\epsilon_m} \dots \sum_{k_m=0}^{\epsilon_m} h_m(k_1, k_2, \dots, k_m) \prod_{j=1}^m u(t - k_j) \quad (2.29)$$

sendo  $u$  é a entrada,  $y$  a saída e  $h_m$  um *kernel* de  $m$ -ésimo grau.

A grande desvantagem de utilizar a representação de Volterra para sistemas não-lineares, é a estimação de um grande número de parâmetros, isso se dá pois a série de Volterra tenta reproduzir a saída  $y(t)$  somente em função da entrada  $u(t)$  (AGUIRRE, 2004).

### 2.2.2 Modelos de Hammerstein e de Wiener

Dois modelos bastantes utilizados para representação de modelos não-lineares são os modelos de Hammerstein e de Wiener. Estes modelos têm como característica ser uma composição em cascata de um modelo dinâmico linear  $H(s)$  e um modelo estático não-linear. No modelo de Hammerstein o modelo estático não-linear precede a componente dinâmica linear, como apresentado na Figura 5.

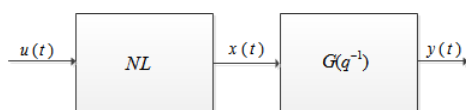


Figura 5: Modelo de Hammerstein.

O Bloco de não-linearidade (NL) pode ser representado por um polinômio, pela equação de não-linearidade ou por modelos paramétricos (SANTOS, 2007). Para a representação de Wiener, o modelo dinâmico linear vem seguido do modelo não-linear, como apresentado na Figura 6.

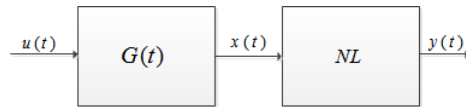


Figura 6: Modelo de Wiener.

Os modelos de Hammerstein e Wiener também apresentam representações discretas, como apresentado em (AGUIRRE, 2004; SANTOS, 2007).

### 2.2.3 Modelo NOBF - *Nonlinear Orthonormal Basis Function*

Um sistema dinâmico genérico, não-linear, pode ser representado como:

$$y(k) = H(\lambda(k)) \quad (2.30)$$

na qual  $\lambda$  é o vetor de estados  $\lambda(k) = [l_1(k)l_2(k) \dots l_d(k)]$  formado pela saída dos filtros ortonormais e  $H$  é o mapeamento estático constituído pela combinação linear desses estados (MACHADO, 2007). A utilização do mapeamento estático  $H$  em combinação com as bases de funções ortonormais gera um NOBF.

Para representação de sistemas não-lineares, pode-se substituir o mapeamento estático  $H$  por um mapeamento não-linear. Desta forma, o modelo seria descrito por uma dinâmica linear que relaciona a entrada  $u(k)$  aos estados ortonormais  $l_i(k)$ , seguida por de um mapeamento estático entre esses dados e a saída  $\hat{y}(k)$ , ou seja, um modelo do tipo Wiener. O mapeamento estático  $H$  pode ser desenvolvido através de séries de Volterra, Combinação Linear, modelo *fuzzy* TS ou rede neural, como apresentado no trabalho de Campello, Oliveira e Amaral (2007) Campello.

Na representação por modelos de Volterra, o operador  $H$  é representado por um polinômio multidimensional, sendo que a complexidade do modelo aumenta exponencialmente com a ordem do polinômio. Neste contexto, as representações utilizando aproximadores universais, *fuzzy* e redes neurais, tem vantagens sobre a representação multidimensional, pois mesmo a quantidade de regras *fuzzy* ou de neurônios aumentando exponencialmente com a dimensão do domínio, é possível aumentar a precisão do mo-

delo através de ajustes paramétricos dos conjuntos *fuzzy* ou funções de ativação neurais (CAMPELLO; OLIVEIRA; AMARAL, 2007).

#### 2.2.4 Identificação de Sistemas Dinâmicos

Por definição, identificar um sistema dinâmicos significa obter um modelo matemático que represente com fidelidade o sistema sob estudo através de amostras de saída/entrada. Na Seção 2.1.1 demonstrou-se como obter o modelo matemático de um sistema a partir de conceitos físicos de sua natureza, a chamada modelagem fenomenológica ou "caixa branca"(SOUZA; PINHEIRO, 2008). No entanto, em muitos sistemas industriais, não se tem conhecimento das partes que constituem o sistema dinâmicos, fazendo com que a modelagem através dos conceitos físico se torne muito complexa. Neste sentido, a obtenção do modelo matemático através de procedimentos de identificação de sistemas se torna muito vantajosa, pois teoricamente não é necessário nenhum conhecimento a priori da dinâmica do sistema. Segundo Aguirre (2004), as principais etapas de um problema de identificação são:

- Coleta de Dados do Processo - Dados da entrada/saída do sistema a ser modelado são coletados.
- Escolha do Modelo Matemático - Conforme as características do sistema a ser modelado, a estrutura do modelo (ARX, ARMAX, Box-Jenkins, dentre outras) é escolhida.
- Determinação da Estrutura do Modelo - Consiste na escolha dos parâmetros do modelo do sistema (SOUZA; PINHEIRO, 2008).
- Estimação de Parâmetros - Através de procedimentos numéricos, os coeficientes do modelo são obtidos.
- Validação do Modelo - Tendo obtido o modelo, é necessário verificar se este incorpora as características do sistema original.

A qualidade do modelo obtido através de identificação de sistemas depende diretamente da qualidade dos dados amostrados, da estrutura a ser utilizada e da seleção correta de seus parâmetros. Dentre as ferramentas encontradas na literatura para estimação dos parâmetros do modelo, as ferramentas que utilizam dados do processo se mostram muito



eficientes e são utilizadas com bastante frequência para modelagem de sistemas reais (LJUNG, 1999).

Para processos modelados utilizando equações ou sistemas de equações diferenciais, muitas ferramentas para obtenção dos parâmetros dos modelos são encontradas na literatura. Quando se utiliza representações através de funções de transferência de resposta ao impulso contínuas no tempo, algumas técnicas para obtenção dos parâmetros do modelo podem se basear em informações da resposta ao degrau do sistema (AGUIRRE, 2004) ou ainda, em informações da resposta em frequência do sistema (LJUNG, 1999). No caso de funções de transferência discretizadas no tempo (domínio  $z$ ) os parâmetros do modelo podem ser obtidos através do Estimador de Mínimos Quadrados (MQ). Este método também é utilizado para identificação de representações em espaço de estados discretizadas (MACHADO, 2007).

O estimador de mínimos quadrados é um dos métodos mais conhecidos e utilizados em diversas áreas da ciência e a idéia original do método pode ser encontrada nos estudos astronômicos de Gauss. O método dos mínimos quadrados minimiza de forma quadrática o erro entre as medidas obtidas do sistema real e os valores estimados pelo modelo, e através desta metodologia fornece os parâmetros do modelo (para maiores detalhes consulte o Apêndice A). O estimador de mínimos quadrados apresenta as seguintes propriedades, quando a perturbação agindo no processo em malha aberta é branca e aditiva (LJUNG, 1999; MACHADO, 2007).

- Não polarizado, ou seja, o valor esperado (média) do erro de estimação é nulo;
- Melhor estimador linear não polarizado BLUE <sup>2</sup>, implicando que ele minimiza a variância do erro de estimação;
- Consistente, pois a variância do erro de estimação tende a zero na medida que o número  $N$  de medições tende a infinito;

Estas propriedades fazem do método de mínimos quadrados muito atrativo para estimação dos parâmetros de modelos de sistemas reais.

---

<sup>2</sup>BLUE - Best Linear Unbiased Estimator

### 2.2.5 Resumo

Neste capítulo foi apresentado uma introdução a modelagem de sistemas dinâmicos utilizando equações diferenciais, funções de transferência, representação em espaço de estados e base de funções ortonormais, sendo esse último modelo de maior interesse nesta dissertação. Foram apresentados também algumas realizações para sistemas não-lineares como as séries de Volterra, o modelo de Hamerstein e Wiener e os modelos NOBF. Dentre as realizações para modelagem de sistemas não-lineares, a modelagem utilizando *fuzzy* BFO é de especial interesse neste trabalho e será apresentada com detalhes no próximo capítulo.

No processo de identificação de sistemas dinâmicos, cujos parâmetros do modelo precisam ser estimados, o estimador de mínimos quadrados é um dos métodos mais utilizados, e de especial interesse neste trabalho.

## 3 Modelagem de Sistemas Não-Lineares por Fuzzy-GOBF

O conceito de conjuntos *fuzzy*, também conhecida como teoria de conjuntos nebulosos, foi proposto por Zadeh (1965) como uma generalização da idéia de conjuntos ordinários ou *crisp* (não *fuzzy*). Atualmente a teoria de conjuntos fuzzy vem sendo utilizada em diversos segmentos de mercado como controle industrial, medicina, mercado financeiro, aviação e etc. Neste trabalho, os conjuntos *fuzzy* serão utilizadas para modelagem de sistemas dinâmicos não-lineares utilizando base de funções ortonormais generalizadas nos consequentes das regras dos modelos. Os modelos *fuzzy* apresentam a característica de descrever sistemas de forma aproximada, porém com precisão arbitrária, ou seja, como aproximadores universais de funções numa região compacta, apresentando a capacidade de mapear de forma eficiente relações de entrada e saída de sistemas em geral (WANG; MENDEL, 1992a; KOSKO, 1994b).

Este capítulo apresenta uma breve introdução a conjuntos *fuzzy* (Seção 3.1) e descreve um método iterativo para a determinação das funções de pertinência trapezoidais utilizadas nos antecedentes das regras *fuzzy*. Além disso na Seção 3.3.4 será apresentado uma metodologia para a determinação no número de grupos, e por consequência o número de regras do modelo *fuzzy* TS. Por fim, nas Seções 3.4.3 e 3.5 são detalhadas a metodologia de simplificação e otimização dos parâmetros do modelo *fuzzy* TS-GOBF.

### 3.1 Introdução aos Conjuntos *Fuzzy*

Um conjunto *fuzzy* é definido como um conjunto de objetos em que o grau de pertinência pode pertencer ao intervalo contínuo entre 0 e 1,  $I = [0, 1]$ , ao contrário dos conjuntos ordinários em que o elemento pertence ou não pertence ao conjunto  $\{0,1\}$ . Formalmente, um conjunto  $A$  do universo de discurso  $U$  é definido por uma função de pertinência  $\mu_A$

que mapeia os elementos do universo de discurso no intervalo contínuo  $[0, 1]$ .

$$\mu_A : U \rightarrow [0, 1] \quad (3.1)$$

Neste contexto, a função  $\mu_A(x)$  indica com que grau o elemento  $x$  pertence ao conjunto  $A$ , ou seja, o grau de compatibilidade entre  $x$  e o conceito expresso por  $A$ , como apresentado em (SANDRI; CORREA, 1999):

- $\mu_A(x) = 1$  indica que  $x$  é completamente compatível com  $A$ ;
- $\mu_A(x) = 0$  indica que  $x$  é completamente incompatível com  $A$ ;
- $0 < \mu_A(x) < 1$  indica que  $x$  é parcialmente compatível a  $A$ , com grau  $\mu_A(x)$ ;

O conceito de variável linguística é fundamental quando se trabalha com conjuntos *fuzzy*. Uma variável linguística  $T(u)$  em um universo de Discurso  $U$  é definida em um conjunto de termos, nomes ou terminologias, podendo assumir um valor linguístico dentre vários outros, em um conjunto de termos linguísticos. Por exemplo, sendo  $u$  velocidade, o conjunto de termos  $T(u)$  poderia ser formado por  $T(\text{velocidade}) = \{\text{muito baixa, baixa, media, rápida, muito rápida}\}$  (SIMOES; SHAW, 2007). Por definição, função de pertinência é uma função numérica gráfica ou tabulada, que atribui valores de pertinência *fuzzy* para valores de uma variável em seu universo de discurso. As funções mais utilizadas na literatura são funções trapezoidais, triangulares, gaussiana ou ainda do tipo *singleton* (MACHADO, 2007; SIMOES; SHAW, 2007). A Figura 8 mostra funções de pertinência do tipo trapezoidal que são de especial interesse neste trabalho.

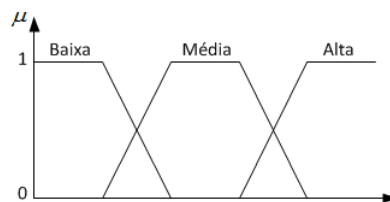


Figura 7: Funções de Pertinência Trapezoidal.

Em sistemas *fuzzy*, a conversão das grandezas do mundo real provenientes de sensores industriais, sistemas computadorizados ou até mesmo de outras etapas do sistema de controle para números *fuzzy* é chamada de *fuzzyficação*.

A Inferência em sistemas *fuzzy* consiste em avaliar um conjunto de regras, conhecido como base de conhecimento, que descrevem a dependência entre as variáveis linguísticas de entrada e saída do sistema. Os antecedentes das regras *fuzzy* possuem uma estrutura geral da forma:

$$\begin{aligned} &\text{Se } x_1 \text{ é } A_{1,1} \text{ e } x_2 \text{ é } A_{1,2} \text{ e } \dots \text{ e } x_d \text{ é } A_{1,d} \\ &\text{Se } x_1 \text{ é } A_{2,1} \text{ e } x_2 \text{ é } A_{2,2} \text{ e } \dots \text{ e } x_d \text{ é } A_{2,d} \\ &\quad \vdots \\ &\text{Se } x_1 \text{ é } A_{k,1} \text{ e } x_2 \text{ é } A_{k,2} \text{ e } \dots \text{ e } x_d \text{ é } A_{k,d} \end{aligned}$$

sendo  $x_1, \dots, x_d$  as entradas do sistema e  $A_{1,1}, \dots, A_{k,d}$  os termos linguísticos dessas variáveis definidos como funções de pertinência, que cobrem todo o universo de discurso (MACHADO, 2007).

As regras de sistemas *fuzzy* são ativadas em paralelo, ou seja, quando uma entrada é fornecida, ela aciona várias regras com diferentes graus de ativação para inferir um resultado ou uma saída. A tradução da combinação das diferentes regras ativadas é conhecida como *defuzzificação*. Dentre os métodos mais utilização para *defuzzificação* pode-se citar o método do centro de área, o método da média dos máximos e o método do centro dos máximos (SANDRI; CORREA, 1999; MACHADO, 2007; SIMOES; SHAW, 2007).

Os diferentes tipos de estruturas *fuzzy* encontradas na literatura são os modelos clássicos de Mandami e o de Larsen e os modelos de interpolação formados pelos modelos de Takagi-Sugeno e Tsukamoto (SANDRI; CORREA, 1999). É de especial interesse neste trabalho o modelo introduzido por Takagi-Sugeno (1985), que na literatura referente à modelagem *fuzzy*, é conhecido como modelo Takagi-Sugeno (TS).

## 3.2 Modelos Fuzzy TS-BFO

Na modelagem de sistemas dinâmicos não-lineares, o modelo *fuzzy* Takagi-Sugeno vem sendo utilizado com considerável frequência, e consiste na interpolação de múltiplos modelos locais lineares. Esta abordagem é aplicada também no projeto de controladores e como aproximador universal de funções. O modelo TS baseado em regras apresenta a forma demonstrada a seguir (MACHADO, 2011):

$$\underbrace{\text{Se } x_1 \text{ é } A_{i,1} \text{ e } \dots \text{ e } x_j \text{ é } A_{i,j}}_{\text{antecedente}} \text{ então } \underbrace{y_i = f_i(x)}_{\text{consequente}} \quad (3.2)$$

A primeira parte da regra, conhecida como antecedente, é definida como uma proposição de combinações *fuzzy* "Se  $x$  é  $A_i$  e ... e  $x_j$  é  $A_{i,j}$ ", sendo  $x_j$  uma variável real não *fuzzy* e  $A_{i,j}$  um conjunto *fuzzy* definido pela função de pertinência

$$A_{i,j}(x) : \mathbf{R} \rightarrow [0, 1]$$

na qual  $i = 1, \dots, K$  representa a  $i$ -ésima regra, sendo  $K$  o número de regras, e  $j = 1, \dots, \varsigma$ , com  $\varsigma$  o número de variáveis nos antecedentes das regras. O grau de desempenho  $\mu_i$  para um dado valor no vetor de regressão  $x$  é avaliado como o grau de ativação da função de pertinência no conjunto  $A_{i,j}$  (SIMOES; SHAW, 2007). O antecedente é usualmente descrito pela combinação dos conjuntos *fuzzy* em cada sub-espaco univariável definido para cada componente de  $x$ . Os antecedentes são combinados utilizando conectivos lógicos **E**. O grau de ativação da  $i$ -ésima regra,  $\mu_i(x)$  é dada por:

$$\mu_i(x) = A_{i,1}(x_1) \wedge A_{i,2}(x_2) \wedge \dots \wedge A_{i,\varsigma}(x_\varsigma) \quad (3.3)$$

na qual o operador ( $\wedge$ ) é utilizado para conjunção. Para o operador produto, tem-se:

$$\mu_i(x) = A_{i,1}(x_1) \cdot A_{i,2}(x_2) \cdot \dots \cdot A_{i,\varsigma}(x_\varsigma) \quad (3.4)$$

A segunda parte da regra, denominada de consequente, é, em geral, uma função das variáveis da premissa. Normalmente, as regras se diferenciam pelos conjuntos *fuzzy* dos antecedentes e também pelas funções  $f_i(x)$  dos consequentes. Foi proposto por Sugeno e seus colaboradores (TAKAGI; SUGENO, 1985) a utilização de funções afins nos consequentes das regras como apresentado na equação (3.5):

$$\text{Se } x \text{ é } A_{i,j} \text{ então } y_i = a_i^T x + b_i \quad (3.5)$$

na qual  $a_i$  é um vetor de parâmetros e  $b_i$  é o *offset*. Este modelo é denominado TS afim. Quando  $b_i = 0, i = 1, \dots, k$ , o modelo é conhecido como TS linear.

Para a equação (3.5) utilizou-se funções lineares e funções afins, porém modelos *fuzzy* TS podem ser formados por qualquer tipo de função no consequente das regras, como funções não-lineares ou ainda funções dinâmicas como equações diferenciais ou equações a diferenças (MACHADO, 2011). Neste trabalho serão utilizadas as funções de Base Ortonormal Generalizada com funções internas do tipo Ladder no consequente das regras do modelo *fuzzy* TS. Nos casos em que se utiliza a mesma BFO para os modelos locais,

é necessário calcular somente a parte estática da formulação de cada modelo, visto que a parte dinâmica é idêntica (OLIVEIRA; CAMPELLO; AMARAL, 1999; CAMPELLO; AMARAL, 2002; CAMPELLO; MELEIRO; AMARAL, 2004; MACHADO, 2011). Desta forma, as regras dos modelos *fuzzy* TS-BFO podem ser definidas como apresentado a seguir:

$$\begin{aligned} \text{Se } \psi_1(k) \text{ é } A_{1,1} \text{ e } \psi_2(k) \text{ é } A_{1,2} \text{ e } \dots \text{ e } \psi_d(k) \text{ é } A_{1,d} \text{ então } y_1 &= C_1^T \Psi(k) \\ \text{Se } \psi_1(k) \text{ é } A_{2,1} \text{ e } \psi_2(k) \text{ é } A_{2,2} \text{ e } \dots \text{ e } \psi_d(k) \text{ é } A_{2,d} \text{ então } y_2 &= C_2^T \Psi(k) \\ &\vdots \\ \text{Se } \psi_1(k) \text{ é } A_{K,1} \text{ e } \psi_2(k) \text{ é } A_{K,2} \text{ e } \dots \text{ e } \psi_d(k) \text{ é } A_{K,d} \text{ então } y_K &= C_K^T \Psi(k) \end{aligned}$$

sendo  $\psi(k)^T = [\psi_1(k), \psi_2(k), \dots, \Psi_d(k)]$ ,  $d$  o número de estados da base BFO e  $K$  o número de regras do modelo *fuzzy*-TS.

A saída global do modelo *fuzzy*  $\hat{y}(k)$  é obtida através da interpolação clássica TS da saída dos filtros locais, como apresentado na equação (3.6) (MACHADO, 2011; NELLES, 2001).

$$\hat{y}(k) = \frac{\sum_{i=1}^K (\mu_i(k) C_i \psi(k))}{\sum_{i=1}^K \mu_i(k)} \quad (3.6)$$

na qual  $\mu_i(k)$  é o valor da ativação da  $i$ -ésima regra, como definido na equação (3.4),  $\psi(k)$  o valor da saída de cada modelo local BFO e  $K$  o número de regras ou modelos locais.

A utilização de modelos *fuzzy* TS-BFO tem se mostrado muito eficaz para modelagem e controle de sistemas dinâmicos (MACHADO, 2007; CAMPELLO; AMARAL, 2002; MACHADO, 2011; MEDEIROS; AMARAL; CAMPELLO, 2006). No entanto, as principais dificuldades em se utilizar modelos *fuzzy* TS-BFO em identificação de sistemas dinâmicos, estão em determinar o número  $K$  de regras *fuzzy* com seus antecedentes e consequentes, a sintonia dos conjuntos *fuzzy*  $A_{i,j}$  nas premissas das regras e a otimização do(s) polo(s) da BFO. Neste trabalho, a determinação do número de regras *fuzzy* e dos conjuntos que compõem os antecedentes das regras serão obtidos utilizando o algoritmo de Gustafson-Kessel juntamente com os critérios de avaliação de agrupamento, no qual é possível avaliar qual modelo *fuzzy* apresenta melhor desempenho na representação do sistema a ser modelado. O(s) polo(s) das funções ortonormais, os coeficientes da expansão da série das BFOs e os parâmetros das funções de pertinência presente nos antecedentes das regras também serão otimizados utilizando algoritmos de otimização não-linear apresentado no trabalho de Machado (2011).

### 3.3 Projeto de Modelos *fuzzy*-TS BFO

O projeto de modelos *fuzzy* TS-GOBF, envolvem algumas etapas. Dentre elas, pode-se destacar: determinação dos parâmetros do modelo inicial (número de polos da GOBF, número de funções e número de variáveis na premissa); determinação dos antecedentes das regras *fuzzy* através de agrupamento *fuzzy*; definição do número de grupos que melhor agrupe os dados; obtenção dos parâmetros dos modelos locais através da estimação por mínimos quadrados locais; simplificação da base de regras do modelo *fuzzy*; otimização dos parâmetros do modelo *fuzzy* TS-GOBF inicial. A seguir, serão detalhadas todas as etapas descritas para a obtenção do modelo *fuzzy* TS-GOBF.

#### 3.3.1 Agrupamento

Um conjunto de regras *fuzzy* pode ser obtido com informações fornecidas por um especialista no sistema em questão ou através da resposta do processo a um sinal de excitação utilizando, por exemplo, algoritmos de agrupamento *fuzzy* (*fuzzy clustering*). O intuito do agrupamento é identificar grupos naturais de dados, a partir de um grande número de dados, para produzir uma representação fidedigna do comportamento do sistema (EVERITT; LANDAU; LEESE, 2001). Os principais algoritmos de agrupamento podem ser classificados em três grupos, como apresentado a seguir (HELLENDOORN; DRIANKOV, 1997):

1. Algoritmos que utilizam uma medida de distância adaptativa, tais como Gustafson-Kessel (GUSTAFSON; KESSEL, 1979) ou algoritmos de estimação de máxima verossimilhança *fuzzy* (GATH; GEVA, 1989).
2. Algoritmos baseados em protótipos hiperplanares, também conhecidos como variedade linear *fuzzy* e *c*-elipsóide *fuzzy* (BEZDEK, 1981).
3. Algoritmos de *c*-regressão, os quais utilizam protótipos definidos por funções de regressão (HATHAWAY R. E BEZDEK, 1993).

Os algoritmos de agrupamento enumerados, podem ser vistos com uma extensão do algoritmo *c-means* (BEZDEK, 1981), que minimizam a variância dos dados para os protótipos (centros) dos grupos. Neste trabalho as informações dos grupos obtidos a partir do algoritmo de Gustafson-Kessel, detalhado a seguir, serão utilizadas para se obter modelos *fuzzy*-TS que representem com fidelidade o sistema sob estudo, utilizando o menor



número de grupos (regras/modelos locais) possível (BEZDEK, 1981; BABUSKA; VERBRUGGEN, 1996).

### 3.3.2 Algoritmo de Gustafson-Kessel

Um algoritmo de agrupamento *fuzzy* aplicado a um conjunto de dados medidos em um sistema com  $N$  objetos  $X = x_1, x_2, \dots, x_N$ , apresenta como resultado uma matriz de partição *fuzzy* desses objetos em um dado número  $K$  de grupos, tal que  $U = [\varpi_{i,j}]_{K \times N}$ , sendo  $U$  uma matriz de partição *fuzzy*  $K \times N$  cujos elementos  $\varpi_{i,j}$  apresentam o grau de pertinência do  $j$ -ésimo objeto ao  $i$ -ésimo grupo *fuzzy*.

A maioria dos algoritmos de agrupamento *fuzzy*, procura minimizar alguma extensão da função objetivo como é apresentada na equação (3.7) (VENDRAMIN, 2012).

$$\sum_{i=1}^K \sum_{j=1}^N \varpi_{i,j}^m D_{i,j} \quad (3.7)$$

com restrição  $\varpi_{i,j} \in [0, 1]$  é:

$$\sum_{j=1}^N \varpi_{i,j} = 1, \quad 1 \leq i \leq K, \quad (3.8)$$

sendo  $m > 1$  em (3.7) o parâmetro que controla a *fuzzificação* do grupo, tipicamente  $m = 2$ ,  $d(i, j)$  uma dada distância entre o  $j$ -ésimo objeto e a  $i$ -ésimo grupo. Esta é uma definição padrão de agrupamento com uma função objetivo *fuzzy* em que se deve minimizá-la através de um processo iterativo (BEZDEK, 1981). Gustafson e Kessel (1979) generalizaram o algoritmo de agrupamento para um método de distância adaptativa, como apresentado em (3.9):

$$d^2(z_j, v_i) = (z_j - v_i)^T M_i (z_j - v_i) \quad (3.9)$$

na qual  $M_i$  é uma matriz definida positiva adaptada de acordo com a forma atual do grupo e pode ser escrita em função da matriz de covariância do grupo,  $F_i$ :

$$F_i = \frac{\sum_{j=1}^N \varpi_{i,j}^m (z_j - v_i)(z_j - v_i)^T}{\sum_{j=1}^N \varpi_{i,j}^m} \quad (3.10)$$

A Matriz  $M_i$  utilizado em (3.9) é calculada como a matriz inversa normalizada da

matriz de variância do grupo, dada por:

$$M_i = \det(F_i)^{\frac{1}{d+1}} F_i^{-1} \quad (3.11)$$

Depois de calculadas as matrizes  $M_i$ , calculam-se as distâncias quadráticas  $d^2(z_j, v_i)$  e atualizam-se as matrizes de particionamento  $U$  através de seus elementos dados por:

$$\varpi_{i,j} = \frac{d^2(z_j, v_i)^{\frac{-1}{m-1}}}{\sum_{l=1}^K d^2(z_j, v_l)^{\frac{-1}{m-1}}} \quad (3.12)$$

na qual, se  $d^2(z_j, v_i) = 0$  para algum  $j = k$ , faça  $\varpi_{i,j} = 1$  e  $\varpi_i = 0$ ,  $\forall j \neq k$ . Os passos citados acima são repetidos até que a convergência seja atingida, isto é,  $\|U_L - U_{L-1}\| < \varepsilon$  sendo  $\|\cdot\|$  uma norma matricial,  $L$  é o passo da iteração e  $\varepsilon$  é uma tolerância adotada para o termino da otimização, com  $\varepsilon = 0,01$  ou  $\varepsilon = 0,001$  tipicamente, para dados normalizados (BABUSKA; VERBRUGGEN, 1997). Após a convergência do algoritmo GK, a matriz de particionamento  $U$ , os protótipos  $v_i$  dos grupos e as matrizes de covariância  $F_i, i = 1, 2, \dots, K$  são obtidos.

### 3.3.3 Determinação do Número de Regras

Algoritmos de agrupamento *fuzzy* normalmente requerem que o número de grupos  $K$  desejados pelo usuário seja previamente definido. Sendo assim, é importante determinar o menor número de grupos que apresente o melhor agrupamento dos dados do processo a ser identificado (MACHADO, 2011). Na literatura são encontrados diferentes critérios de avaliação para agrupamentos *fuzzy*, porém, cada um apresenta um resultado menos ou mais preciso, dependendo das características dos dados provenientes do problema.

Utilizando a metodologia proposta em Machado (2007), a quantidade ideal de grupos, para o modelo *fuzzy* TS-GOBF abordado neste trabalho, é obtida através de um conjunto de critérios que serão detalhados a seguir.

- **Critério *Fuzzy* Hipervolume**

Baseado no conceito que um bom agrupamento *fuzzy* é aquele que não apresenta grupos dispersos, Gath e Geva (1989) propuseram o critério de avaliação *Fuzzy* Hipervolume (FHV). Neste critério, as medidas de avaliação são baseadas no critério de hipervolume ou densidade do agrupamento *Fuzzy*, como apresentado na equação (3.13):

$$FHV = \sum_{i=1}^N [\det(F_i)]^{-\frac{1}{2}} \quad (3.13)$$

na qual  $F_i$  é a matriz de covariância do  $i$ -ésimo grupo *fuzzy*, dada por:

$$F_i = \frac{\sum_{j=1}^N h(i|z_j)(z_j - v_i)(z_j - v_i)^T}{\sum_{j=1}^N h(i|z_j)} \quad (3.14)$$

Quando o valor do parâmetro *fuzzificador* é escolhido  $m = 2$ , sendo  $h(i|z_j)$  a probabilidade de seleção do  $i$ -ésimo grupo dada a  $j$ -ésima amostra  $z_j$ , a probabilidade se aproxima do grau de ativação da  $j$ -ésima amostra ao  $i$ -ésimo grupo (GATH; GEVA, 1989). Desta forma é possível obter a matriz de covariância  $F_i$  através da equação (3.15).

$$F_i = \frac{\sum_{j=1}^N \varpi_{i,j}(z_j - v_i)(z_j - v_i)^T}{\sum_{j=1}^N \varpi_{i,j}} \quad (3.15)$$

Por estarem os autovalores de  $F_i$  diretamente relacionados à variância do  $i$ -ésimo grupo, o determinante da equação (3.13) apresenta a medida de dispersão  $n$ -dimensional do grupo. Melhores agrupamentos *fuzzy* apresentam grupos mais compactos, ou seja, menor valor de FHV (MACHADO, 2011).

- **Critério APD - Average Partition Density**

Uma outra forma de verificar a qualidade do agrupamento *fuzzy* foi apresentado nos trabalhos de Gath e Geva (1989) e utiliza a densidade média de particionamento para determinar o número ideal de grupos. Este critério é definido através da equação:

$$APD = \frac{1}{K} \sum_{i=1}^K \frac{R_i}{[\det(f_i)]^{\frac{1}{2}}} \quad (3.16)$$

na qual  $F_i$  é a matriz de covariância do  $i$ -ésimo grupo *fuzzy*,  $K$  é o número de grupos e  $R_i$  é a soma dos elementos centrais dado por (GATH; GEVA, 1989):

$$R_i = \sum_j \varpi_{i,j} \forall j \text{ tal que } (z_j - v_i)^T F_i^{-1} (z_j - v_i) < 1 \quad (3.17)$$

Como apresentado na equação (3.16), para os grupos mais compactos o valor do determinante será menor e  $R_i$  será maior. Desta forma, conclui-se que os melhores agrupamentos serão aqueles que apresentarem maior valor de APD (MACHADO, 2011).

- **Cr terio AWCD - Average Within-Cluster Distance**

Com o mesmo objetivo dos outros cr terios j  apresentados, o cr terio AWCD avalia o n mero ideal de grupos *fuzzy*. O cr terio utiliza a medida da dist ncia m dia interna de cada grupo, atrav s da equa o (3.18) (MACHADO, 2007).

$$AWCD = \frac{1}{K} \sum_{i=1}^K \frac{\sum_{j=1}^N \varpi_{i,j}^m \|x_j - v_i\|^2}{\sum_{j=1}^N \varpi_{i,j}^m} \quad (3.18)$$

na qual  $\|\cdot\|$  e  $m$  referem-se a mesma norma e ao par metro *fuzzificador* utilizado no algoritmo de agrupamento, que neste caso utilizou-se o algoritmo de GK com  $m = 2$ . Este cr terio utiliza o valor m dio das dist ncias internas dos grupos, sendo que a dist ncia interna de um determinado grupo   dada pela m dia ponderada das dist ncias entre todas as amostras e o prot tipo (centro) do grupo, com cada dist ncia ponderada pelo grau de pertin ncia da amostra ao grupo  $\varpi_{i,j}$ .

Os valores de AWCD em (3.18) decrescem monotonicamente, assim que o n mero de grupos aumenta, sendo o n mero ideal de grupos, o ponto no gr fico cujo valor de AWCD descreve um joelho mais acentuado como apresentado nos trabalhos de (MACHADO, 2007) e (MACHADO, 2011).

- **Fuzzy Silhouette - FS**

Apresentado no trabalho de Campello e Hruschka (2006), o cr terio de avalia o do n mero de grupos *FS*   uma generaliza o *fuzzy* do cr terio ASWC (*Average Silhouette Width Criterion*) ou *Crisp Silhouette* (KAUFMAN; ROUSSEUW, 2009; EVERITT; LANDAU; LEESE, 2001).

Seja uma amostra  $j \in 1, 2, \dots, n$  pertencente ao grupo  $i \in 1, 2, \dots, K$ . Considerando um agrupamento *crisp* (e.g. *k-means*) a amostra  $j$  est  mais pr xima do prot tipo do grupo  $i$  do que de qualquer outro grupo. Por outro lado, considerando um agrupamento *fuzzy* a amostra  $j$  pertence ao grupo  $i$  com um grau de pertin ncia  $\varpi_{i,j}$  maior que para qualquer outro grupo  $q$ , sendo  $q \neq i$ . Desta forma, a silhueta da amostra  $j$  pode ser escrita como:

$$s_j = \frac{b_{i,j} - a_{i,j}}{\max(a_{i,j}, b_{i,j})} \quad (3.19)$$

na qual  $a_{i,j}$    a m dia da dist ncia da amostra  $j$  para todas as outras amostras do grupo  $i$  e  $b_{i,j}$    o menor valor da dist ncia m dia da amostra  $j$  para todas as outras amostras do grupo  $q$ , sendo  $q \neq i$ , calculado sobre os grupos  $q = 1, 2, \dots, K$ , representando a dissimilaridade da amostra  $j$  para o grupo vizinho mais pr ximo do seu grupo.

Levando-se em consideração o grau de ativação  $\varpi_{i,j}$  da amostra  $j$  dentro do grupo  $i$ , pode-se construir um critério generalizado chamado *fuzzy silhouette* definido por:

$$FS = \frac{\sum_{j=1}^N (\varpi_{i,j} - \varpi_{q,j})^\alpha s_j}{\sum_{j=1}^N (\varpi_{i,j} - \varpi_{q,j})^\alpha} \quad (3.20)$$

sendo  $\varpi_{i,j}$  e  $\varpi_{q,j}$  o primeiro e segundo maior elemento da  $j$ -ésima coluna da matriz de partição *fuzzy*. O coeficiente  $\alpha$ , com  $\alpha > 0$ , dá maior ou menor peso a diferença  $(\varpi_{i,j} - \varpi_{q,j})$ . A influência do parâmetro  $\alpha$  no critério de avaliação de grupos FS foi discutido em Campello e Hruschka (2006). Quando  $\alpha = 0$  na equação (3.20) as informações relativas ao agrupamento *fuzzy* não são desconsideradas, levando o critério *fuzzy-FS* ao mesmo resultado do critério *Crisp Silhouette* original. O valor padrão de  $\alpha$  para o caso *fuzzy* é igual a 1 (CAMPELLO; HRUSCHKA, 2006).

A diferença  $(\varpi_{i,j} - \varpi_{q,j})$  dá maior peso aos dados pertencentes a determinado grupo com maior proximidade do protótipo do grupo do que àqueles em regiões mais acentuada sobreposição entre os grupos. Desta forma, o valor ideal de grupos é representado pelo maior valor de *FS* (MACHADO, 2011).

### 3.3.4 Determinação do Número de Grupos

A utilização isolada de um dado critério de avaliação do número de grupos *fuzzy* nem sempre é a melhor forma de determinar o número ideal de grupos e conseqüentemente o número de modelos locais. Cada critério de avaliação apresenta uma característica diferente sendo mais ou menos preciso dependendo dos dados do problema em questão (CAMPELLO; HRUSCHKA, 2006; BEZDEK; PAL, 1998). Neste contexto, aconselha-se utilizar um conjunto de critérios para auxiliar na tomada de decisão da quantidade de número de grupos pode representar de maneira satisfatória os dados sob estudo.

No trabalho de Machado (2007) foi proposta uma metodologia para determinação do número ideal de grupos que utiliza um esquema de votação. Essa metodologia despreza toda informação quantitativa (valores) dos critérios, usando apenas a informação de qual número de grupos apresentou maior valor na maioria dos critérios, e.g. através da média aritmética ou pelo produto dos resultados (MACHADO, 2007; MACHADO; AMARAL; CAMPELLO, 2007).

O método consiste em primeiramente executar o algoritmo GK, detalhado na Seção 3.3.2, para cada valor de grupos especificado em um dado intervalo de valores especificado pelo usuário. Ao final de cada iteração do algoritmo, as partições *fuzzy* são analisadas

pelos critérios previamente detalhados neste trabalho, e seus respectivos resultados armazenados. Como o critério AWCD original decresce monotonicamente com o número de grupo e seu o resultado de melhor agrupamento é indicado no ponto de maior variação da curva, propõe-se a utilização do critério AWCD modificado apresentado em Machado (2007). Depois de executados os critérios todos os dados são normalizados, e especialmente para o critério FHV descrito na Seção 3.3.3, o valor dos resultados são subtraídos de 1 para que o método também tenha o valor ideal do número de grupos dado pelo maior valor. Após a normalização do resultado dos critérios, calcula-se a média aritmética para um determinado número de grupos e o resultado geral é obtido pelo ponto com maior valor.

A partir do agrupamento realizado considerando o número ideal de grupos, a Seção seguinte apresentará com detalhes a construção do modelo *fuzzy* através da determinação dos antecedentes e consequentes das regras.

## 3.4 Construção dos Modelos *Fuzzy*

Definido o número de grupos, é necessário obter as regras do modelo TS que determinarão os modelos locais, ou seja, obter os antecedentes e consequentes das regras. Neste trabalho serão utilizadas nos antecedentes das regras funções trapezoidais obtidas através das projeções dos grupos *fuzzy* como será detalhado na Seção 3.4.1. Para os consequentes, é necessário determinar os parâmetros da função que estará no consequente dos modelos locais *fuzzy*-TS. Para este trabalho serão utilizadas Base de Funções Ortonormais Generalizadas nos consequentes das regras, e seus parâmetros, serão inicialmente obtidos através do Método do Mínimos Quadrados, como será apresentado na Seção 3.4.2.

### 3.4.1 Determinação dos Antecedentes das Regras

Como detalhado na Seção 3.3.2, as medidas de distância (equação (3.9)) definem os grupos de dados como hiperelipsóides cujo formato é descrito através das matrizes de covariância  $F_i$ , no qual os autovetores de  $F_i$  definem a orientação dos eixos da hiperelipsóides e os autovalores associados aos autovetores definem o comprimento dos eixos. Como apresentado em Babuska e Verbruggen (1997), o agrupamento ao redor da superfície de regressão resulta em hiperelipsóides planas que podem ser vistas localmente como hiperplanos. O autovetor  $\Phi_i = [\phi_{i,1} \dots \phi_{i,d+1}]$ , associado ao menor autovalor da  $i$ -ésima matriz de covariância  $F_i$  é normal ao hiperplano correspondente (MACHADO, 2007).

Os antecedentes  $A_i$  das regras *fuzzy* podem ser obtidos através pela projeção dos valores de ativação dos objetos com relação aos grupos no(s) eixo(s) do espaço de entrada (espaço unidimensional das variáveis das premissas - estados BFO no presente trabalho). Os antecedentes serão compostos pelas funções de pertinência que aproxima a casca convexa das projeções por funções convexas, podendo ser funções trapezoidais, triangulares, gaussianas e etc. Neste trabalho serão utilizadas funções trapezoidais, como ilustrado na Figura 8. Funções trapezoidais possuem um intervalo de pertinência unitária (*core* não *singleton*) que permite a um único modelo local representar sozinho uma determinada região de operação do sistema dinâmico sob estudo.

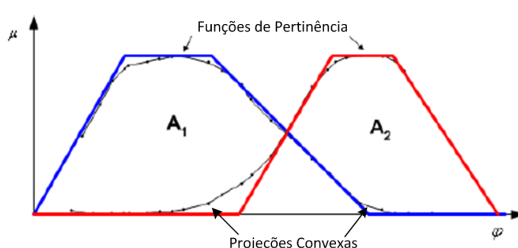


Figura 8: Funções de Pertinência Trapezoidal.

Obtidas as funções de pertinência do modelo *fuzzy*, é necessário obter os consequentes das regras. Para este trabalho os consequentes serão compostos por BFO lineares ou afins, obtidas através do método de Mínimos Quadrados.

### 3.4.2 Determinação dos Consequentes das Regras

O consequente das regras de um modelo *fuzzy* TS pode ser composto por diversos tipos de função, como funções afins, funções lineares, equações diferenciais dentre outras (YAGER; FILEV, 1994). Por apresentar vantagens estruturais como apresentado na Seção 2.1.4, serão utilizadas nesta dissertação as Base de Funções Ortonormais Generalizadas (Seção 2.1.5) para formar o consequente das regras do modelo *fuzzy* TS.

Para se determinar os parâmetros da parte estática das representações matriciais GOBF podem ser utilizadas duas abordagens: utilizando mínimos quadrados a estimação global, no qual os parâmetros são estimados de uma só vez (globalmente); e a estimação local, no qual os parâmetros são estimados separadamente para cada modelo local, ou seja, cada regra tem sua função independente estimada localmente (LJUNG, 1999; MACHADO, 2007; DELGADO; ZUBEN; GOMIDE, 2001).

- **Estimação Global**

A partir do erro de estimação  $\xi(k) = y(k) - \hat{y}(k)$ , sendo  $y(k)$  a saída medida do sistema e  $\hat{y}(k)$  a saída estimada, pode-se usar para otimização dos parâmetros a seguinte função de custo:

$$J = \frac{1}{2N} \sum_{k=1}^N \xi(k)^T \xi(k) \quad (3.21)$$

sendo  $N$  o número de amostras do sinal. Reescrevendo a função de custo (3.21) na forma vetorial:

$$J = \frac{1}{2N} \|Y - \Upsilon C\|^2 \quad (3.22)$$

na qual  $Y^T = [y(1) \ y(2) \ \dots \ y(N)]$  é o vetor de saídas medidas do sistema,  $C^T = [C_1 \ C_2 \ \dots \ C_K]$  é o vetor coluna com os parâmetros dos consequentes de cada regra e a matriz  $\Upsilon$ :

$$\begin{aligned} \Upsilon &= [\Psi_{i,j}]_{i=1,2,\dots,N;j=1,2,\dots,K} \\ \Psi_{k,j} &= \psi(k)\mu_j(k) \end{aligned} \quad (3.23)$$

sendo  $\mu_j(k)$  o valor de ativação para cada regra no instante  $k$ .

O mínimo para a equação (3.23), conforme demonstrado no Apêndice A, é obtido por:

$$\hat{C} = [\Upsilon^T \Upsilon]^{-1} \Upsilon^T Y \quad (3.24)$$

A inversão em (3.24) poderá representar um esforço computacional elevado, quando o número de  $K$  de regras for elevado (MACHADO, 2007).

- **Estimação Local**

A partir do erro de estimação em relação ao  $j$ -ésimo modelo linear local,  $\xi_j(k) = y(k) - C_j^T \cdot \psi(k)$  os parâmetros lineares podem ser determinados de maneira local utilizando-se  $K$  estimadores de mínimos quadrados independentes. A função de custo dos modelos locais, para cada região de validade é definida por:

$$J = \frac{1}{2N} \sum_{j=1}^K \xi_j(k)^T \mu_j \xi_j(k) \quad (3.25)$$

Os termos de ponderação  $\mu_j(k)$ ,  $k = 1, 2, \dots, N$  são introduzidos na otimização do critério (3.13) garantindo que os vetores de entrada (estados da BFO) que estejam mais próximos dos protótipos dos grupos são ponderados com maior peso nos cálculos dos



parâmetros da função no conseqüente da regra correspondente. Os parâmetros lineares da  $j$ -ésima regra, considerando o critério (3.25), são obtidos por (LJUNG, 1999; MACHADO, 2007):

$$\hat{C} = [\Upsilon^T \Lambda_j \Upsilon]^{-1} \Upsilon^T \Lambda_j Y \quad (3.26)$$

sendo:

$$\Upsilon = \begin{bmatrix} \psi(1)^T \\ \psi(2)^T \\ \vdots \\ \psi(2)^N \end{bmatrix}$$

e:

$$\Lambda_j = \text{diag}(\mu_j(1), \mu_j(2), \dots, \mu_j(N)) \quad (3.27)$$

Como apresentado em Murray-Smith (1994), a otimização local dos parâmetros lineares apresenta melhores resultados quando os antecedentes das regras estão sobreparametrizados (número elevado de regiões de validade). Para parâmetros determinados com o critério local, os modelos locais associados a cada região de validade são forçados a ajustar localmente os dados. Desta forma, a estimação pelo critério local dos modelos apresenta características de interpretabilidade, importantes para o projeto de controladores locais multimodelos que serão abordados e utilizados neste trabalho (MACHADO, 2007).

A construção dos modelos *fuzzy* com seus antecedentes e conseqüentes, nem sempre apresentam modelos parcimoniosos, ou seja, modelos mais enxutos que melhor representem todo o universo de discurso do sistema. Neste sentido, é conveniente utilizar métodos para simplificação da base de regras do modelo *fuzzy* como detalhado a seguir.

### 3.4.3 Simplificação dos Modelos *Fuzzy*

Em modelos *fuzzy* gerados a partir de dados de entrada/saída do sistema, podem ocorrer redundância de regras, de funções de pertinência entre si ou com relação ao universo de discurso, ou mesmo com grupos que podem ser fundidos no espaço  $L \times Y$  dos produtos  $\Psi(k) \times y(k)$ . Para estes casos existem métodos para realizar a simplificação da base de regras a partir da fusão/eliminação de regras e/ou funções de pertinência similares (BABUSKA, 1998; MACHADO, 2011).

- **Modelos com grupos compatíveis no espaço  $L \times Y$  (*Compatible cluster merging - CCM*)**

Proposto inicialmente por Krishnapuram e Freg (1992), o algoritmo CCM trabalha com a fusão de grupos com protótipos próximos e com os hiperelipsóides que os definem alinhados no mesmo hiperplano. Originalmente o limitante superior de grupos é escolhido pelo usuário, e o algoritmo é aplicado sucessivamente até que o número de grupos similares convirja para um valor no qual não há mais fusões a serem feitas. Para o trabalho em questão, o número ideal de grupos discutido na Seção 3.3.4 é utilizado como limitante superior do algoritmo CCM.

O algoritmo infere o grau de compatibilidade entre os grupos baseado em parâmetros obtidos através de cálculos com os autovalores e autovetores da matriz que define os grupos. Kaymak e Babuska (1995) propuseram um algoritmo para avaliar se dois grupos são paralelos no mesmo hiperplano ou se o centro de dois grupos são suficientemente próximos para que tais grupos sejam fundidos. A Figura 9 mostra um exemplo de como será a composição de um novo agrupamento após a fusão e como serão as funções de pertinência nas premissas das regras. A Figura 9.a apresenta o agrupamento inicial com os grupos 2 e 3 com uma grande similaridade e a Figura 9.b apresenta o novo agrupamento e as novas funções de pertinência após a aplicação do algoritmo CCM (MACHADO, 2011).

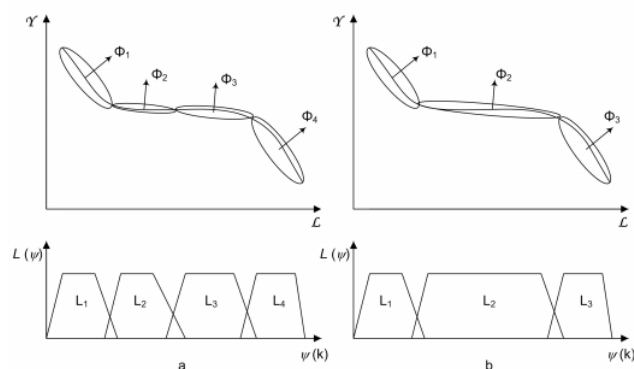


Figura 9: Fusão de grupos similares.

Na construção de uma base de regras eficiente, outras similaridades devem ser analisadas como a similaridade entre a função de pertinência e o universo de discurso e a similaridade entre funções de pertinência obtidas pelas projeções dos grupos.

- **Similaridade entre uma função de pertinência e o universo de discurso**

A similaridade entre uma função de pertinência  $A$  e o universo de discurso se dá quando:

$$A(x) \cong 1, \quad \forall x \in X$$

na qual  $X$  representa o universo de discurso da variável de interesse  $x$ . Quando este tipo de função de pertinência aparece ela deve ser excluída, pois não acrescenta nenhuma informação relevante ao modelo. Como será detalhado a seguir, conjuntos com similaridade com relação ao universo de discurso  $X$  superior a um dado limite  $\rho$  podem ser eliminados já que o a função de pertinência pode cobrir quase a totalidade do universo de discurso.

- **Similaridade entre funções de pertinência**

Considerando que duas funções dificilmente são idênticas, é possível estabelecer um grau de similaridade entre duas funções como apresentado abaixo:

$$s_p = S(A, B) = grau(A = B), s_p \in [0, 1]$$

sendo que o valor de  $s_p$  é próximo de 1 quando as funções são altamente similares e  $s_p$  é próximo de 0 quando as funções são não-similares.

Considerando duas funções de pertinência  $A(x)$  e  $B(x)$  relativas ao subconjunto no universo de discurso  $x$ , a simplificação de regras do modelo *fuzzy* utilizando medidas de similaridade devem satisfazer a 4 critérios.

1. Funções de pertinência cujos suportes não apresentam superposição devem ser consideradas totalmente desiguais,  $s_p = 0$

$$S(A, B) = 0 \Leftrightarrow A(x)B(x) = 0, \quad \forall x \in X.$$

2. Funções de pertinência com superposição devem apresentar valor de similaridade  $s_p > 0$

$$S(A, B) > 0 \Leftrightarrow \exists x \in X, \text{ tal que } A(x)B(x) \neq 0.$$

3. Somente funções de pertinência iguais apresentam valor de similaridade  $s = 1$

$$S(A, B) = 1 \Leftrightarrow A(x) = B(x), \quad \forall x \in X.$$

4. A similaridade entre duas funções de pertinência não deve ser influenciada por escalonamento ou deslocamento do domínio em que elas são definidas

$$S(A, B) = S(A', B'), \quad A'(l + kx) = A(x), \quad B'(l + kx) = B(x), \quad k, l \in \mathfrak{R}, k > 0$$

Na literatura, os métodos de medida de similaridade entre funções de pertinência podem ser classificados em similaridade geométrica *fuzzy* e similaridade baseada na teoria de conjuntos *fuzzy*. As medidas geométricas de similaridade apresentadas no trabalho de Zwick, Carlstein e Budescu (1987), são melhor aplicáveis na medida de similaridade de conjuntos *fuzzy* distintos, enquanto as medidas baseadas na teoria de conjuntos *fuzzy* são melhor aplicáveis na análise entre conjuntos que se sobrepõem, ou seja, mais indicados para analisar quão iguais são as funções de pertinência do modelo. Nesta dissertação, utiliza-se as medidas de similaridade baseadas na teoria de conjuntos *fuzzy* apresentada em detalhes no trabalho de Machado (2011), que satisfaz os quatro critérios descritos anteriormente.

### 3.5 Otimização do modelo *fuzzy* TS-GOBF

Obtido o modelo *fuzzy* TS-GOBF inicial, como discutido na Seção 3.4, e executadas as operações de fusão e eliminação das funções de pertinência também discutido na Seção 3.4.3, pode-se realizar a otimização dos parâmetros do modelo TS-GOBF, minimizando o erro quadrático entre a saída do modelo e a saída medida em cada instante de tempo. Como apresentado em (MACHADO, 2011) o processo de otimização se dá através da minimização de uma função de custo do erro quadrático, empregando-se métodos de otimização não-linear, em especial o método de Levenberg-Marquardt (NELLES, 2001)

$$\min_{\theta} J = \frac{1}{2} \sum_{k=1}^N (\hat{y}(k) - y(k))^2 \quad (3.28)$$

na qual  $\hat{y}(k)$  é a saída estimada do modelo *fuzzy* TS-GOBF;  $y(k)$  é a saída medida do sistema;  $N$  é o número de amostras considerado;  $\theta$  é o vetor dos parâmetros a serem otimizados;

A função de custo é minimizada com relação aos parâmetros  $\theta = [\Upsilon \ C_{f_i} \ L_{i,j}]^T$ , sendo  $\Upsilon$  o vetor com os parâmetros relativos ao(s) polo(s) do modelo,  $C_{f_i}$  a matriz com os coeficientes da expansão dos modelos BFO locais presentes no consequente das regras, definida na Seção 3.4.2, e  $L_{i,j}$  os parâmetros das funções de pertinência presentes nos antecedentes das regras. Para os modelos GOBF com funções internas, utilizados neste trabalho, os

parâmetros a serem otimizados com relação ao(s) polo(s)  $\Upsilon$  serão  $\Upsilon = [\gamma_1 \dots \gamma_{nb}]$  como apresentado na equação (2.27).

A obtenção do modelo final *fuzzy* TS-BFO a partir dos dados de entrada e saída do sistema, proposta no trabalho de Machado (2011) e de especial interesse neste trabalho, segue uma sequência de passos desde a obtenção do modelo inicial através do algoritmo de agrupamento GK até a otimização dos parâmetros do modelo por técnicas de otimização não-linear. O algoritmo a seguir descreve o procedimento a ser utilizado para a obtenção do modelo *fuzzy* TS-GOBF otimizado (MACHADO, 2011).

1. Atribua valores iniciais de parâmetros  $\gamma_l$  (normalmente de maneira aleatório ou baseado em algum conhecimento prévio) e calcule a saída dos filtros ortonormais.
2. Dado um conjunto de amostras de entrada (saída dos filtros ortonormais) e saída do sistema dinâmico, aplica-se o algoritmo de agrupamento de Gustafson-Kessel com o método de determinação do melhor número de grupos a fim de se obter o modelo *fuzzy* TS-GOBF, como apresentado na Seção 3.3.3.
3. Analisar a similaridade entre os grupos obtidos conforme descrito na Seção 3.4.3.
  - **Se** a similaridade for maior que um dado valor limite:  
Realizar o agrupamento dos grupos similares, atualizar o agrupamento *fuzzy* e retornar ao item 3.
  - **Senão** seguir para o procedimento 4.
4. Analisar a similaridade entre os conjuntos *fuzzy* (funções de pertinência) resultantes conforme descrito na Seção 3.4.3.
  - **Se** a similaridade for maior que um dado valor limite:  
Funde-se os conjuntos similares como descrito na Seção 3.4.3, atualiza-se a base de regras *fuzzy* e retornar ao item 4.
  - **Senão** seguir para o procedimento 5.
5. Analisar a similaridade entre os conjuntos *fuzzy* e o universo de discurso.
  - **Se** o valor de similaridade for maior que um valor limite:  
Excluir os conjuntos similares ao universo de discurso, atualizar a base de regras do modelo *fuzzy* e voltar ao item 5.

- **Senão** seguir para o procedimento 6.
6. Realizado o procedimento de análise e eliminação das similaridades do modelo inicialmente obtido conforme descrito nas etapas anteriores, realiza-se o processo de otimização dos parâmetros do modelo *fuzzy* TS-GOBF:
  7. Calcule o gradiente  $\nabla_{\theta} J = [\nabla_{\gamma}^T J \quad \nabla_{C_{fi}}^T J \quad \nabla_{L_{i,j}}^T J]^T$  utilizando as equações detalhadas em (MACHADO, 2011), sendo  $\Upsilon$  o vetor com os parâmetros relativos ao(s) polo(s) do modelo,  $C_{fi}$  a matriz com os coeficientes da expansão em série do modelo BFO locais presentes nos consequentes das regras, definidas na Seção 3.4.2, e  $L_{i,j}$  as funções de pertinência presentes nos antecedentes das regras do modelo *fuzzy* TS-BFO.
  8. Use o valor de  $\nabla_{\theta} J$  no algoritmo de otimização não-linear proposto em Machado (2011) para atualizar o parâmetro  $\theta$ .
  9. Volte ao passo 7 até que um critério de parada seja atingido, como por exemplo, um EQM menor que um dado valor.

Por representar com fidelidade o sistema sob estudo, neste trabalho o modelo final *fuzzy* TS-BFO obtido através do algoritmo apresentado, será utilizado em um sistema de controle preditivo para sistemas não-lineares, utilizando uma abordagem de multimodelos locais como apresentado na Seção 4.

## 3.6 Resumo

Este capítulo apresentou os conceitos fundamentais do projeto de modelos com Base de Funções Ortonormais Generalizadas com funções internas para representação matemática de sistemas dinâmicos não-lineares. Para esta finalidade, utilizou-se modelos *fuzzy* Takagi-Sugeno com GOBF nos consequentes das regras (TS-GOBF). O modelo *fuzzy* TS-GOBF foi obtido utilizando uma abordagem *bottom-up*, no qual primeiramente se determinou um modelo inicial *fuzzy* a partir do agrupamento GK dos dados utilizados nos antecedentes das regras do modelo. O número apropriado de grupos foi obtido através de uma combinação de critérios de avaliação e os coeficientes das BFOs através do método de Mínimos Quadrados.

Obtido o modelo inicial TS-GOBF, foi apresentado um processo de otimização, proposto por Machado (2011), utilizado para simplificar os modelos iniciais obtidos, eliminando e fundindo grupos, conjuntos e regras *fuzzy* redundantes. Além disso, apresentou-se

---

a metodologia para sintonia fina dos parâmetros do modelo obtido através de métodos de otimização não-linear utilizando o cálculo analítico dos gradientes da saída do modelo com relação aos parâmetros de interesse.

## 4 *Controle Preditivo Não-Linear*

O presente capítulo tem como objetivo apresentar uma discussão sobre as técnicas de controle preditivo e em especial a técnica de controle preditivo por multi modelos. Sistemas de controle preditivo baseados em modelo, surgem como umas das mais promissoras estratégias de controle na indústria de processos, pois muitas características em um projeto de controle industrial prático podem ser exploradas neste tipo de abordagem, como a trajetória de referência futura, predição de perturbações e a possibilidade de incorporar restrições. Na Seção 4.1 será apresentado um breve histórico das técnicas de controle preditivo e serão abordados os principais fundamentos referentes ao controlador preditivo linear. Na Seção 4.2 será apresentado o controlador preditivo não-linear que utiliza um modelo *fuzzy* TS-GOBF como preditor e controladores preditivos lineares, que atuam de forma proporcional aos modelos locais nos consequentes das regras do modelo.

### 4.1 Controle Preditivo

Historicamente o conceito de controle preditivo surgiu na década de 70 nos trabalhos de Richalet et al. (1978) e de Cutler e Ramaker (1980) com o DMC *Dynamic Matrix Controller*. Posteriormente, o trabalho de Clarke, Mohtadi et al. (1987) apresentou o GPC *Generalized Predictive Controller* que difundiu na literatura a metodologia MBPC *Model Based Predictive Controller*. Atualmente, os controladores preditivos não estão restritos ao meio acadêmico e são utilizados com sucesso na indústria de processos e no controle de máquinas e equipamentos (LEWIS; EVANS; SANDOZ, 1991; QIN; BADGWELL, 2000; SEKI et al., 2001; ZHAO et al., 2001; QIN; BADGWELL, 1997). A Tabela 1 apresenta algumas das aplicações industriais de MBPC apresentadas no trabalho de Qin e Badgwell (2003).



Tabela 1: Aplicações Comercial de MBPC Linear.

(QIN; BADGWELL, 2003)

Área	Adersa	Aspen Technology	Honeywell Hi-Spec	Invensys	SGS	Total
Refinaria	1200	480	280	25	-	1985
Petroquímica	450	80	280	25	-	550
Química	100	20	03	21	-	144
Papel	18	50	-	-	-	68
Ar e Gás	-	10	-	-	-	10
Utilidade	-	10	-	04	-	14
Metalurgia	08	06	07	16	-	37
Alimentos	-	-	41	10	-	51
Polímeros	17	-	-	-	-	17
Fornos	-	-	42	03	-	45
Aeroespacial	-	-	13	-	-	13
Automotiva	-	-	07	-	-	07
Outras	40	40	1045	26	450	1601
<b>Total</b>	<b>1833</b>	<b>696</b>	<b>1438</b>	<b>125</b>	<b>450</b>	<b>4512</b>

Os controladores preditivos MBPC pertencem a classe de algoritmos que utilizam o modelo do processo para obter a lei de controle, utilizando o modelo do sistema para o cálculo da previsão de saída do processo e com base nos valores dessa previsão, um critério de custo é minimizado em relação às ações futuras do sinal de controle (CAMACHO; BORDONS, 2004). As quatro etapas que caracterizam um controle preditivo são apresentadas a seguir:

1. Modelagem do Processo - O modelo do sistema é utilizado para prever o comportamento do processo dentro de um horizonte de tempo futuro, denominado horizonte de predição;
2. Definição do Critério de Custo - O desempenho do sistema em malha fechada durante o horizonte de predição é especificado através de um critério de custo, definido a partir da saída prevista, do sinal de referência e do esforço de controle;
3. Otimização do Critério de Custo - O critério de custo é minimizado em relação ao conjunto de futuros sinais de controle (em um horizonte de tempo denominado horizonte de controle) a serem aplicados no processo durante o horizonte de previsão;
4. Atuação do Sinal de Controle - Somente o primeiro sinal de controle resultante da otimização do critério de custo é aplicado no sistema a ser controlado. No instante seguinte da amostragem, o processo é repetido;

Diferentes algoritmos de MBPC são encontrados na literatura e as principais diferenças estão relacionadas com o tipo de modelo utilizado no cálculo da previsão de saída, à definição do critério de custo a ser minimizado e a formulação do problema (CLARKE, 1994). A Figura 11 apresenta o modelo genérico de um MBPC, na qual pode-se observar as quatro etapas do controlador descritas anteriormente.

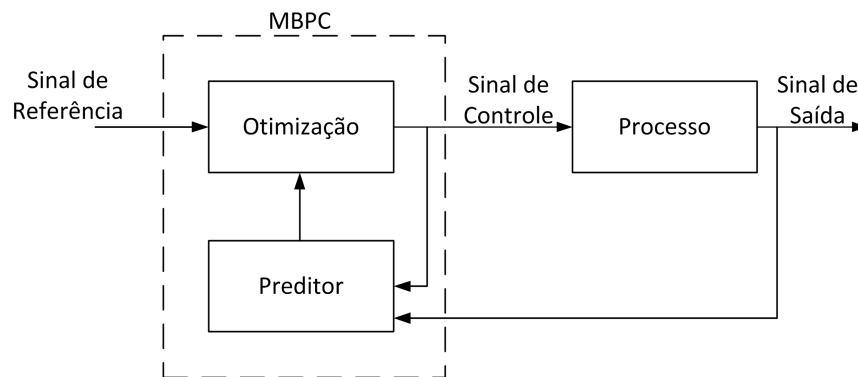


Figura 10: Diagrama de Blocos do Controlador Preditivo.

Nos algoritmos MBPC, a determinação do modelo do sistema é uma etapa fundamental no desempenho do controlador. Em relação aos modelos utilizados, os controladores preditivos podem ser classificados em duas categorias (OLIVEIRA, 1997): Controladores que utilizam modelos estruturados, por exemplo MBPC com modelo CARIMA; e controladores que utilizam modelos não estruturados, como o MBPC com modelo de resposta ao impulso ou MBPC com modelos baseado em BFO, de especial interesse neste de especial interesse neste trabalho. Quando comparados a outras metodologias de controle, os controladores preditivos apresentam as seguintes características (CAMACHO; BORDONS, 2004):

- São indicados para controlar desde processos com dinâmicas simples até processos com grandes atrasos de transporte, processos de fase não-mínima ou processos instáveis;
- Compensa os atrasos de transporte intrinsecamente;
- A extensão de seu algoritmo para o tratamento de restrições é conceitualmente simples;
- Possui uma metodologia aberta baseada em princípios básicos que possibilita futuras extensões;
- Sistemas multivariáveis podem ser controlados por esta abordagem;

As características citadas, fazem do algoritmo de controle preditivo muito atrativo para controle de sistemas dinâmicos industriais. Na Seção a seguir, serão apresentados os passos necessários para o desenvolvimento de um sistema de controle preditivo.

### 4.1.1 Modelagem do Processo e Previsão da Saída $j$ Passos a Frente

Uma das principais etapas para o projeto de um MBPC é a modelagem do sistema a ser controlado, pois o desempenho do sistema em malha fechada está diretamente relacionado com a qualidade do modelo do sistema sob estudo. Neste trabalho, utiliza-se um modelo *fuzzy* com estrutura TS e GOBFs nos consequentes das regras. A escolha de modelos *fuzzy* deve-se ao fato destes modelos serem aproximadores universais em um dado espaço compacto (KOSKO, 1994b; WANG; MENDEL, 1992a) e também porque permitem a composição do modelo não-linear global a partir de múltiplos modelos locais lineares, o que favorece a interpretabilidade e projeto de controladores.

Para o sistema de controle proposto neste trabalho, os modelos de predição utilizados nos consequentes das regras *fuzzy* serão modelos GOBF com estrutura *Ladder* representados sob a forma variacional, como mostrado na equação (2.20), com a adição de uma perturbação  $\xi(k)$ .

$$\begin{aligned}\Delta\Psi(k+1) &= A_f\Delta\Psi(k) + B_f\Delta u(k) \\ \hat{y}(k) &= \hat{y}(k-1) + C_f^T\Delta\Psi(k) + \Delta\xi(k)\end{aligned}\quad (4.1)$$

A previsão de saída  $j$  passos a frente é dada por:

$$\hat{y}(k+j/k) = \mathcal{E} \left\{ \frac{1}{\Delta} C^T \Delta\Psi(k+j) + \frac{1}{\Delta} \Delta\xi(k+j) \right\} \quad (4.2)$$

sendo  $\mathcal{E}$  a média aritmética ponderada pelas probabilidades de ocorrência. Assumindo que o sinal de perturbação  $\xi(k+j)$  tem valor médio igual a zero, a equação (4.2) pode ser reescrita como:

$$\hat{y}(k+j/k) = \mathcal{E} \left\{ \frac{1}{\Delta} C^T \Delta\Psi(k+j) \right\} \quad (4.3)$$

ou ainda:

$$\hat{y}(k+j/k) = \hat{y}(k+j-1/k) + C^T \Delta\Psi(k+j) \quad (4.4)$$

Substituindo  $\Delta\Psi(k+j)$  de (4.1) em (4.4) obtém-se:

$$\hat{y}(k+j/k) = \hat{y}(k+j-1/k) + C^T \{ A\Delta\Psi(k+j-1) + B\Delta u(k+j-1/k) \} \quad (4.5)$$

na qual  $\Delta u(k + j - 1/k)$  é variação do sinal de controle no instante  $k + j - 1$ , calculada a partir da informação disponível no instante  $k$ .

Supondo que  $\Delta u(k + j - 1/k) = 0, \forall j \geq N_u$  e através de sucessivas substituições obtém-se:

$$\hat{y}(k + j/k) = \hat{y}(k + j - 1/k) + C^T A^j \Delta \Psi(k) + C^T \sum_{i=1}^{N_u} A^{j-i} B \Delta u(k + i - 1/k) \quad (4.6)$$

na qual  $N_u$  é um parâmetro de projeto, denominado horizonte de controle,  $A^j = 0$  para  $j < 0$  e  $\Psi(k) = 0$  para  $k \leq 0$ .

Substituindo  $\hat{y}(k + j - 1/k)$  em (4.6), vem:

$$\hat{y}(k + j/k) = \hat{y}(k + j - 2/k) + C^T (A^{j-1} + A^j) \Delta \Psi(k) + C^T \sum_{i=1}^{N_u} (A^{j-i-1} + A^{j-i}) B \Delta u(k + i - 1/k) \quad (4.7)$$

Repetindo-se esta substituição sucessivamente, obtém-se:

$$\hat{y}(k + j/k) = y(k) + C^T (K_j - I) \Delta \Psi(k) + C^T \sum_{i=1}^{N_u} K_{j-i} B \Delta u(k + i - 1/k) \quad (4.8)$$

com a matriz  $I$  sendo a matriz identidade de ordem  $m$ ,  $m$  o número de estados ortonormais do modelo e  $K_j = \sum_{i=0}^j A^i$ , com  $K_j = 0$  para  $j < 0$ .

Na equação (4.8), que representa a previsão de saída do sistema a ser controlado, pode-se observar dois termos distintos: um termo é função das saídas passadas do processo, denominado previsão de saída livre e outro termo é função das informações futuras a respeito de processo, denominado previsão de saída forçada. Desta forma, pode-se reescrever a equação (4.8) como:

$$\hat{y}(k + j/k) = \hat{y}_f(k + j/k) + \hat{y}_l(k + j/k) \quad (4.9)$$

sendo:

$$\hat{y}_l(k + j/k) = y(k) + C^T (K_j - I) \Delta \Psi(k) \quad (4.10)$$

e

$$\hat{y}_f(k + j/k) = C^T \sum_{i=1}^{N_u} K_{j-i} B \Delta u(k + i - 1/k) \quad (4.11)$$

sendo  $\hat{y}_f$  a resposta forçada e  $\hat{y}_l$  a resposta livre.

Para valores de  $j$  entre  $N_1$  e  $N_y$ , que determinam o início e o final do horizonte de previsão, a equação da previsão de saída (4.9) pode ser reescrita na forma matricial descrita abaixo:

$$\hat{y} = G\Delta u + \hat{y}_l \quad (4.12)$$

sendo:

$$\hat{y} = [\hat{y}(k + N_1/k) \quad \hat{y}(k + N_1 + 1/k) \quad \dots \quad \hat{y}(k + N_y/k)] \quad (4.13)$$

$$G = \begin{bmatrix} g_{N_1} & g_{N_1-1} & \dots & g_{N_1-N_u+1} \\ g_{N_1+1} & g_{N_1} & \dots & g_{N_1-N_u+2} \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_y} & g_{N_y-1} & \dots & g_{N_y-N_u+1} \end{bmatrix} \quad (4.14)$$

$$\Delta u = [\Delta u(k/k) \quad \Delta u(k + 1/k) \quad \dots \quad \Delta u(k + N_u - 1/k)] \quad (4.15)$$

$$\hat{y}_l = [\hat{y}_l(k + N_1/k) \quad \hat{y}_l(k + N_1 + 1/k) \quad \dots \quad \hat{y}_l(k + N_y/k)] \quad (4.16)$$

com:

$$g_i = C^T K_{i-1} B \quad (4.17)$$

### 4.1.2 Definição da Lei de Controle Preditivo Linear

Em sistemas de controle preditivo baseados em modelos, a lei de controle é obtida minimizando-se um dado critério de custo, que geralmente é definido em função dos erros da previsão de saída do processo em relação a um sinal de referência e da variação do sinal de controle a ser aplicado ao processo durante o horizonte de predição. No entanto, para alguns casos, o critério de custo é definido de outra forma, como por exemplo no critério com múltiplos modelos de referência (IRWING; FALINOWER; FONTE, 1986).

Para todos os casos em controle preditivo, o critério de custo é definido utilizando normais espaciais e temporais para os sinais envolvidos (OLIVEIRA, 1997). A norma espacial relaciona os sinais em um mesmo horizonte de predição, ou seja, descreve a relação entre a previsão de saída  $\hat{y}(k + j/k)$ , o valor de referência  $\omega(k + j)$  e a ação de controle  $\Delta u(k + j/k)$  em um dado passo de previsão  $j$ . As normas temporais relacionam os sinais considerando todo horizonte de previsão.

Os principais tipos de normas em controle preditivo são:

1. Norma 1:  $\|x\|_1 = \sum_i |x_i|$
2. Norma 2:  $\|x\|_2 = \sqrt{\sum_i x_i^2}$
3. Norma Infinita:  $\|x\|_\infty = \max\{x_i\}$

Como abordado em (CAMPO; MORARI, 1987), o critério de custo a ser minimizado poder ser descrito de diversas maneiras utilizando as normas citadas acima, porém o mais encontrado na literatura é o critério de custo quadrático com norma temporal 1 e espacial 2. Desta forma, a sequência de sinais de controle a serem aplicados no sistema é obtida minimizando o critério de custo em relação às variações futuras do sinal de controle. A lei de controle geral para controladores MBPC é dada por:

$$\begin{aligned} \Delta u &= \arg \min \{F(\hat{y}(k+j/k) - \omega(k+j), \Delta u(k+j-1/k), j = N_1, \dots, N_y)\} \\ &s.a \\ \Delta u(k+j/k) &= 0 \quad \forall j \geq N_u \end{aligned} \quad (4.18)$$

na qual:

$F(\cdot)$  é o critério de custo do controlador, definido de acordo com a escolha das normas espacial e temporal.

$N_1$  é o horizonte inicial de previsão de saída;

$N_y$  é o horizonte final de previsão de saída;

$N_u$  é o horizonte de controle;

$\hat{y}(k+j/k)$  é a previsão de saída  $j$  passos a frente, calculada através da equação (4.8);

$\omega(k)$  é o sinal de referência;

$\Delta u(k+j/k)$  é a variação no sinal de controle no instante  $k+j$ , calculada em  $k$ .

A solução do problema de otimização (4.18) resulta em um vetor  $\Delta u$  composto de  $N_u$  variações do sinal de controle dada por:

$$\Delta u = [\Delta u(k/k) \quad \Delta u(k+1/k) \quad \dots \quad \Delta u(k+N_u-1/k)] \quad (4.19)$$

Somente o primeiro elemento de  $\Delta u$  é aplicado no processo,  $u(k) = \Delta u(k/k) + u(k-1)$ , e no próximo instante de amostragem todo processo é repetido. A definição do critério

de custo do sistema de controle preditivo baseado em modelo BFO proposto por Oliveira (1997) e utilizado neste trabalho, é detalhada na Seção 4.1.3.

### 4.1.3 Controle Preditivo Linear com BFO e Critério de Custo Quadrático

O controlador preditivo baseado em modelo BFO, de interesse neste trabalho, apresenta uma função de custo do tipo quadrática. Este tipo de critério é muito utilizado em algoritmos de controle preditivo pois possibilita uma solução analítica para a lei de controle quando não existem restrições nas variáveis do processo (MACHADO, 2007). O critério de custo quadrático é dado por:

$$J_q(\Delta u) = \sum_{j=N_1}^{N_y} (\hat{y}(k+j/k) - \omega(k+j))^2 + \sum_{j=1}^{N_u} \lambda \Delta u^2(k+j-1) \quad (4.20)$$

sendo  $\lambda (\lambda \geq 0)$  uma parâmetro do controlador que pondera os futuros incrementos do sinal de controle.

Escrevendo a equação (4.20) sob a forma variacional, obtém-se:

$$J_q(\Delta u) = \Delta u^T Q \Delta u + f^T \Delta u + d \quad (4.21)$$

na qual:

$$Q = G^T G + \lambda I \quad (4.22)$$

$$f = 2G^T (\hat{y}_l - \omega) \quad (4.23)$$

$$d = (\hat{y}_l - \omega)^T (\hat{y}_l - \omega) \quad (4.24)$$

e

$$\omega = [\omega(k+N_1) \ \omega(k+N_1+1) \ \dots \ \omega(k+N_y)] \quad (4.25)$$

sendo  $I$  a matriz identidade,  $G$ ,  $\Delta u$  e  $\hat{y}_l$  dados pelas equações (4.14), (4.15) e (4.16), respectivamente.

A lei de controle preditivo consiste em minimizar a função de custo (4.21) em relação a  $\Delta u$ . Esta operação se trata de um problema de programação quadrática (BAZARAA; SHETTY, 1979), no qual um ótimo local para o problema também é um ótimo global. Quando não existem restrições nos sinais de entrada/saída do processo, o problema de

otimização admite uma solução analítica dada pela equação (4.26).

$$\Delta u = -\frac{1}{2}Q^{-1}f \quad (4.26)$$

Quando há restrições nos sinais, a solução é obtida numericamente através de algoritmos de programação quadrática com restrições, como detalhado no trabalho de Oliveira (1997).

## 4.2 Controlador Preditivo Não-Linear

A utilização de controladores preditivos não-lineares vem se apresentando como uma técnica promissora para controle de processos não-lineares, especialmente quando o processo atua em uma faixa de operação ampla ou a não linearidade do processo é significativa o bastante para tornar o desempenho do controlador linear inadequado. Dentre as técnicas encontradas na literatura, pode-se destacar a técnica MMAC<sup>1</sup>, de especial interesse nessa dissertação. Ela é uma estratégia de controle baseada em modelo que incorpora um conjunto de pares modelo/controlador que operam em diferentes regiões de operação do sistema, e apresentam um desempenho melhor quando comparados com sistemas de controle que utilizam um único modelo/controlador atuando em toda a faixa de operação do sistema (SCHOTT; BEQUETTE, 1997).

Este trabalho tem como objetivo a elaboração de um sistema de controle preditivo não-linear através de um controlador preditivo multimodelos utilizando preditores dados por modelos *fuzzy* TS-GOBF com funções internas tipo *ladder*, já detalhado na Seção 3.2. A utilização de modelos *fuzzy* para a modelagem do processo, deve-se ao fato destes modelos serem aproximadores universais de funções em um espaço compacto (WANG; MENDEL, 1992a; KOSKO, 1994b). Os modelos *fuzzy* TS permitem ainda, a composição do modelo global a partir de múltiplos modelos locais, favorecendo a interpretabilidade e projeto de controladores.

Para a implementação do sistema de controle proposto, o primeiro passo é obter o modelo do sistema a ser controlado a partir de dados de entrada/saída do processo, como será descrito com detalhes na Seção 5.2.1. Depois de obtidas as regras *fuzzy* do modelo TS e os modelos locais GOBF, pode-se então implementar o controlador preditivo não-linear de interesse aplicando controladores preditivos lineares (como apresentado em 4.1.3) para cada modelo local definido na etapa de modelagem, como exemplificado na Figura 11.

---

<sup>1</sup>Multiple Model Adaptive Control



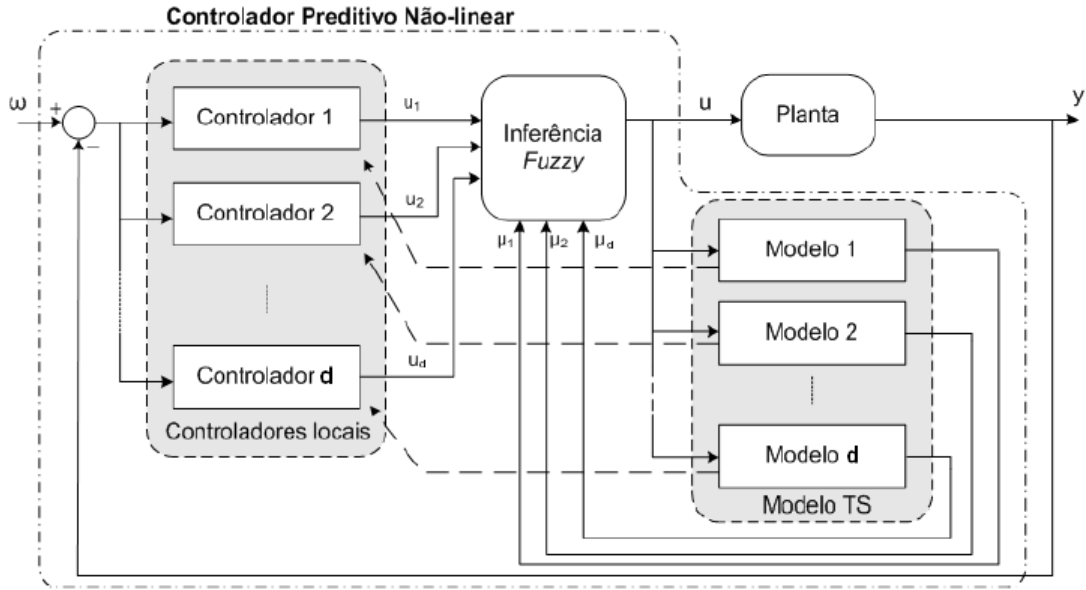


Figura 11: Controlador Preditivo Proposto.

Baseado na resposta do sistema a ser controlado e na previsão de saída dos modelos locais, os controladores lineares atuam sobre os modelos locais de maneira a corrigir a resposta de cada modelo isoladamente. A ação de controle global é obtida pela combinação ponderada pelo modelo *fuzzy* TS, das ações de cada controlador local. O sinal de controle global, além de excitar a planta e os modelos locais, também gera os estados das GOBFs que tem a função de ativar a regra do modelo TS e a saída do predictor. Desta forma, a ação de controle global é obtida através da equação:

$$u(k) = \frac{\sum_{i=1}^K \mu_i(k) u_i(k)}{\sum_{i=1}^K \mu_i(k)} \quad (4.27)$$

na qual  $\mu_i(k)$  é o valor da ativação da  $i$ -ésima regra (modelo local),  $u_i$  a ação de controle determinada pelo  $i$ -ésimo controlador e  $K$  o número de modelos/controladores locais.

### 4.3 Resumo

Neste capítulo apresentou-se um algoritmo de controle preditivo para sistemas não-lineares com multimodelos locais utilizando estruturas *fuzzy* TS-GOBF para inferência das ações de controle. Para controladores MBPC a modelagem do processo a ser controlado é vital, e em sistemas com grandes não-linearidades esse processo pode ser muito complexo. Nesse sentido, o controlador proposto apresenta grande vantagem por trabalhar com a combinação modelos/controladores lineares. Esta operação simplifica a solução

---

do problema global, uma vez que a identificação do sistema e o projeto de controle podem ser decompostas em  $k$  pares modelo/controlador tornando o sistema menos oneroso computacionalmente.

O capítulo a seguir apresenta os procedimentos necessários para obtenção do modelo *fuzzy* TS-GOBF utilizado neste trabalho, bem com os passos para implantação do sistema de controle proposto em um Controlador Lógico Programável Comercial.

## 5 *Sistema de Controle Preditivo Embarcado*

Este capítulo tem como objetivo apresentar os passos necessários para a implementação do Controle Preditivo não-linear proposto nesta dissertação, em um Controlador Lógico Programável comercial. Inicialmente pretende-se apresentar uma breve introdução sobre Controladores Lógicos Programáveis (CLP), com especial interesse para a estrutura interna e para os tempos de execução dos ciclos de máquina. Na Seção 5.2.1 será detalhado o algoritmo para a identificação e otimização do modelo *fuzzy* TS-GOBF, obtido através dos sinais de entrada/saída do sistema dinâmico utilizando o CLP. E por fim na Seção 5.2.2, será detalhado o algoritmo para obtenção da ação de controle dos controladores locais e a ação de controle global.

### 5.1 Introdução a Controladores Lógicos Programáveis

Historicamente, os primeiros sistemas de controle lógico surgiram durante a revolução industrial no final de século XIX com o objetivo de automatizar algumas tarefas críticas e repetitivas nas linhas de montagem (FRANCHI; CAMARGO, 2009). No início, o controle dos processos era realizado com complexos dispositivos mecânicos que eram projetados para funções específicas sem qualquer flexibilidade. Com o surgimento dos dispositivos eletromecânicos, tais sistemas mecânicos foram aos poucos substituídos por sistemas de controle que utilizavam relés e contadores. Desta forma, possibilitou-se o desenvolvimento de funções de controle mais eficientes e com uma flexibilidade maior que a obtida com sistemas mecânicos.

Com a evolução dos processos industriais, os sistemas de controle que utilizavam dispositivos eletromecânicos se tornaram ineficientes, pois em uma mesma linha de produção vários produtos começaram a ser produzidos e a mudança da lógica de controle em

painéis de relés demandavam altos gastos de tempo e dinheiro. A partir da necessidade existente na indústria em geral, principalmente na indústria automobilística, em 1968 uma companhia americana chamada Bedford Associated criou, a pedido da General Motors, um dispositivo de computação chamado *MODICON - Modular Digital Controller* que era facilmente programável e atendia as exigências industriais (CAPELLI, 2008), sendo considerado o primeiro Controlador Lógico Programável.

Os primeiros CLPs que surgiram no mercado eram equipamentos grandes, caros e executavam somente o controle de processos discretos, ou seja, não possuíam entradas e saídas analógicas. No início da década de 80, com a evolução da eletrônica e da computação, surgiram os primeiros CLPs que realizavam leitura de grandezas analógicas e consequentemente, estavam aptos a executar alguns algoritmos de controle clássico, como por exemplo o controlador PID (MORAES; CASTRUCCI, 2007). Atualmente os CLPs são utilizados em inúmeras aplicações industriais e até mesmo em aplicações residenciais e comerciais.

### 5.1.1 Arquitetura dos CLPs e Princípio de Funcionamento

De acordo com o *IEC - International Electrotechnical Commission*, o Controlador Lógico Programável é definido como:

*"Sistema eletrônico operando digitalmente, projetado para uso em um ambiente industrial, que usa memória programável para a armazenagem interna de instruções orientadas para o usuário para implementar funções específicas, tais como lógicas, sequencial, temporização, contagem e aritmética, para controlar, através de entradas e saídas analógicas e digitais, vários tipos de máquinas e processos. O controlador programável e seus periféricos são facilmente integráveis em um sistema de controle industrial e facilmente usados em suas funções previstas."*

Como definido pelo IEC, o CLP é um dispositivo microprocessado para ambientes industriais que pode ser programado para executar funções para controle de dispositivos, máquinas e processos. Este controle é realizado através de funções específicas como lógica de controle digital, controle de tempo, operações matemáticas, controle em malha fechada, transmissão de dados, dentre outras. Para executar tais funções de controle e monitoração, o CLP é composto internamente por cinco blocos básicos, como demonstrado na Figura 12.

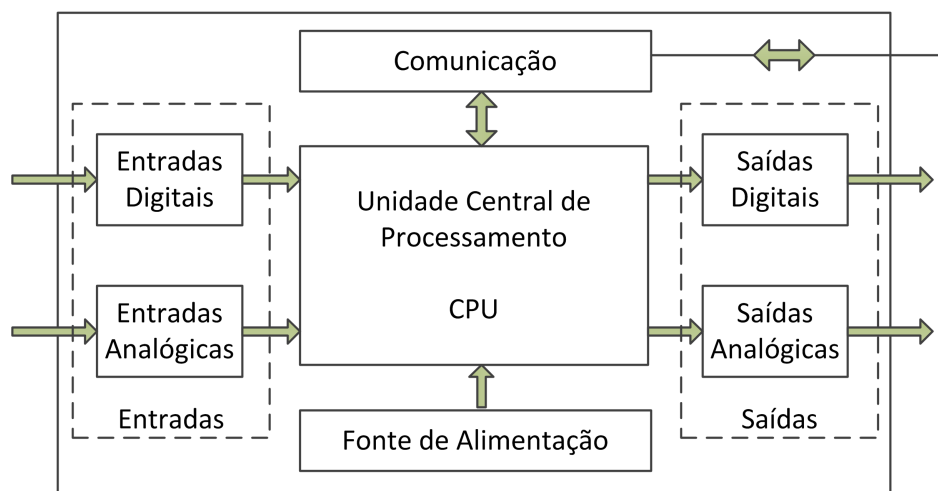


Figura 12: Diagrama em Blocos do CLP.  
(MORAES; CASTRUCCI, 2007)

A fonte de alimentação, é responsável pelo fornecimento de todas as tensões utilizadas nos componentes eletrônicos presentes na CPU (*Central Processing Unit*) e nos módulos de entrada e saída. Os módulos de entrada e saída são as interfaces entre os elementos de campo (sensores e atuadores) e a unidade central de processamento conhecida como CPU. Já a CPU, é responsável pela execução do programa embarcado no processador e pelo gerenciamento das tarefas do CLP. A CPU recebe os sinais das entradas digitais e analógicas dos elementos de campo (em casos especiais informações via protocolo de rede) e executa as funções lógicas e aritméticas programadas. Ao final de cada execução do programa, a CPU atualiza as saídas digitais e analógicas que farão a interface entre a CPU e os elementos de acionamento como motor, lâmpada, contator, relés e etc.

Quando um CLP está em operação, ou seja, está em *Modo Run*, o sistema operacional embarcado no microprocessador executa ciclos de máquina contínuos chamado de *scan*. Cada ciclo de máquina do CLP é composto por três processos distintos: leitura das entradas, execução dos programas e atualização das saídas (FRANCHI; CAMARGO, 2009) como mostrado na Figura 13.

A execução de um ciclo de máquina completo ocorre em um tempo pré estabelecido denominado Tempo de Ciclo ou *Cycle Time*, e sua configuração está relacionada com o número de entradas e saídas conectadas ao CLP, e com a complexidade dos programas a serem executados. A Figura 14 apresenta o diagrama de execução dos ciclos de máquina do CLP.

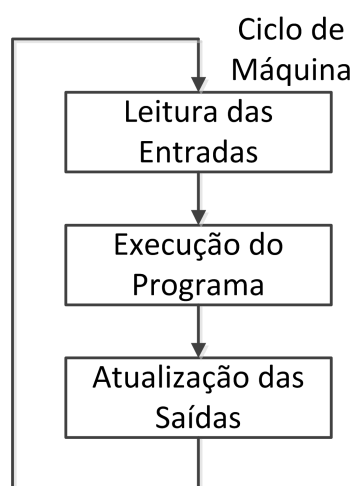
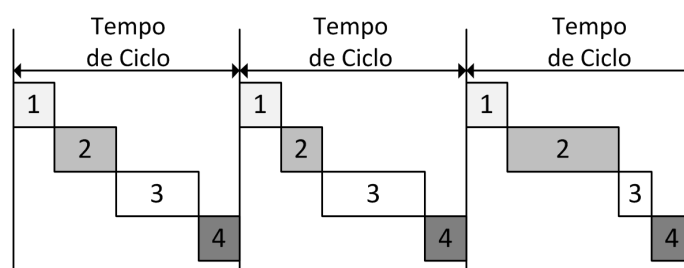
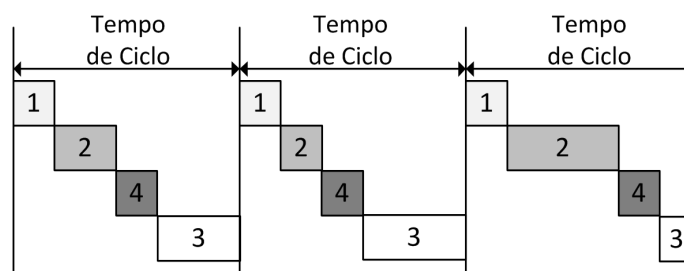
Figura 13: *Scan* do CLP.

Figura 14: Tempo de Ciclo do CLP.

A etapa 1 na Figura 14 representa a atualização das entradas do CLP. Já a etapa 2 corresponde a execução do(s) programa(s) armazenados no CLP e diferentemente da etapa 1, nem sempre a execução dos programas possui mesma duração, pois em determinados momentos do processo a execução de uma parte específica do programa pode levar mais tempo para ser finalizada. A etapa 3 representa o tempo de espera ou *Idle Time* utilizado para execução de outras tarefas da CPU, como por exemplo a atualização de telas da IHM (Interface Homem Máquina). Nos ciclos de máquina que o tempo de execução dos programas (etapa 2) ocorre em um tempo maior, o tempo de espera é praticamente zero e neste caso as tarefas secundárias, como atualização de telas, ocorrerão no próximo ciclo que houver *Idle Time*. Por fim, a etapa 4 representa a atualização das saídas do controlador.

Em alguns casos específicos, a atualização das saídas deve ocorrer imediatamente após a execução do programa no CLP. Para estes casos, utiliza-se uma configuração especial de *Cycle Time* denominada *Fast Reaction*, em que a atualização das saídas ocorre após a execução do programa e o tempo de espera ocorre após a atualização (etapa 4) como demonstrado na Figura 15.

Figura 15: Tempo de Ciclo do CLP *Fast Reaction*.

Para controle de sistemas dinâmicos, especial interesse nessa dissertação, utiliza-se a configuração de *Cycle Time* padrão (Figura 14) no qual o tempo de amostragem das entradas e atualização das saídas devem ocorrer sempre em períodos iguais. A seguir, serão apresentadas de forma sucinta as linguagens de programação utilizadas para desenvolvimento dos programas utilizados nos CLPs.

### 5.1.2 Linguagens de Programação

Desenvolvido com o intuito de automatizar processos industriais digitais, os primeiros CLPs eram programados utilizando uma linguagem de programação não padronizada específica para automação industrial denominada *Ladder*. Em função da falta de padronização a capacitação dos profissionais em programação de sistemas de automação gerava gastos de tempo e dinheiro, pois o profissional deveria estar apto a programar utilizando diferentes variações da linguagem *Ladder* em diferentes fabricantes de CLPs. Nesse contexto, foi criado um grupo de trabalho no IEC (*International Electrotechnical Commission*) para estabelecer normas para desenvolvimento, fabricação e utilização dos CLPs (FRANCHI; CAMARGO, 2009).

Publicada no início dos anos 90, a norma IEC 1131 foi a primeira norma internacional relacionada a Controladores Lógicos Programáveis e abordava o ciclo de vida e fabricação desses equipamentos. Assim que recebeu a primeira revisão, a norma passou a se chamar IEC 61131 e teve várias publicações ao longo dos anos como detalhado abaixo (CAPELLI, 2008):

- **Parte 1:** terminologia, conceitos e informações gerais (1992);
- **Parte 2:** teste de construção elétrica e mecânica (1992);
- **Parte 3:** linguagens de programação (primeira versão), estrutura de software e execução do programa (1993);

- **Parte 4:** seleção, instalação e manutenção de CLPs (1995);
- **Parte 5:** comunicação com outros dispositivos baseados em MAP (*Manufacturing Automation Protocol*) (1995);
- **Parte 6:** comunicação via *fieldbus* (1995);
- **Parte 7:** programação *Fuzzy* (1997);
- **Parte 8:** linguagens de programação (1999);

Visando atender o maior número de usuários e as mais diferentes aplicações, a publicação IEC 61131-3 definiu a sintaxe e a semântica de cinco linguagens de programação (FRANCHI; CAMARGO, 2009), como apresentado abaixo:

- Linguagem Ladder *LD - Ladder Diagram*: Linguagem gráfica baseada na lógica de relés e contatos elétricos, utilizada para a realização de circuitos de controle e acionamento. Está presente na grande maioria dos modelos de CLPs atuais e é bastante utilizada para programação de lógicas de intertravamento digital;
- Lista de Instruções *IL - Instruction List*: Inspirada na linguagem *assembly* e puramente sequencial, utiliza instruções que possuem um operador para desenvolver lógicas entre variáveis. Por ser extremamente limitada, não é utilizada atualmente em programação de máquinas e equipamentos;
- Sequenciamento Gráfico de Funções *SFC - Sequential Function Chart*: Linguagem de programação gráfica que permite o desenvolvimento de ações sequenciais, paralelas e alternativas existentes em aplicações industriais. A linguagem SFC vem sendo utilizada com sucesso em aplicações sequenciais de grande porte na indústria de processos;
- Texto Estruturado *ST - Structured Text*: Linguagem textual de alto nível e muito poderosa, inspirada na linguagem Pascal. Por englobar todos os elementos essenciais de uma linguagem de programação moderna, como as instruções condicionais (IF-THEN-ELSE e CASE OF) e as instruções de repetição como (FOR; WHILE e REPEAT), essa linguagem é muito utilizada para controle de processos complexos que necessitam de uma programação estruturada.
- Diagrama de Blocos de Funções *FBD - Function Block Diagram*: Linguagem de programação gráfica muito utilizada na Europa, no qual os elementos lógicos são expressos por blocos interligados, semelhante aos utilizados em eletrônica digital. Essa



linguagem permite um desenvolvimento modular do software, característica muito importante para controle de processos contínuos presentes nas indústrias químicas, petrolíferas, alimentícias, dentre outras. Usualmente os blocos são constituídos utilizando a linguagem de Texto Estruturado;

Com o aumento da complexidade dos processos industriais a serem controlados, o desenvolvimento de novos algoritmos se faz necessário em processos reais. Com a publicação na norma IEC 61131-3 e com a necessidade de mercado, no início dos anos 2000 surgiram os primeiros CLPs que permitiam ao operador utilizar linguagens de programação padronizadas e já bastante difundidas no ramo de computação como o Texto Estruturado. Essa nova funcionalidade, permitiu que algoritmos de controle que antes eram embarcados somente em sistemas computacionais de alta performance e alto custo (e.g. *CompactRIO*) fossem implementados em CLPs comerciais.

Neste contexto, neste trabalho propõe-se a implementação do algoritmo de controle preditivo não-linear baseado em modelos *fuzzy* TS-GOBF, descrito com detalhes na Seção 4.2, em um controlador lógico programável comercial para o controle de processos. Na Seção a seguir serão descritas as metodologias utilizadas para a implementação do controlador proposto em um CLP.

## 5.2 Implementação do Controle Preditivo Não-Linear em um Controlador Lógico Programável

A implementação do Controlador Preditivo Não-Linear em um CLP comercial, pode ser dividida em duas etapas distintas: identificação e cálculo das ações de controle. Na etapa de identificação (Seção 5.2.1) o modelo *fuzzy* TS-GOBF que melhor representa o sistema em questão é obtido. A partir deste modelo, os parâmetros do controlador preditivo são calculados (OLIVEIRA, 1997) e embarcados no CLP. Por fim, as ações de controle são calculadas ciclicamente no CLP utilizando o algoritmo descrito na Seção 5.2.2.

### 5.2.1 Identificação

Como descrito anteriormente, o primeiro passo para a implementação do controlador em questão consiste em obter o modelo do processo a ser controlado. A obtenção do modelo *Fuzzy* TS-GOBF se dá de maneira *offline* a partir de dados de entrada e saída do

processo, como detalhado no fluxograma apresentado na Figura 16.

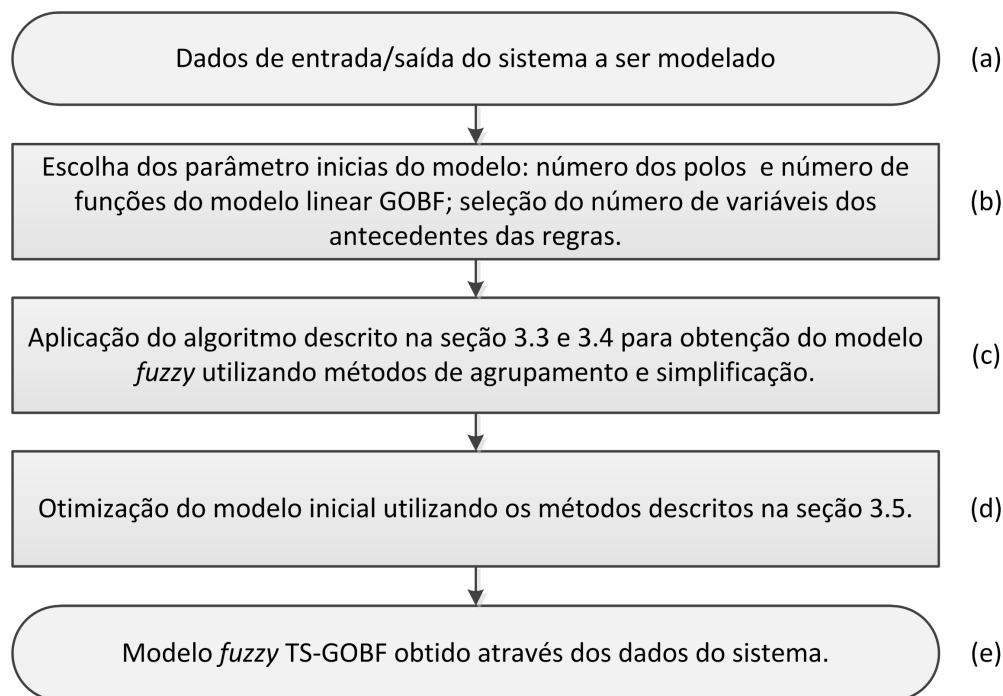


Figura 16: Fluxograma dos passos para obtenção do modelo TS-GOBF.

A primeira etapa do algoritmo para determinação do modelo da planta consiste na obtenção dos dados do processo através de um sistema de aquisição implementado em um CLP. Um conjunto de degraus, previamente configurados pelo operador, é aplicado na planta e os sinais de entrada e saída são armazenados em dois vetores com  $n$  posições cada. O tempo de ciclo do CLP na etapa de aquisição, ou seja, o tempo de amostragem dos sinais de entrada e saída da planta, deve ser configurado de acordo com a dinâmica do sistema e deve assim obedecer o critério de Nyquist (PHILLIPS; NAGLE, 1995). Os dois vetores, gerados com  $n$  posições a partir dos dados da planta, são exportados para um software matemático, e então, os algoritmos para obtenção do modelo *fuzzy* TS-GOBF (etapas (b), (c) e (d) no fluxograma apresentado na figura 16) serão executados. Para esta dissertação em especial, utilizou-se o software MATLAB®.

A segunda etapa do algoritmo consiste na escolha dos parâmetros do modelo inicial. Nesta etapa são especificados o número  $n_b$  de polos da GOBF, o número de funções  $n_f$  presentes na base e o número de variáveis nos antecedentes das regras *fuzzy*,  $\varsigma \leq n_f$ . Quando não se tem conhecimento a priori do sistema a ser modelado, é usual repetir o algoritmo de identificação, demonstrado na Figura 16, para diferentes parâmetros do modelo inicial, como por exemplo, variando-se o número de polos e/ou o número de funções presentes na base.

Depois de escolhidos os parâmetros do modelo inicial e de posse do conjunto de amostras de entrada e saída do sistema, aplica-se o algoritmo de Gustafsson-Kessel com o método de determinação do melhor número de grupos para obtenção do modelo *fuzzy* TS-GOBF inicial, como apresentado na Seção 3.3. Obtido o modelo *fuzzy* TS-GOBF inicial, utilizam-se métodos para análise da similaridade entre os grupos utilizando a metodologia proposta Machado (2011) e detalhada na Seção 3.4.3.

Tendo-se concluído esta etapa com a possível eliminação das similaridades do modelo inicial, realiza-se o processo de otimização não-linear do modelo para o ajuste dos polos, dos coeficientes da expansão das funções ortonormais presentes nos consequentes das regras e dos parâmetros das funções de pertinência nos antecedentes das regras. Neste trabalho, utiliza-se o algoritmo de otimização não-linear proposto em (MACHADO, 2011) e descritos na Seção 3.5.

O modelo final *fuzzy* TS-GOBF do sistema a ser controlado é então embarcado no CLP juntamente com as matrizes e parâmetros do controlador. Além disso, é necessário também se embarcar no CLP o algoritmo de controle responsável pelo cálculo das ações de controle do sistema. A seguir, serão descritos os passos utilizados para o cálculo das ações de saída do controlador preditivo não-linear em questão.

### 5.2.2 Cálculo das Ações de Controle

Com base nas informações do modelo obtido através do algoritmo apresentado na Figura 16 (detalhado nas Seções 3.3, 3.4, 3.5) e nos parâmetros do controlador, o cálculo das ações de controle são realizadas pelo CLP seguindo o fluxograma apresentado na Figura 17 (de acordo com o apresentado nas Seções 4.1.1, 4.1.2 e 4.1.3).

O algoritmo de cálculo para as ações de saída do controlador preditivo proposto, ocorrem ciclicamente no CLP em um *Cycle Time* previamente definido pelo usuário. A seguir serão detalhadas as etapas do algoritmo apresentado.

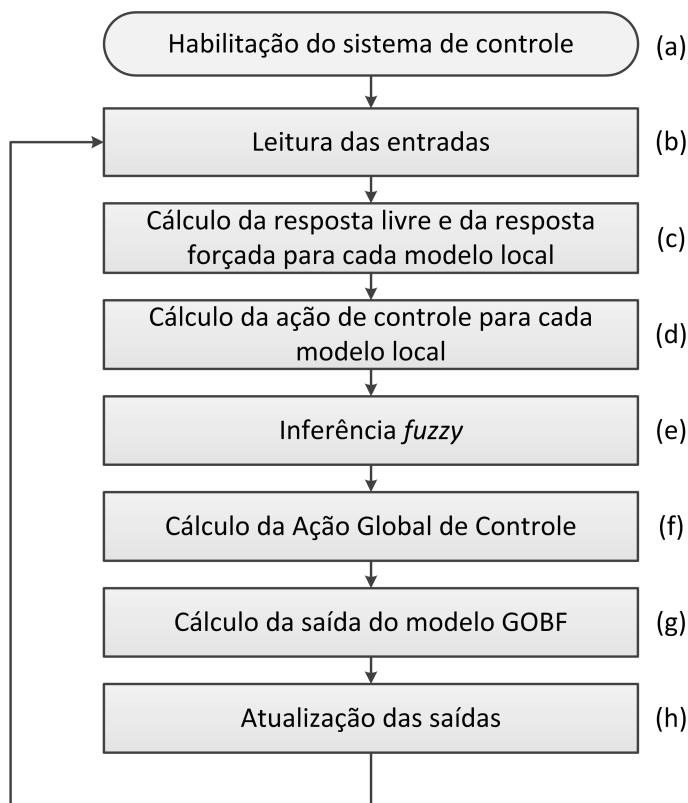


Figura 17: Fluxograma dos passos para execução do controle TS-GOBF.

- **Passo (b) - Leitura das Entradas do CLP**

Como detalhado na Seção 5.1.1, a leitura das entradas digitais/analógicas ocorre no início de cada ciclo do CLP, ou seja, a cada ciclo de máquina a variável associada ao(s) módulo(s) de entradas digitais e analógicas são atualizadas. Para este modelo de controlador, sempre que o valor da variável a ser controlada (entrada analógica) for atualizado, ela será convertida para uma grandeza de interesse, por exemplo, para uma planta de nível a variável da entrada analógica (variável tipo *INTEGER* de 16 bits sinalizado com valor de -32768 a 32767) é convertida em centímetros de coluna de líquido e utilizada para o cálculo da resposta livre dos modelos locais como será demonstrado a seguir.

- **Passo (c) - Cálculo da resposta livre e da resposta forçada para cada modelo local**

A previsão de saída do processo a ser controlado, definida na equação (4.9), é composta por dois termos distintos, a saída livre  $\hat{y}_l$  e a saída forçada  $\hat{y}_f$ . Para o controle preditivo com GOBF, a previsão de saída livre e a previsão de saída forçada dos modelos locais, são calculadas seguindo os digramas em blocos apresentado nas Figuras 18 e 19 respectivamente (como apresentado nas Seção 4.1.2 e 4.1.3).

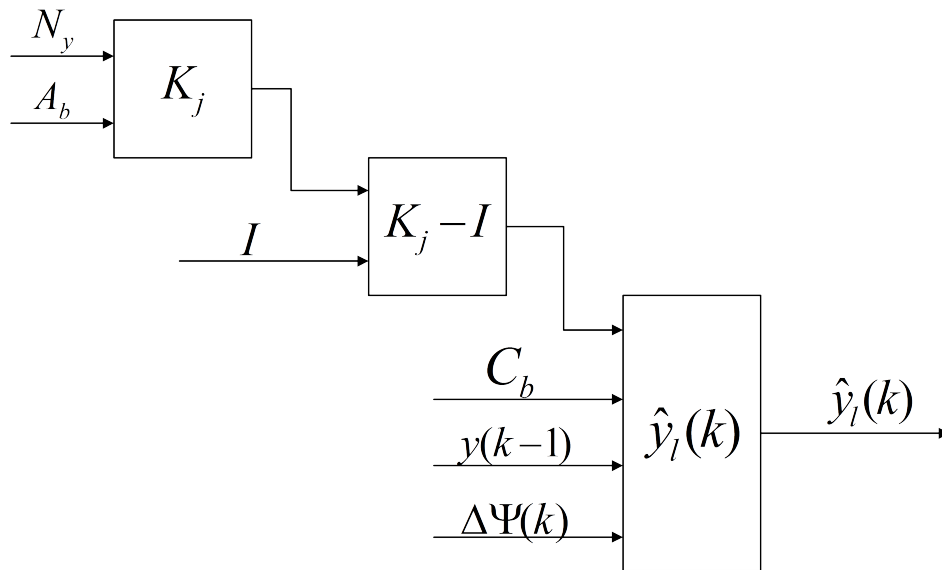


Figura 18: Cálculo da Resposta Livre do Modelo Local.

na qual:

- $N_y$  é o horizonte final da previsão de saída;
- $A_b$  é obtida diretamente a partir dos parâmetros do modelo, como apresentado na equação (2.27);
- $I$  é a matriz identidade de ordem  $m \times m$ , sendo  $m$  o número de estados ortonormais do modelo;
- $C_b$  é o vetor dos coeficientes da expansão das funções ortonormais, como apresentado na equação (2.27);
- $y(k-1)$  é a saída da planta controlada no instante  $k-1$ ;
- $\Delta\Psi(k)$  é a variação da saída das funções ortonormais;
- $\hat{y}_l$  é a previsão de saída livre do modelo local;

O cálculo do vetor  $f$  (conforme equação (4.23)) com a previsão de saída do sistema  $j$  passos a frente é dada por:

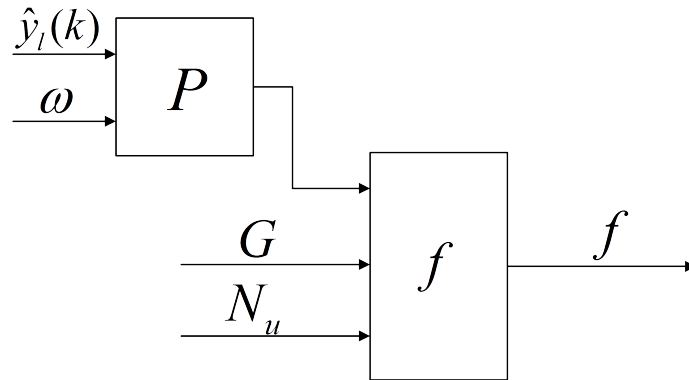


Figura 19: Cálculo do vetor  $f$  do modelo local.

na qual:

- $\hat{y}_l(k)$  é a saída livre do modelo local calculado a partir do diagrama apresentado na Figura 18;
- $\omega$  é o *Set Point* do controlador;
- $N_u$  é o horizonte de controle;
- $G$  é uma matriz de dimensão  $N_y \times N_u$  formada a partir dos elementos  $g_i$  calculados pela equação (4.17) de maneira *offline*;

A metodologia para o cálculo das saídas livre e forçada do sistema de controle são válidas para cada modelo local isolado. Os cálculos devem ser repetidos  $K$  vezes de acordo o número de modelos locais utilizados na representação do sistema dinâmicos em questão.

**• Passo (d) - Cálculo da ação de controle para cada modelo local**

Calculado o vetor com dimensão  $N_y$  da resposta livre do sistema e o vetor  $f$  com dimensão  $N_u$ , pode-se então, calcular a ação de controle para cada modelo local, seguindo o diagrama apresentado na Figura 20.

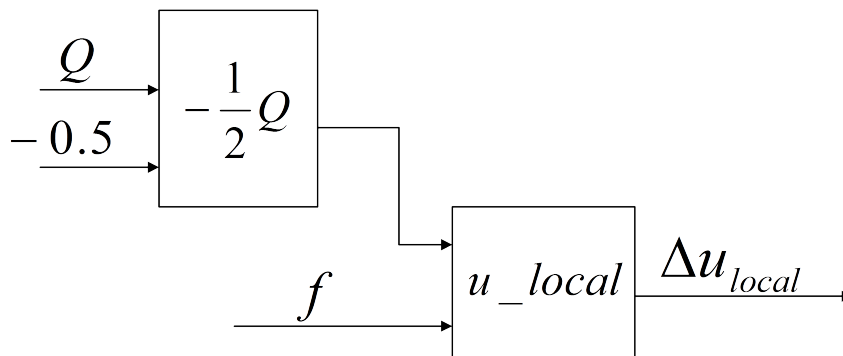


Figura 20: Cálculo da ação de controle local.

na qual:

- $Q$  é uma matriz de dimensão  $N_u \times N_u$  calculada previamente de maneira *offline* através da equação (4.22), sendo  $\lambda$  um parâmetro do controlador que pondera as futuras ações de controle.

- $f$  é vetor da resposta do sistema  $j$  passos a frente, calculado a partir do diagrama da Figura 19;

O vetor de saída  $u_{local}$  com dimensão  $N_u$ , apresentado no diagrama da Figura 20, deve ser repetido  $K$  vezes de acordo com o número de modelos locais.

• **Passo (e) - Inferência Fuzzy**

A ação de controle global do sistema de controle preditivo utilizado nesta dissertação é obtida pela combinação ponderada das ações de cada controlador local, sendo que essa ponderação é dada pelo grau de ativação  $\mu_i$  da  $K$ -ésima regra do modelo fuzzy TS. Obtida a base de regras do modelo *fuzzy* TS (detalhado na Seção 3.3) os graus de ativação são obtidos através da interpolação clássica TS proposta no trabalho de Takagi-Sugeno (1985), como apresentado na Figura 21.

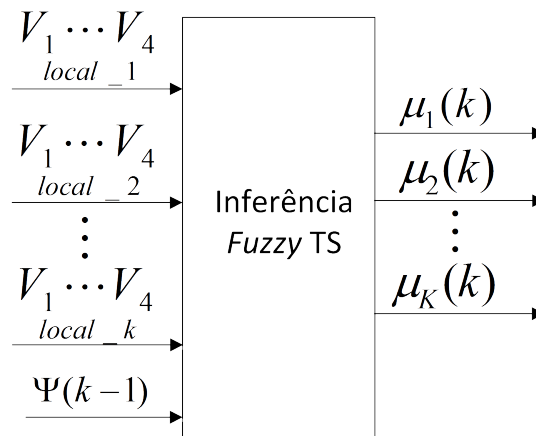


Figura 21: Cálculo da ação de controle global.

na qual:

- $V_1 \dots V_4$  é a função de pertinência trapezoidal de cada modelo local;

- $\mu_K(k)$  é o grau de ativação de cada regra (modelo local) como detalhado na Seção 3.4.1;

- $\Psi(k-1)$  é vetor de dimensão  $d$  das saídas dos filtros ortonormais no instante  $(k-1)$ ;

• **Passo (f) - Cálculo da Ação de Controle Global**

Obtidos os graus de ativação  $\mu_k$  das  $K$  regras do modelo *fuzzy* TS, a ação de controle global pode ser obtida através da equação (4.27), como apresentado na Figura 22.

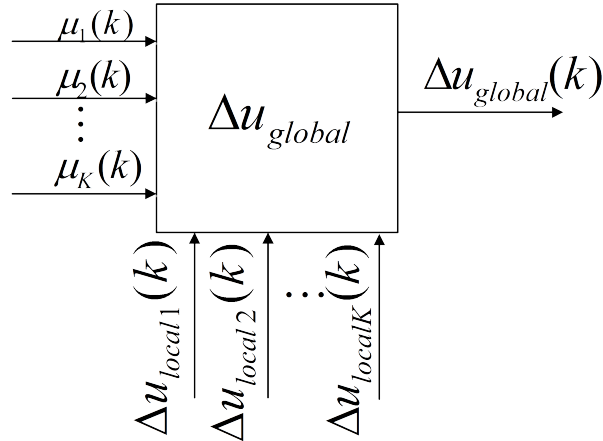


Figura 22: Cálculo da ação de controle global.

na qual:

- $\mu_1 \dots \mu_k$  é o grau de ativação da  $k$ -ésima regra do modelo *fuzzy*.
- $\delta u_{local1} \dots \delta u_{localk}$  é a ação de controle para cada modelo local obtidos através da equação (4.26), como apresentado no diagrama da Figura 20.

O cálculo da ação de controle global, apresentada na Figura 22, é executado em cada ciclo de máquina do CLP e aplicado na planta controlada e nos modelos locais, que fornecerão informações para o cálculo das ações dos controladores locais no próximo ciclo de máquina, ou seja, no instante  $(k + 1)$ .

• **Passo (g) - Cálculo da saída do modelo GOBF**

A ação de controle global calculada anteriormente, é utilizada para calcular a saída das funções GOBF (equação (2.27)) no instante atual  $k$ . A saída dos filtros ortonormais serão utilizadas para o cálculo da resposta livre do sistema dinâmico  $y_l$  e como variável nas premissas do modelo *fuzzy* TS-GOBF. A Figura 23 apresenta um diagrama para o cálculo da saída das funções ortonormais.



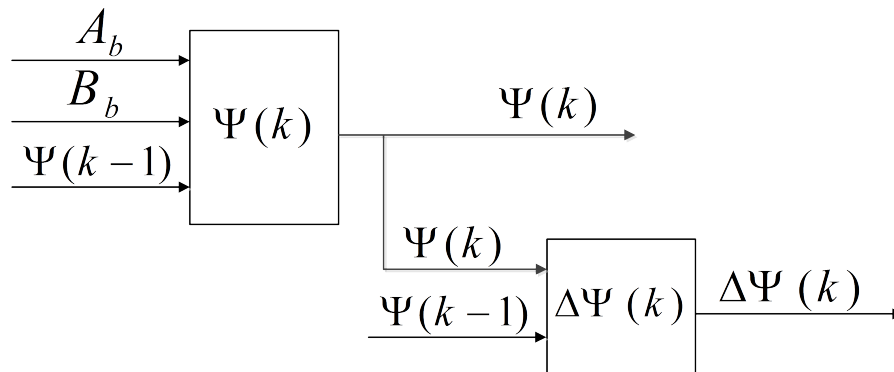


Figura 23: Cálculo da saída das funções GOBF.

na qual:

- $A_b$  e  $B_b$  são obtidas diretamente a partir dos parâmetros do modelo, como apresentado na equação (2.27);
- $\Psi(k-1)$  é vetor de dimensão  $d$  das saídas dos filtros ortonormais no instante  $(k-1)$ ;

Quando se utiliza simulação, as saídas  $\hat{y}_k$  dos modelos locais podem ser obtidas multiplicando a saída dos filtros ortonormais  $\Psi(k)$  pelo vetor  $C_{local}$  que contém os coeficientes da expansão da série de funções ortonormais, como apresentado na Figura 24.

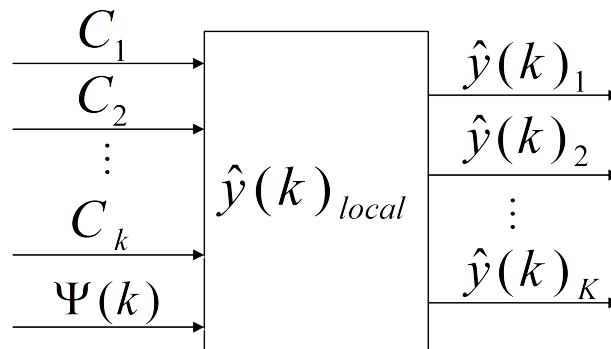


Figura 24: Saída dos modelos locais.

A saída global do modelo pode ser obtida pela ponderação das saídas de cada modelo local, como detalhado na Figura 25.

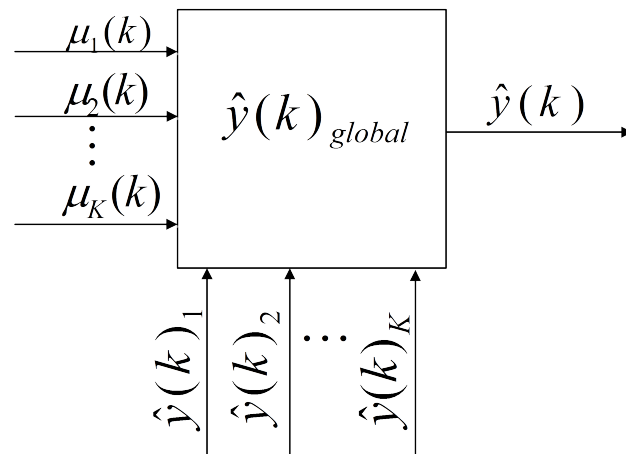


Figura 25: Saída global do sistema.

Para o trabalho em questão, as saídas locais não precisam ser calculadas, pois a saída global é medida diretamente do sistema dinâmico através da entrada analógica do CLP.

- **Passo (h) - Atualização das Saídas do CLP (h)**

Como detalhado na Seção 5.1.1, a atualização das saídas do CLP podem ocorrer de duas formas. Na configuração mais utilizada no meio industrial, denominada *without jitter*, a atualização das saídas ocorre sempre no mesmo instante do ciclo do CLP (Figura 14). Outra configuração utilizada em alguns casos especiais na indústria é denominada *fast reaction*, e a atualização ocorre sempre após a execução do programa no CLP. Para o algoritmo de controle utilizado nessa dissertação, utiliza-se a configuração padrão *without jitter*, pois para o bom funcionamento do sistema de controle é necessário garantir que a atualização da saída analógica do CLP, que está associada a variável da ação global, ocorra sempre no mesmo instante.

## 5.3 Resumo

O presente capítulo teve como objetivo apresentar a metodologia para implementação do controlador preditivo não-linear de multimodelos locais em um controlador lógico programável comercial. Inicialmente, apresentou-se um breve histórico sobre CLPs, sua utilização na indústria e suas principais características, com especial interesse nos diferentes ciclos de execução do CLP e nas linguagens de programação definidas pela norma IEC 61131-3.

---

Na segunda Seção, apresentou-se o algoritmo utilizado para a obtenção do modelo *fuzzy* TS-GOBF a partir dos dados do processo a ser controlado, amostrados pelo CLP, e utilizando a metodologia detalhada nos Capítulos anteriores. Detalhou-se também os passos necessários para o cálculo das ações dos controladores locais e o cálculo da ação de controle global. As etapas de determinação do modelo do processo e do cálculo das ações de controle apresentadas neste capítulo, foram implementadas utilizando a linguagem de programação Texto Estruturado (detalhada anteriormente) e definidas com um bloco de função do CLP. Desta forma, tem-se um sistema de controle preditivo que pode ser adaptado para operar em plantas com diferentes características dinâmicas.

O capítulo a seguir apresenta os resultados práticos da implementação proposta nesta dissertação e a comparação destes resultados com algoritmos de controle clássico, muito utilizados na indústria.

## 6 Aplicações e Resultados

O presente capítulo tem como objetivo apresentar aplicações e resultados das propostas apresentadas neste trabalho, que são a identificação e otimização de sistemas dinâmicos utilizando modelos *fuzzy* TS-GOBF e a implementação de um controlador preditivo não-linear que utiliza multimodelos lineares em um Controlador Lógico Programável comercial.

### 6.1 Descrição Sistema de Nível

Para a validação das propostas de modelagem e controle apresentadas nesta dissertação, utilizou-se duas plantas didáticas para controle de nível, como apresentado na Figura 26.

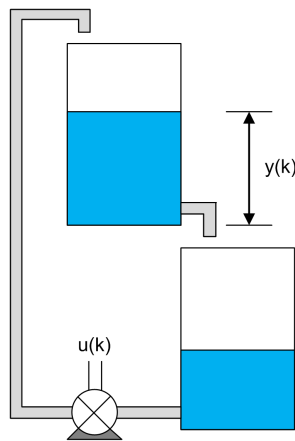


Figura 26: Sistema de Controle de Nível.

A ação de controle atua sobre a alimentação elétrica da bomba, localizada no tanque inferior, alterando a vazão do líquido bombeado para o tanque superior e consequentemente alterando a altura do nível  $y(k)$  (variável controlada). As principais causas de não-linearidade em sistema de nível são o fluxo de saída do líquido que é proporcional a  $\sqrt{y}$ , segundo a lei de Bernoulli para pequenos orifícios, e a saturação existente em  $y$  devido à posição do orifício acima da base do tanque.

Os dois sistemas de nível utilizados apresentam um funcionamento semelhante, diferenciando apenas na capacidade de armazenamento dos tanques e no tipo de sensor utilizado para realizar a medida da altura da coluna de líquido. Para a primeira planta, fabricada pela Datapool Equipamentos Didáticos e utilizada na abordagem linear, a medida da coluna de líquido é feita utilizando um sensor capacitivo, na qual a altura do líquido altera o valor da capacitância. Já para a segunda planta de nível, fabricada pelo Festo Didatic e utilizada na abordagem não-linear, a medida da coluna de líquido é realizada através de um sensor ultrassônico. Para comprovar a eficiência e viabilidade do sistema de controle proposto neste trabalho, serão realizadas comparações da resposta dos sistemas dinâmicos utilizando um algoritmo de controle clássico PID e o sistema de controle preditivo *fuzzy* TS-GOBF.

## 6.2 Modelagem e controle de nível em um tanque - Abordagem Linear

Como primeiro exemplo de aplicação do sistema de controle abordado neste trabalho, propõe-se o controle de uma planta de nível didática com dois tanques acoplados, como apresentado Figura 27, utilizando uma abordagem linear.



Figura 27: Sistema de Nível Didático - Datapool.

Como apresentado na Seção 5.2, o primeiro passo para implementar o sistema de controle preditivo proposto é a obtenção do modelo do sistema a ser controlado. Neste trabalho, os dados do processo utilizados para a obtenção do modelo GOBF e para a implementação do sistema de controle, foram amostrados utilizando um CLP comercial modelo 4PP045.0571-62 fabricado pela B&R Automation, apresentado na Figura 28, com as seguintes características:



Figura 28: CLP utilizado na identificação e controle dos sistemas de nível.  
(B&R-AUTOMATION, 2008)

- Processador: ELAN SC520 100 MHz;
- Memória Principal: 64 MB DRAM;
- Memória de Programa: 48 KB SRAM;
- Interface: Ethernet 10/100 Mbit/s;
- Tempo de Ciclo Mínimo: 4 ms;
- Display: LCD 5.7 polegadas com Touch Screen resistivo;
- Módulos: Entrada e Saída analógica (4 a 20 mA ou 0 a 10 Vdc) com conversor A/D de 12 bits;

Para a identificação do sistema de nível, aplicou-se um sinal de tensão na entrada do processo (tensão na bomba elétrica) e os dados de entrada/saída, ou seja, a tensão de alimentação da bomba e altura da coluna de líquido, foram amostrados com um tempo de amostragem  $T_s = 0,5$  segundos. Estes dados foram armazenados em dois vetores com 1480 posições totalizando 740 segundos de ensaio. A Figura 29 apresenta os dados do ensaio do sistema em malha aberta.

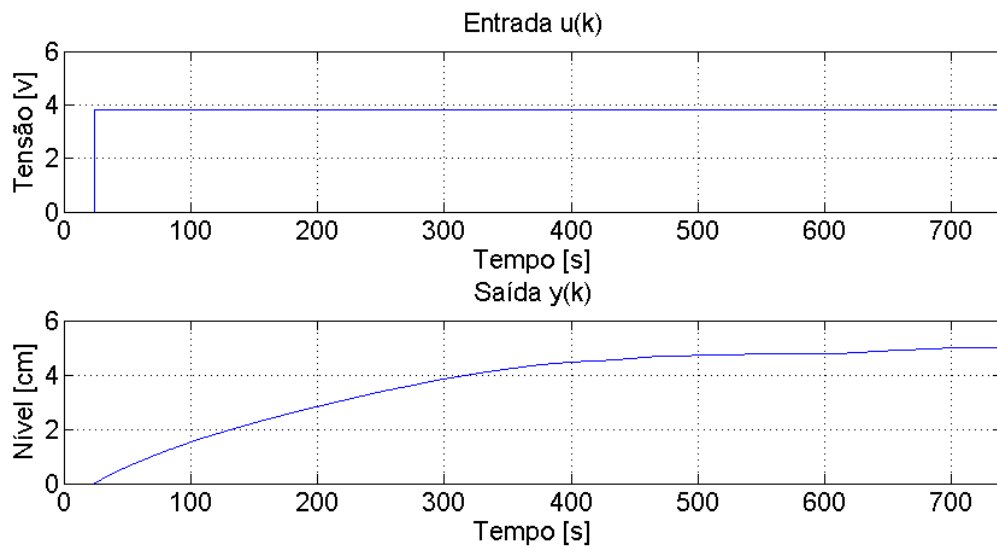


Figura 29: Dados utilizados na identificação do sistema - Abordagem Linear.

Analisando os dados obtidos no ensaio, pode-se concluir que o sistema em questão apresenta comportamento típico de um sistema de primeira ordem. Aplicando o algoritmo de identificação propostos em Machado (2011), tem-se que os parâmetros do modelo que melhor representa o sistema sob análise, através da representação em espaço de estados apresentado em (2.27), são dados por:

$$A = \begin{bmatrix} 0,9977 & 0 \\ 0,0046 & 0,9977 \end{bmatrix}$$

$$B = \begin{bmatrix} 0,0677 & -0,0675 \end{bmatrix}^T$$

$$C = \begin{bmatrix} 0,0407 & 1,3412 \cdot 10^{-6} \end{bmatrix}$$

Analisando os autovalores da matriz  $A$  conclui-se que os polos do modelo se encontram em 0.9977. Estes valores estão coerentes com o comportamento do sistema em questão.

Por utilizar uma abordagem linear do sistema de controle preditivo em questão, nesta primeira etapa do trabalho, o modelo matricial apresentado será utilizado em todas as faixas de operação da planta, ou seja, o controlador utilizará um único modelo local para o cálculo das ações de controle. Para fins de validação, é possível observar na Figura 30 que o modelo do sistema dinâmico identificado representa com boa exatidão o sistema dinâmico a ser controlado.

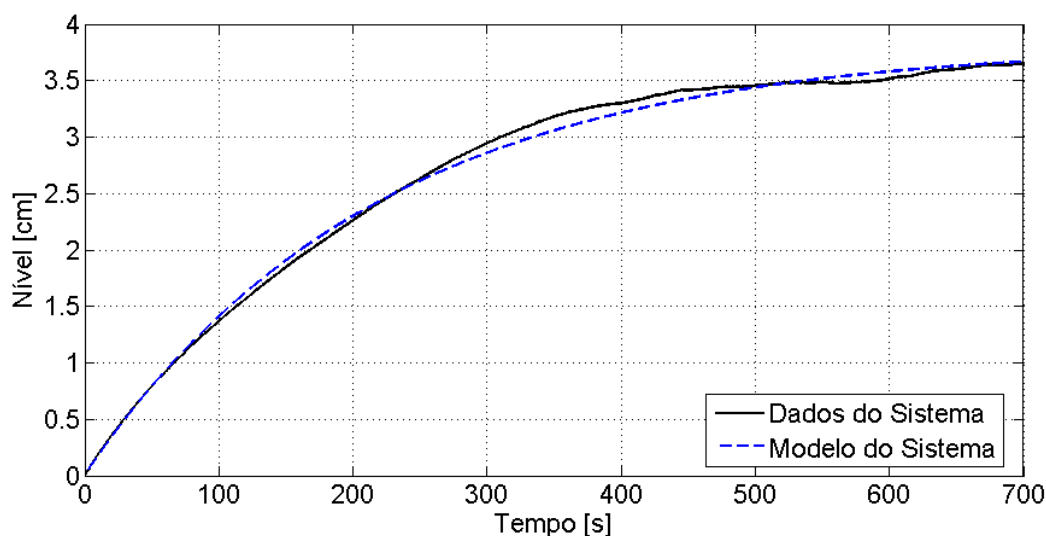


Figura 30: Comparação entre o sistema real e o modelo obtido.

Para a implementação do sistema de controle, estabeleceu-se um horizonte de predição  $N_y = 10$ , um horizonte de controle  $N_u = 4$  e o fator de ponderação das ações de controle  $\lambda = 0.1$ . Com esses parâmetros pretende-se atender as especificações de projeto que são: *overshoot* máximo de 5%, tempo de acomodação máximo de 30 segundos e erro em regime permanente inferior a 2%.

Utilizando as matrizes do modelo  $A$ ,  $B$  e  $C$ , e os parâmetros do sistema de controle  $N_y$ ,  $N_u$  e  $\lambda$ , as matrizes  $G$  (equação (4.14)) e  $Q^{-1}$  (equação (4.22)) foram calculadas de maneira *offline* e embarcadas no CLP na rotina de controle preditivo implementada. O cálculo *offline* das matrizes  $G$  e  $Q^{-1}$  propiciam um ganho computacional significativo ao sistema, pois evitam a inversão de matrizes de ordem  $N_u \times N_u$  na rotina de controle do CLP.

Com objetivo de verificar o funcionamento do sistema de controle abordado, foram estabelecidos dois degraus consecutivos de referência, em 5 e 7 [cm] no nível do tanque superior, ambos com duração de 75 segundos. A Figura 31 apresenta a resposta do sistema em malha fechada.



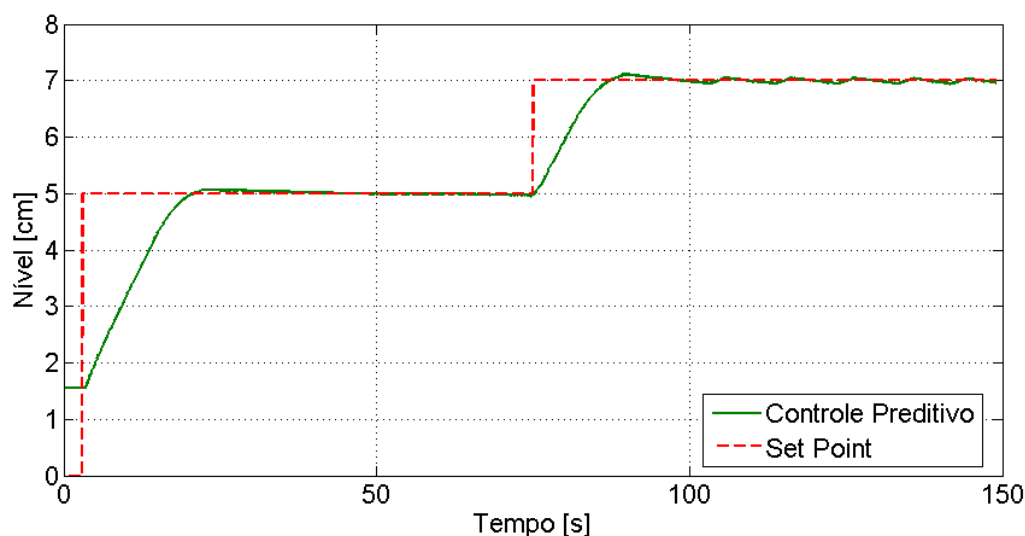


Figura 31: Resposta do sistema de nível - Controlador Preditivo Linear.

Como pode-se observar, o sistema de controle preditivo embarcado no CLP, atuou corretamente na planta, fazendo com que o nível desejado (*Set Point*) fosse atingido com um baixo valor de *overshoot*, em torno de 1%, e tempo de acomodação de aproximadamente 21 segundos. Com o objetivo de validar o controle preditivo desenvolvido, realizou-se uma comparação entre a resposta do sistema dinâmico utilizando o Controlador Preditivo GOBF e um controlador PID clássico, como apresentado na equação (6.1).

$$D(z) = K_p + K_i \frac{T_s z - 1}{2 z + 1} + K_d \frac{z - 1}{T_s z} \quad (6.1)$$

Esta comparação se justifica pois este tipo de controlador é o mais utilizado na indústria (ASTRÖM; HÄGGLUND, 1995). Esta popularidade se deve principalmente à simplicidade no ajuste dos seus parâmetros e por estar disponível em quase todos os CLPs.

Os parâmetros do controlador PID,  $K_p$ ,  $K_i$  e  $K_d$  foram calculados utilizando o método do Lugar Geométrico das Raízes (PHILLIPS; HARBOR, 1991) considerando como parâmetro de projeto o tempo de subida com valor semelhante ao obtido através do Sistema de Controle Preditivo, aproximadamente 15 segundos. Os parâmetros obtidos foram:  $K_p = 20,24$ ,  $K_i = 2,00$  e  $K_d = 0,10$ . Assim como realizado para o controle preditivo, aplicou-se dois degraus consecutivos de referência 5 e 7 [cm] no nível do tanque superior, com duração de 75 segundos cada. A resposta do sistema de nível controlada por um algoritmo PID, pode ser observada na Figura 32. A Figura 33 apresenta a comparação entre os dois sistemas de controle atuando na planta de nível em questão.

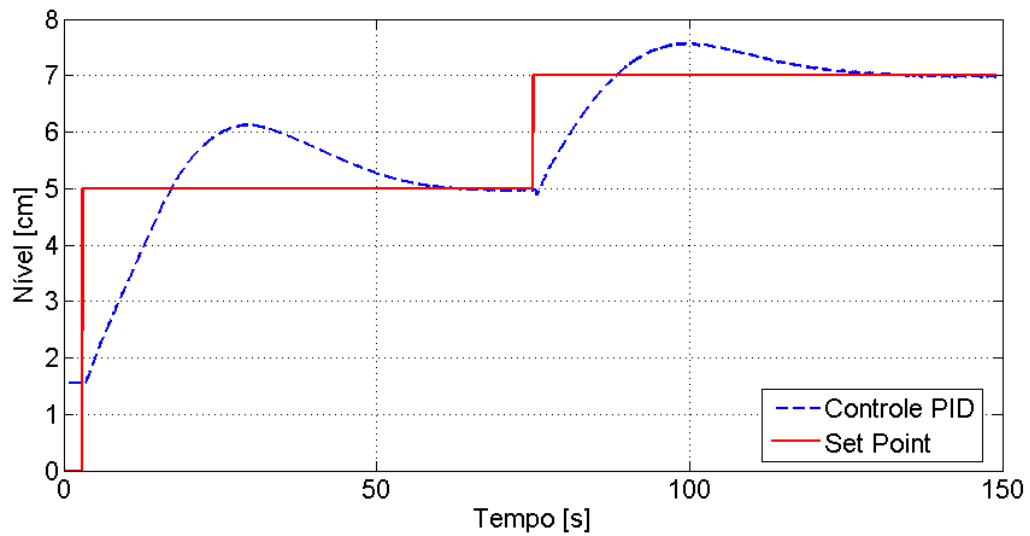


Figura 32: Resposta do sistema de nível - Controlador PID.

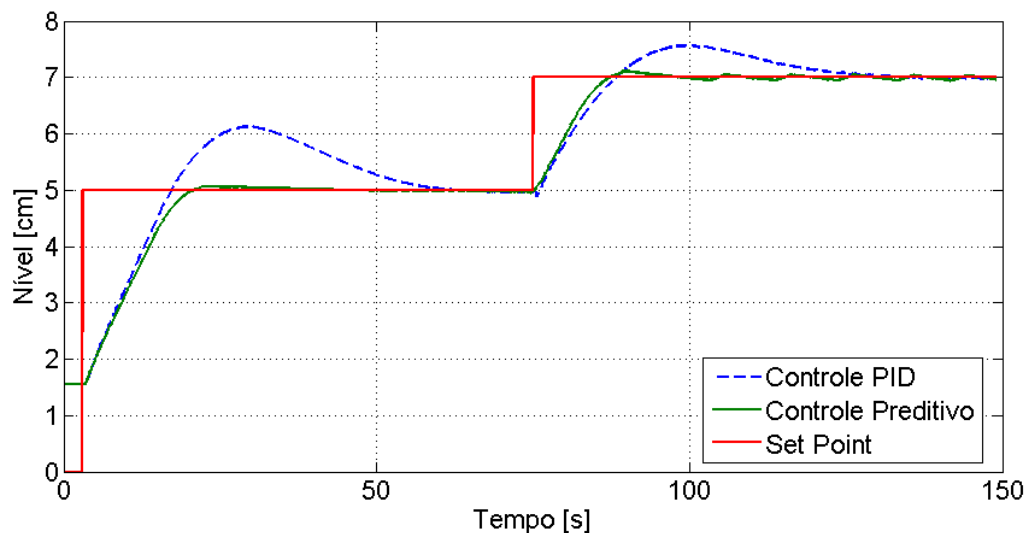


Figura 33: Comparação entre as respostas do Controlador PID e o Controlador Preditivo Linear.

Conforme apresentado na Figura 33, em ambos sistemas de controle a planta atingiu os valores de nível especificados, no entanto, pode-se observar que para o controlador preditivo proposto o sistema apresenta tempo de acomodação e *overshoot* inferiores quando comparados ao sistema controlado pelo PID.

Apesar de apresentar um desempenho superior ao sistema de controle PID para o sistema de nível em questão, o Controlador Preditivo não apresentou um custo computacional significativamente superior àquele apresentado pelo algoritmo PID. Os valores

de utilização da CPU do CLP para cada um dos controladores durante sua execução, obtidos através de um sistema de monitoramento denominado *Profiler*, são apresentados na Tabela 2.

Tabela 2: Custo Computacional dos Controladores - Abordagem Linear.

Controlador	Tempo de Ciclo	Utilização CPU
PID	500 ms	1,58%
Preditivo	500 ms	1,58%

Os resultados apresentados na Figura 33 e na Tabela 2 mostram que o sistema de controle preditivo proposto nesta dissertação, proporcionou um significativo ganho no comportamento dinâmico da planta de nível quando comparado ao controlador PID, sem aumento na taxa de utilização do CLP, uma vez que para os dois controladores a taxa de utilização da CPU foi a taxa mínima, em torno de 1,58%. Mesmo o sistema de controle preditivo apresentando um resultado satisfatório, é possível observar na Figura 31, que a resposta do sistema não apresenta características dinâmicas semelhantes para todas as faixas de operação da planta. Este fato se deve à utilização de um controlador linear no controle de uma ampla faixa de operação de um sistema intrinsecamente não-linear. Esta característica motiva a utilização de uma abordagem não-linear no controle do sistema de nível em uma ampla faixa de operação.

### 6.3 Modelagem e controle de nível em um tanque - Abordagem Não-Linear

Para exemplificar a utilização do sistema de controle preditivo não-linear apresentado neste trabalho, utilizou-se uma segunda planta de nível didática desenvolvida pela FESTO DIDATIC, como apresentado na Figura 34.

A fim de realizar a identificação e validação do modelo do sistema utilizando modelos *Fuzzy* TS-GOBF, utilizou-se dois conjuntos de dados de entrada/saída do sistema, amostrados pelo Controlador Lógico Programável (descrito na Seção 6.2). Para a aquisição do primeiro conjunto de dados, utilizados para a identificação, aplicou-se 10 degraus consecutivos na bomba elétrica com duração de 360 segundos cada. Este conjunto de dados, apresentado na Figura 35, possui 3600 pares de amostras de entrada/saída com tempo de amostragem de 1 segundo, totalizando 60 minutos de ensaio.



Figura 34: Planta Didática de Nível - Festo Didatic.

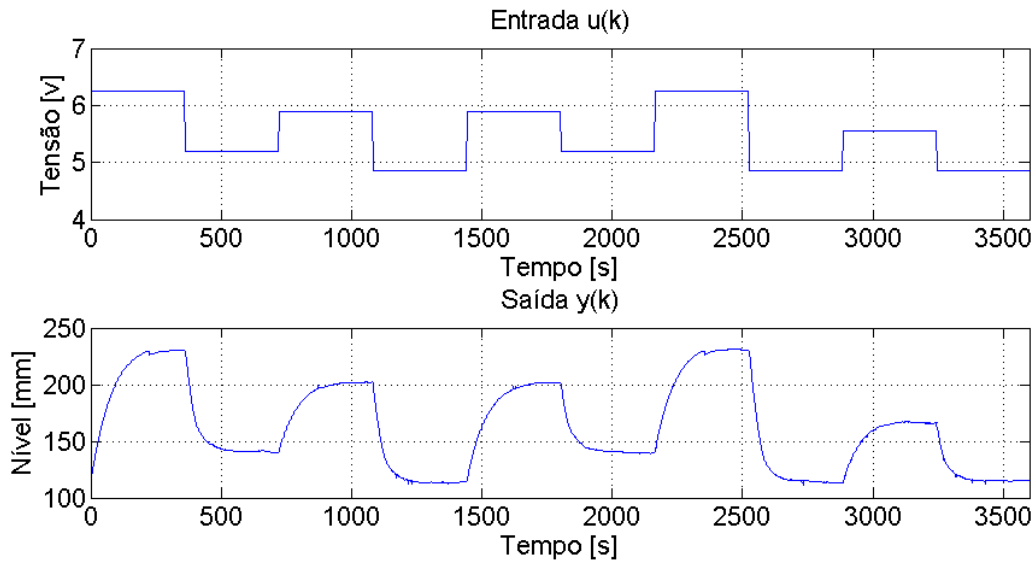


Figura 35: Dados utilizados na estimação do modelo.

Como apresentado no fluxograma da Figura 16, obtidos os dados do sistema a ser modelado, deve-se escolher os parâmetros do modelo *fuzzy* TS-GOBF inicial, ou seja, o número dos polos e funções do modelo linear GOBF, e o número de variáveis dos antecedentes das regras. Para o sistema em questão, utilizou-se inicialmente nos consequentes das regras, modelos locais com duas funções ortonormais generalizadas na base com polos escolhidos aleatoriamente, e uma variável (saída do primeiro filtro ortonormal) nos antecedentes das regras. Para a obtenção do modelo *fuzzy* TS inicial, utilizou-se o algoritmo de GK aplicados aos dados de identificação.

Obtidos os grupos, é possível a partir destes determinar as funções de pertinência dos antecedentes das regras através da projeção dos valores de ativação dos objetos nos subespaços de entrada, como descrito na Seção 3.4.1, e calcular os coeficientes dos modelos locais presentes nos consequentes das regras, utilizando para este caso, estimação local (Seção 3.4.2). O modelo *fuzzy* TS-GOBF inicial é composto por 3 regras *fuzzy* (modelos locais) cujas funções de pertinência são apresentadas na Figura 36. A Figura 37 apresenta a comparação entre a saída do modelo *fuzzy* TS-GOBF inicial e a saída amostrada do sistema dinâmico, utilizando um grupo de dados de validação.

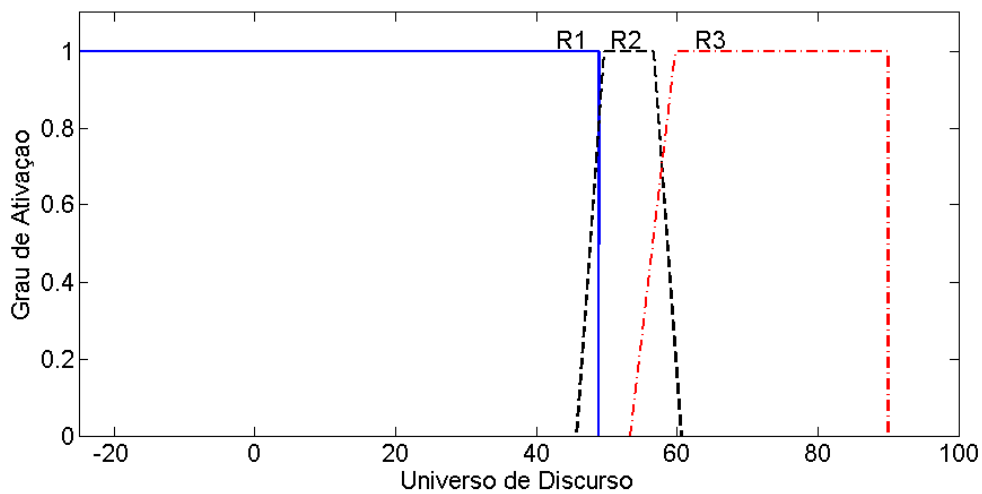


Figura 36: Funções de pertinência presentes nos antecedentes das regras do modelo TS-GOBF inicial.

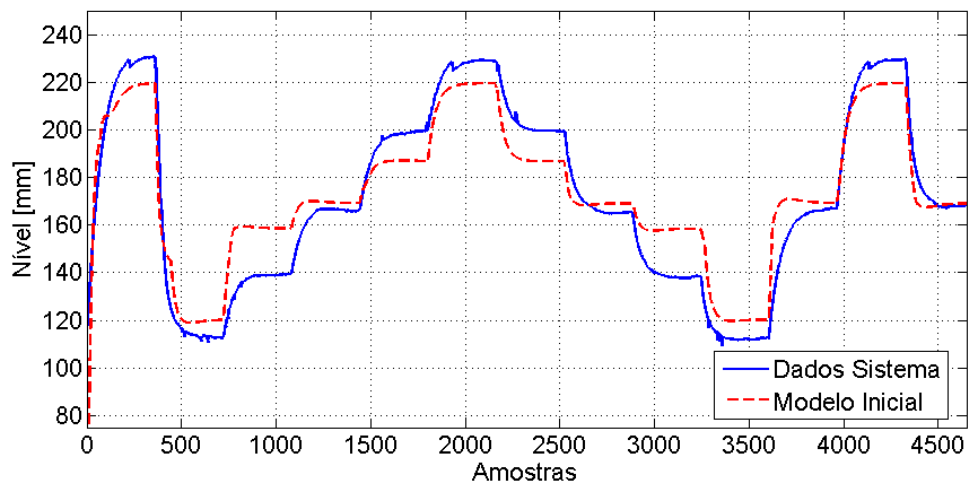


Figura 37: Comparação entre o sistema real o modelo *fuzzy* TS-GOBF inicial.

Como pode-se observar na Figura 37, o modelo *fuzzy* TS-GOBF inicial, obtido através dos passos detalhados anteriormente, não representa o sistema dinâmico de nível com fidelidade, apresentando um erro médio quadrático igual  $EQM = 140,6043$ . Desta forma, para obtenção do modelo que melhor represente o sistema, aplicou-se o algoritmo de simplificação dos modelos *fuzzy* (Seção 3.4.3) e o algoritmo de otimização (Seção 3.5) propostos no trabalho de Machado (2011). O modelo *fuzzy* TS-GOBF otimizado é composto por 3 regras *fuzzy* (modelos locais) cujas funções de pertinência são apresentadas na Figura 38. O consequente de cada regra é formado por duas funções ortonormais generalizadas de primeira ordem com dois polos reais  $p_1 = 0,985$  e  $p_2 = 0,984$ . A comparação entre a saída do modelo otimizado e a saída amostrada do sistema dinâmico, é apresentada na Figura 39. Como pode-se observar, o modelo *fuzzy* TS-GOBF otimizado representa com fidelidade o sistema de nível utilizado, com um erro quadrático médio  $EQM = 3,1703$ .

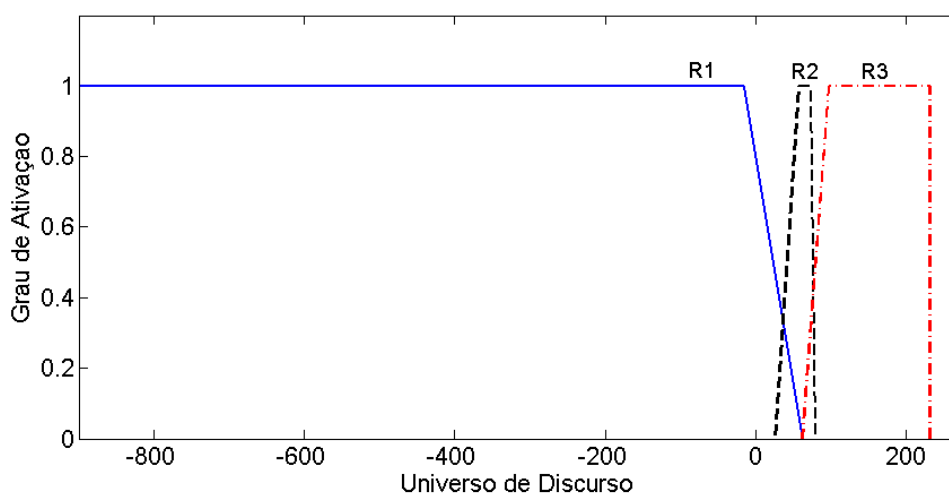


Figura 38: Funções de pertinência presentes nos antecedentes das regras do modelo TS-GOBF otimizado.

Obtido o modelo adequado para representação do sistema a ser controlado, a próxima etapa para a implementação do sistema de controle é escolher os parâmetros dos controladores preditivos lineares que atuarão nos modelos locais. Para o sistema em questão, estabeleceu-se experimentalmente um horizonte de predição  $N_y = 10$ , um horizonte de controle  $N_u = 2$  e o fator de ponderação igual sobre a ação de controle  $\lambda = 200$ . Utilizando os parâmetros estabelecidos e as matrizes  $A$ ,  $B$  e  $C_{local}$ , obtidas diretamente do modelo otimizado, as matrizes  $G$  (equação (4.14)) e  $Q_{local}^{-1}$  (equação (4.22)) foram calculadas de maneira *offline* utilizando o software MATLAB<sup>®</sup>, e embarcadas no Controlador Lógico Programável modelo B&R 4PP045.0571-62, já detalhado anteriormente.

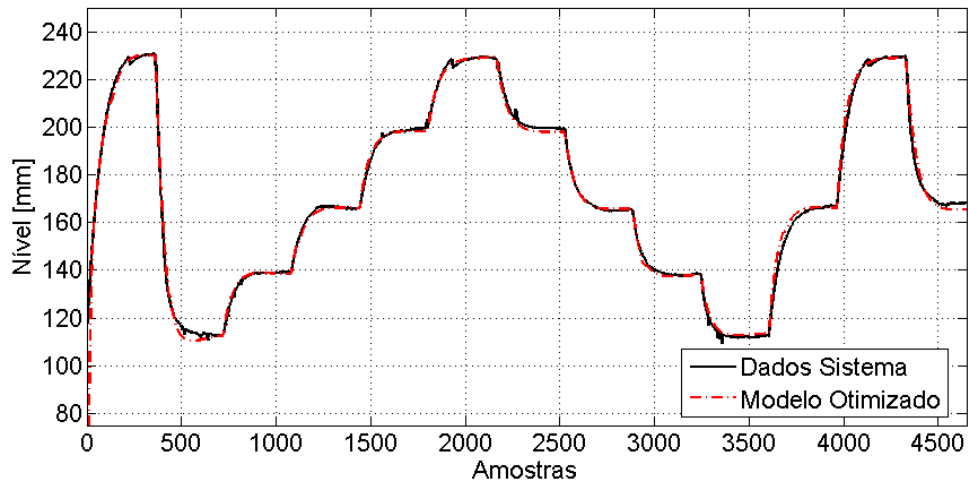


Figura 39: Comparação entre o sistema real o modelo *fuzzy* TS-GOBF final.

Para verificar o funcionamento do sistema de controle para sistemas não-lineares proposto neste trabalho, foram aplicados 8 degraus consecutivos de referência para o nível do tanque superior. A resposta do sistema para o controlador preditivo não-linear, pode ser observada na Figura 40.

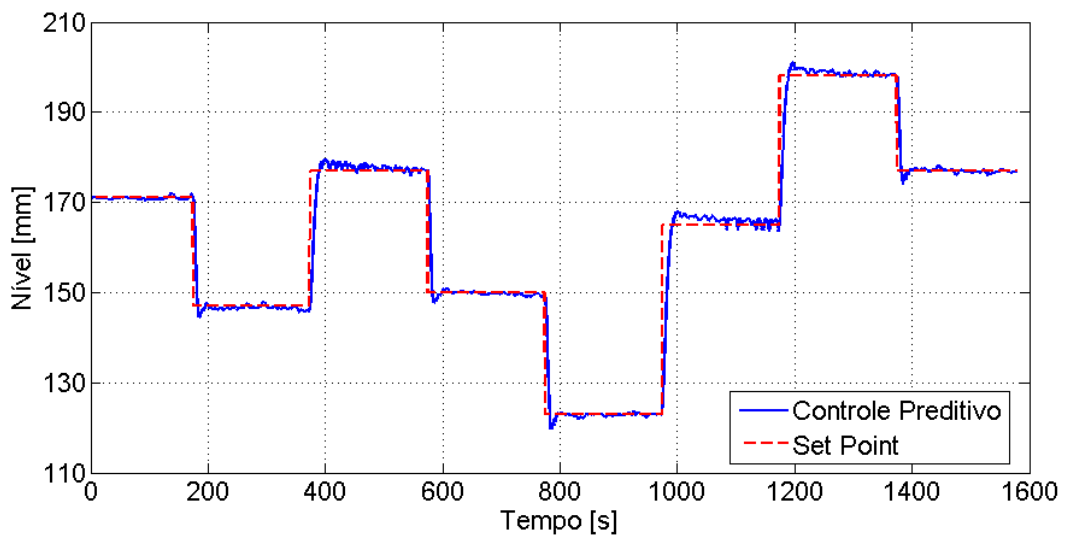


Figura 40: Resposta do Sistema de Nível - Controlador Preditivo Não-Linear.

É possível observar na Figura 40, que a resposta do sistema de nível controlada atingiu os níveis de referência aplicados, com um tempo de acomodação  $T_a < 35$  s e *overshoot* máximo em torno de 1%, valores abaixo do especificado no projeto do controlador. Diferente da abordagem linear, apresentada anteriormente (Seção 6.2), o sistema de nível apresentou um comportamento semelhante em todas as faixas de operação. Essa característica comprova a ação do sistema de controle preditivo em questão, nas não linearidades da planta de nível.

Por se tratar de uma aplicação para controladores industriais, realizou-se uma comparação entre a resposta da planta utilizando o Controlador Preditivo *fuzzy* TS-GOBF e um controlador PID já implementado em um bloco específico do CLP denominado *LCRS-limPID*. Para a sintonia deste controlador, utilizou-se o método do Lugar Geométrico das Raízes com parâmetro de projeto o tempo de acomodação com valor semelhante ao apresentado na resposta com o controlador preditivo (aproximadamente 30 segundos). Os parâmetros obtidos foram:  $K_p = 0,1535$ ,  $K_i = 0,1556$  e  $K_d = 0,10$ . Assim como realizado para o controlador preditivo, aplicou-se oito degraus consecutivos na planta. A resposta do sistema para o controle PID pode ser observada na Figura 41 e a comparação da resposta entre os dois controladores na Figura 42.

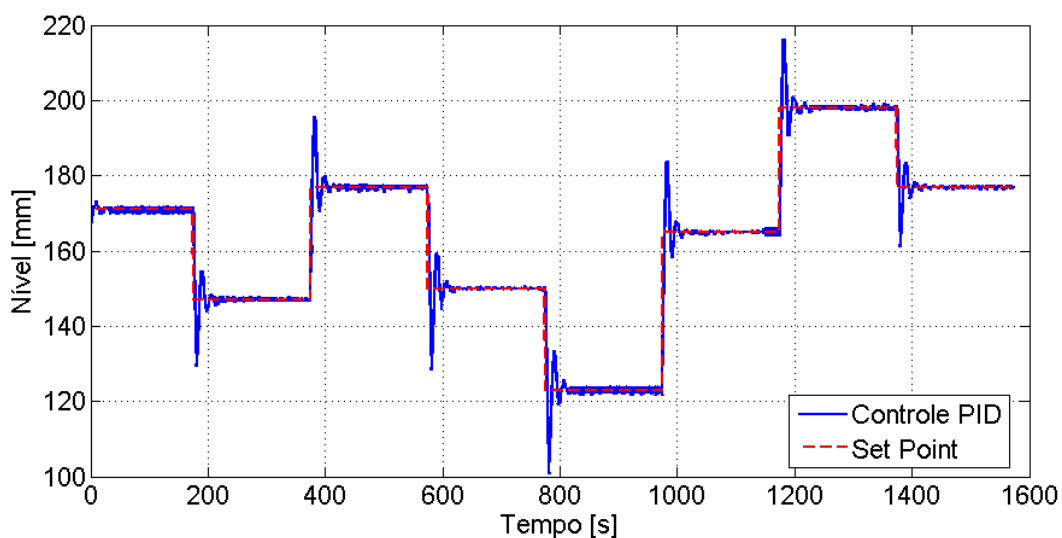


Figura 41: Resposta do Sistema de Nível - Controlador PID.



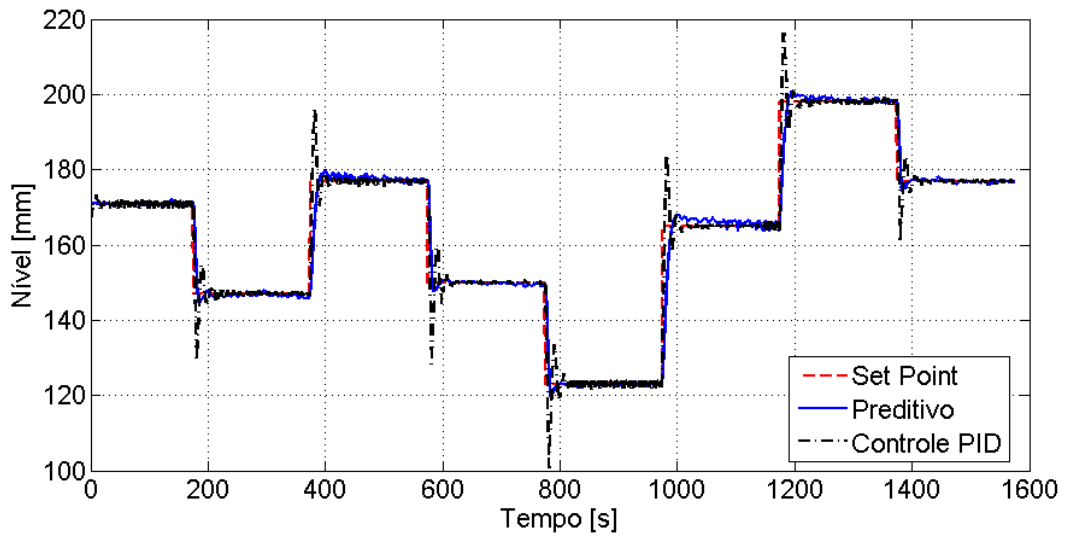


Figura 42: Comparação entre as respostas do Controlador PID e o Controlador Preditivo Não-Linear.

Como observado na Figura 42, os dois controladores fizeram com que o sistema de nível atingisse os valores de referência aplicados, porém é possível perceber na Figura 42, que o sistema de controle PID apresentou um *overshoot* maior que aquele apresentado pelo controlador preditivo. Também é possível observar que o *overshoot* apresentado, nas diferentes regiões de operação, não é o mesmo. Esse comportamento demonstra a não-linearidade do sistema sob controle, e a deficiência do algoritmo PID quando utilizado no controle de plantas não lineares.

Assim como realizado para a abordagem linear do controlador preditivo proposto, realizou-se a comparação da taxa de utilização da CPU do CLP para o controle do sistema de nível utilizando o algoritmo PID e utilizando o controlador preditivo *fuzzy* TS-GOBF. É possível observar na Tabela 3, que mesmo o controlador preditivo apresentando um desempenho superior ao controlador PID, a taxa de utilização da CPU não foi significativamente superior, comprovando a viabilidade de implementação do algoritmo proposto em CLPs comerciais de pequeno e médio porte.

Tabela 3: Custo Computacional dos Controladores - Abordagem Não-Linear

Controlador	Tempo de Ciclo	Utilização CPU
PID	1 s	1,76%
Preditivo	1 s	2,98%

Os resultados apresentados na Figura 42 e na Tabela 3, demonstram que com a utilização do sistema de controle preditivo proposto, obteve-se um significativo ganho no comportamento dinâmico do sistema sob controle, sem aumento expressivo na taxa de utilização da CPU do CLP, quando comparado com o algoritmo de controle clássico PID.

## 6.4 Resumo

Este capítulo teve como objetivo apresentar os resultados obtidos com a utilização do controlador preditivo proposto, embarcado em um CLP, em sistemas dinâmicos reais.

Na primeira Seção apresentou-se a aplicação do sistema de controle preditivo com modelos GOBF para controle de uma planta didática de nível utilizando uma abordagem linear. Inicialmente o sistema de nível foi identificado utilizando a metodologia apresentada no capítulo anterior. O modelo GOBF obtido, juntamente com os parâmetros e matrizes do controlador preditivo foram embarcados no CLP. Como se pode verificar através dos ensaios apresentados, utilizando o controlador preditivo GOBF o sistema de nível apresentou uma resposta superior quando comparada com o sistema de nível controlado por um algoritmo PID, sem aumento significativo no custo computacional do CLP. Apesar do sistema ter uma resposta adequada para controlador preditivo, foi possível observar que a resposta se deteriora para amplas faixas de operação, comprovando assim a não-linearidade do sistema de nível utilizado e a deficiência da abordagem linear deste tipo de controlador para controle de sistemas não-lineares.

Para exemplificar a utilização do sistema de controle proposto para sistemas não-lineares, a segunda Seção trouxe a aplicação do controlador *fuzzy* TS-GOBF em uma segunda planta de nível didática. Identificou-se o sistema de nível em uma abordagem não-linear utilizando modelos *fuzzy* TS-GOBF, como apresentado no algoritmo da Figura 16. Como se pode comprovar, através dos resultados apresentados, o modelo inicial *fuzzy* TS-GOBF não foi adequado para a representação do sistema a ser controlado, desta forma aplicou-se o método de otimização do modelo proposto por Machado (2011) e detalhado na Seção 3.5. Para aplicação do controlador preditivo na referida planta, o modelo otimizado e os parâmetros e matrizes dos controladores locais foram embarcados no CLP. Para comprovar a eficiência do controlador, utilizou-se um conjunto de valores como referência para a altura da coluna de líquido do tanque superior, e como pôde ser observado nos gráficos apresentados, o sistema de controle apresentou um resultado superior para todas as faixas de operação, quando comparado com um sistema de controle clássico PID. Para

---

comprovar a viabilidade do sistema de controle preditivo, a taxa de utilização da CPU do CLP foi comparada entre os dois sistemas de controle utilizados na planta, e mesmo o sistema de controle preditivo *fuzzy* TS-GOBF apresentou um resultado superior, o custo computacional não foi significativamente maior.

Desta forma, fica exemplificada a eficiência e viabilidade de implementação dos métodos de modelagem e controle de sistemas não-lineares.

## 7 Conclusão

Na primeira etapa deste trabalho, apresentou-se uma abordagem para modelagem de sistemas dinâmicos lineares e não-lineares estáveis, através de base de funções ortonormais generalizadas com funções internas em estrutura *ladder*. A utilização de GOBF com estrutura *ladder* na modelagem de sistemas dinâmicos possibilita representar sistemas com múltiplos modos utilizando somente parâmetros reais independente da natureza (reais ou complexos) dos polos. Para a modelagem de sistemas não-lineares, empregam-se modelos *fuzzy* Takagi-Sugeno com GOBF no consequente das regras dos modelos. A representação de sistemas dinâmicos utilizando BFOs apresenta uma série de vantagens quando comparada a outros modelos normalmente utilizados para representação de um mesmo sistema. Dentre as principais vantagens das BFOs, pode-se destacar a ausência de realimentação da saída, a possibilidade da utilização de um número menor de parâmetros no modelo para alcançar uma dada exatidão e a capacidade de lidar de forma robusta com atraso de transporte. Por apresentar tais características, as BFO vem sendo utilizados com sucesso na modelagem de sistemas dinâmicos, como apresentado em (MACHADO, 2007; MACHADO, 2011; CAMPELLO; OLIVEIRA; AMARAL, 2007; MEDEIROS; AMARAL; CAMPELLO, 2006). Desta forma, quando se utiliza modelos *fuzzy* TS-GOBF para representação de sistemas dinâmicos não-lineares, além das características inerentes a modelagem por BFO, outra vantagem encontrada é que os modelos *fuzzy* são aproximadores universais de funções em um espaço compacto e agregam a propriedade de interpretabilidade, úteis no projeto de sistemas de controle.

O principal objetivo dessa dissertação foi utilizar a representação de sistemas por modelos *fuzzy* TS-GOBF no projeto de um sistema de controle preditivo, embarcado em um Controlador Lógico Programável (CLP). O modelo *fuzzy* TS-GOBF do sistema a ser controlado, é obtido através da metodologia proposta por Machado (2011), na qual um modelo *fuzzy* TS-GOBF inicial, sem conhecimento a priori do sistema, é otimizado para obtenção do modelo final. Para a determinação das regras do modelo, utilizou-se o algoritmo de agrupamento *fuzzy* proposto por Gustafson e Kessel (1979), no qual as amostras

dos sinais de entrada/saída do sistema são agrupadas para determinar os antecedentes das regras. O número de regras (modelos locais), que melhor agrupam os dados em questão, é determinado utilizando uma combinação de critérios de avaliação de agrupamentos (MACHADO, 2007). Esta metodologia, pode gerar modelos que apresentem redundância de regras e/ou funções de pertinência do modelo *fuzzy*. Desta forma, utilizou-se algoritmos que simplificam os modelos quando há redundância (KAYMAK; BABUSKA, 1995; MACHADO, 2011). Obtidas as regras simplificadas do modelo, os consequentes das regras são obtidos através do método dos Mínimos Quadrados e os parâmetros das GOBF, dos consequentes das regras, são calculados utilizando método de otimização não linear.

Determinados os modelos locais e estabelecidos os parâmetros necessários, os controladores preditivos locais, atuando nos modelos locais (OLIVEIRA, 1997), foram embarcados em um CLP juntamente com o modelo *fuzzy* TS-GOBF do sistema a ser controlado, como detalhado no Capítulo 5. Através da combinação ponderada das ações dos controladores locais, a ação de controle global é obtida e aplicada ao sistema a ser controlado.

No Capítulo 6 apresentou-se dois exemplos práticos do sistema de controle proposto embarcado em um CLP. No primeiro exemplo realizou-se a identificação e controle de uma planta didática de nível utilizando uma abordagem linear do controlador, ou seja, a identificação foi realizada em torno de um ponto de operação do processo e um único modelo preditor no sistema de controle. No segundo exemplo, uma segunda planta didática de nível foi utilizada para ilustrar o método de identificação e controle propostos agora em uma abordagem não-linear. Para comprovar a eficiência da metodologia proposta, realizou-se a comparação entre a resposta dinâmica dos sistemas de nível utilizando o controlador preditivo proposto, e a resposta dos sistemas utilizando um algoritmo de controle clássico PID. Para as duas abordagens, o sistema de controle proposto apresentou um desempenho superior ao apresentado pelo PID, sem o aumento significativo da taxa de utilização do CLP.

Tendo apresentado a proposta e implementação do controlador, pode-se concluir que o controlador preditivo proposto funciona de maneira adequada no controle de sistemas dinâmicos reais. Sendo assim, a proposta apresentada fornece uma alternativa no controle de processos industriais não-lineares, uma vez que esta abordagem pode ser implementada em um Controlador Lógico Programável comercial, diferentemente do que ocorre hoje, no qual sistemas de controle complexos são implementados em computadores dedicados de alto processamento e alto custo.

## 7.1 Contribuições

Neste trabalho, destacam-se algumas contribuições no sentido de tornar possível as implementações propostas, ou o aperfeiçoamento das técnicas já utilizadas. Algumas destas contribuições podem ser destacadas:

- Proposta de um sistema de controle preditivo não-linear multimodelos, utilizando as bases de funções ortonormais generalizadas (GOBF) com estrutura *ladder* nos consequentes das regras *fuzzy* TS, possibilitando o controle de sistemas dinâmicos de qualquer ordem, através de Base de Funções Ortonormais parametrizadas somente por coeficientes reais.
- Desenvolvimento e Implementação do sistema de controle preditivo *fuzzy* TS-GOBF em Controladores Lógicos Programáveis. Apresentou-se os passos necessários para embarcar o sistema de controle preditivo proposto utilizando um Controlador Lógico Programável (CLP) comercial.

## 7.2 Trabalhos Futuros

Como continuidade do trabalho em questão, a seguir serão apresentadas algumas propostas que irão complementar o que foi desenvolvido até aqui:

- Desenvolvimento e Implementação de um algoritmo adaptativo para identificação, parametrização e controle de sistemas dinâmicos não-lineares, utilizando Controladores Lógicos Programáveis. Uma das principais propostas de continuação deste trabalho, consiste em implementar em um CLP um algoritmo completo para identificação e controle de sistemas dinâmicos no qual o processo de identificação é realizado em tempo real. Este algoritmo permitiria identificar o sistema dinâmico linear/não-linear, e com base nas especificações de projeto, determinar o melhor conjunto de parâmetros para controle do sistema em questão.

## 7.3 Trabalhos publicados e em desenvolvimento

Os trabalhos a seguir foram desenvolvidos durante o mestrado:

- HENRIQUES, J. P. C.; MACHADO, J. B.; FERREIRA, L. H. C. Controle Preditivo Baseado em Modelos-BFO Embarcado em um Controlador Lógico Programável. In: XI Simpósio de Automação Inteligente. 2013. v. 1.
- HENRIQUES, J. P. C.; MACHADO, J. B.; FERREIRA, L. H. C. Sistema de Controle Preditivo Multimodelos *Fuzzy* TS-BFO Embarcado em um Controlador Lógico Programável. In: XI *International Conference on Industry Applications*. 2014. (submetido).

## *Referências*

- AGUIRRE, L. A. *Introdução à Identificação de Sistemas*. 2<sup>a</sup>. ed. Av. Antônio Carlos, 6627. Belo Horizonte, MG, Brasil: Editora UFMG, 2004. ISBN 85-7041-400-5.
- ALLGOWER, F.; ZHENG, A. *Nonlinear Model Predictive Control*. [S.l.]: Springer, 2000. (Progress in Systems and Control Theory). ISBN 9783764362973.
- ASTRÖM, K. J.; HÄGGLUND, T. Pid controllers: theory, design, and tuning. *Instrument Society of America, Research Triangle Park, NC*, 1995.
- BABUSKA, R. *Fuzzy Modeling for Control*. 1st. ed. 101 Philip Drive, Assinippi Park, Norwell, Massachusetts 02061 USA: Kluwer Academic Publishers, 1998. (International Series in Intelligent Technologies).
- BABUSKA, R.; VERBRUGGEN, H. An overview of fuzzy modeling for control. *Control Engineering Practice*, v. 4, n. 11, p. 1593–1606, 1996.
- BABUSKA, R.; VERBRUGGEN, H. Fuzzy set methods for local modelling and identification. In: *Multiple Model Approaches to Modelling and Control*. [S.l.]: Taylor and Francis, 1997.
- BAZARAA, M. S.; SHETTY, C. M. *Nonlinear Programming: Theory and Algorithms*. New York: Wiley, 1979.
- BEZDEK, J. *Pattern Recognition with Fuzzy Objective Function*. New York: Plenum Press, 1981. (1st).
- BEZDEK, J. C.; PAL, N. R. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 28, p. 301–315, 1998.
- BOYD, S.; CHUA, L. Fading memory and the problem of approximating nonlinear operators with volterra series. *Circuits and Systems, IEEE Transactions on*, v. 32, n. 11, p. 1150–1161, Nov 1985. ISSN 0098-4094.
- B&R-AUTOMATION. *User Manual Power Pannel 4PP045.0571-62*. 2008. Disponível em: <<http://www.br-automation.com/pt-br/products/operacao-e-visualizacao/power-panel-45/power-panel-45/4pp0450571-062>>.
- CAMACHO, E.; BORDONS, C. *Model Predictive Control*. [S.l.]: Springer London, 2004. (Advanced Textbooks in Control and Signal Processing). ISBN 9781852336943.
- CAMPELLO, R. J. G. B.; AMARAL, W. C. Takagi-Sugeno fuzzy models within orthonormal basis function framework and their application to process control. In: *Proc. of the 2002 IEEE Int. Conf. on Fuzzy Syst.* [S.l.: s.n.], 2002. v. 2, p. 1399–1404.



- CAMPELLO, R. J. G. B.; HRUSCHKA, E. R. A fuzzy extension of the silhouette width criterion for cluster analysis. *Fuzzy Sets and Systems*, v. 157, n. 21, p. 2858–2875, 2006.
- CAMPELLO, R. J. G. B.; MELEIRO, L. A. C.; AMARAL, W. C. Control of a bioprocess using orthonormal basis function fuzzy models. In: *Proc. 2004 IEEE Int. Conf. on Fuzzy Syst.* [S.l.: s.n.], 2004. v. 2, p. 801–806 vol.2. ISSN 1098-7584.
- CAMPELLO, R. J. G. B.; OLIVEIRA, G. H. C.; AMARAL, W. C. Identificação e Controle de Processos via Desenvolvimentos em Séries Ortonormais: Partes A (Identificação) e B (Controle). *Controle & Automação*, v. 18, n. 3, p. 298–332, 2007.
- CAMPO, P. J.; MORARI, M. Robust model predictive control. In: *American Control Conference, 1987.* [S.l.: s.n.], 1987. p. 1021–1026.
- CAPELLI, A. *Automação Industrial - Controle do Movimento e processo contínuos*. São Paulo, SP: Érica, 2008. ISBN 978-85-365-0117-8.
- CLARKE, D. *Advances in Model-Based Predictive Control*. [S.l.]: Oxford University Press, USA, 1994.
- CLARKE, D.; MOHTADI, C. Properties of generalized predictive control. *Automatica*, v. 25, n. 6, p. 859 – 875, 1989. ISSN 0005-1098.
- CLARKE, D.; MOHTADI, C. et al. Generalized Predictive Control. In: *Automatica*. [S.l.: s.n.], 1987. v. 23, n. 2, p. 137–160. Part i. the basic algorithm and part ii. extensions and interpretations.
- CUTLER, C.; RAMAKER, B. Dynamic Matrix Control - A computer control algorithm. In: *Joint Automatic Control Conference*. [S.l.: s.n.], 1980.
- DELGADO, M.; ZUBEN, F. V.; GOMIDE, F. Local and global estimation of takagi-sugeno consequent parameters in genetic fuzzy systems. In: *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*. [S.l.: s.n.], 2001. v. 3, p. 1247–1252 vol.3.
- DORF, R.; BISHOP, R. *Sistemas de controle modernos*. [S.l.]: LTC, 2009. ISBN 9788521617143.
- EVERITT, B.; LANDAU, S.; LEESE, M. *Cluster Analysis*. [S.l.]: Wiley, 2001. (A Hodder Arnold Publication). ISBN 9780340761199.
- FRANCHI, C. M.; CAMARGO, V. L. A. *Controladores Lógicos Programáveis*. São Paulo, SP: Érica, 2009. ISBN 978-85-365-0199-4.
- GATH, I.; GEVA, A. B. Unsupervised optimal fuzzy clustering. *IEEE transaction on pattern analysis and machine intelligence*, v. 2, n. 7, p. 773–781, 1989.
- GRAY, A. J.; MARKEL, J. A normalized digital filter structure. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, v. 23, n. 3, p. 268–277, Jun 1975.
- GUSTAFSON, D.; KESSEL, W. Fuzzy Clustering with a fuzzy covariance matrix. *Proc. IEEE CDC*, p. 761–766, 1979.

- HATHAWAY R. E BEZDEK, J. Switching regression models and fuzzy clustering. *IEEE transaction on Fuzzy Systems*, v. 1, n. 3, p. 195–204, 1993.
- HELLENDORRN, H.; DRIANKOV, D. *FUZZY MODEL IDENTIFICATION*. [S.l.]: Springer-Verlag, 1997. ISBN 9783540627210.
- HENRIQUES, J. P. C.; MACHADO, J. B.; FERREIRA, L. H. C. Controle Preditivo Baseado em Modelos-BFO Embarcado em um Controlador Lógico Programável. In: *XI Simpósio de Automação Inteligente*. Fortaleza: SBAI, 2013. v. 1.
- HEUBERGER, P. S. C.; Van den Hof, P. M. J.; WAHLBERG, B. *Modelling and Identification with Rational Orthogonal Basis Functions*. [S.l.]: Springer, 2005.
- HOF, P. M. J. van den; HEUBERGER, P. S. C.; BOKOR, J. System identification with generalized orthonormal basis functions. *Automatica*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 31, p. 1821–1834, December 1995. ISSN 0005-1098.
- III, F. J. D.; OGUNNAIKE, B. A.; PEARSON, R. K. Nonlinear model-based control using second-order volterra models. *Automatica*, v. 31, n. 5, p. 697 – 714, 1995.
- IRWING, E.; FALINOWER, C. M.; FONTE, C. Compatible cluster merging for fuzzy modelling. In: *2nd IFAC Workshop on Adaptative Control and Signal Processing*. [S.l.: s.n.], 1986. v. 2, p. 384–386.
- KAUFMAN, L.; ROUSSEEUW, P. *Finding Groups in Data: An Introduction to Cluster Analysis*. [S.l.]: Wiley, 2009. (Wiley Series in Probability and Statistics). ISBN 9780470317488.
- KAYMAK, U.; BABUSKA, R. Compatible cluster merging for fuzzy modelling. In: *Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium., Proceedings of 1995 IEEE Int.* [S.l.: s.n.], 1995. v. 2, p. 897–904 vol.2.
- KOSKO, B. Fuzzy systems as universal approximators. *IEEE Transaction on Computers*, v. 43, n. 11, p. 1329–1333, 1994.
- KOSKO, B. Fuzzy systems as universal approximators. *Computers, IEEE Transactions on*, v. 43, n. 11, p. 1329–1333, Nov 1994. ISSN 0018-9340.
- KRISHNAPURAM, R.; FREG, C.-P. Fitting an unknown number of lines and planes to image data through compatible cluster merging. *Pattern Recognition*, v. 25, n. 4, p. 385 – 400, 1992.
- LEWIS, D.; EVANS, C.; SANDOZ, D. Application of predictive control techniques to a distillation column. *Journal of Process Control*, v. 1, n. 4, p. 207 – 216, 1991.
- LJUNG, L. *System Identification*. 2nd. ed. Upper Saddle River, N.J: Prentice Hall PTR, 1999. (theory for the user). ISBN 0136566952.
- MACHADO, J. B. *Modelagem e controle preditivo usando multimodelos*. Dissertação (Mestrado) — DCA/FEEC/UNICAMP, Fevereiro 2007.

- MACHADO, J. B. *Modelagem de Sistemas Não-lineares por Base de Funções Ortonormais Generalizadas com Funções Internas*. Tese (Doutorado) — DCA/FEEC/UNICAMP, 2011.
- MACHADO, J. B.; AMARAL, W. C.; CAMPELLO, R. J. G. B. Design of OBF-TS Fuzzy Models Based on Multiple Clustering Validity Criteria. In: *19th IEEE ICTAI*. [S.l.: s.n.], 2007. v. 2, p. 336–339. ISSN 1082-3409.
- MEDEIROS, A. V.; AMARAL, W. C.; CAMPELLO, R. J. G. B. GA optimization of generalized OBF TS fuzzy models with global and local estimation approaches. In: *2006 IEEE Int. Conf. on Fuzzy Syst.* [S.l.: s.n.], 2006. p. 1835–1842.
- MORAES, C. de; CASTRUCCI, P. de L. *Engenharia de automação industrial*. [S.l.]: LTC, 2007. ISBN 9788521615323.
- MORARI, M.; GARCIA, C.; PRETT, D. M. Model predictive control: Theory and practice - A survey. *Automatica*, v. 25, n. 3, p. 335–348, 1989.
- MURRAY-SMITH, R. *A Local Model Network Approach to Nonlinear Modeling*. Tese (Doutorado) — University of Strathclyde - UK, set. 1994.
- NELLES, O. *Nonlinear System Identification*. 1st. ed. Berlin Heidelberg New York: Springer-Verlag, 2001. ISBN 3-540-67369-5.
- NINNESS, B.; HJALMARSSON, H.; GUSTAFSSON, F. The fundamental role of general orthonormal bases in system identification. *Automatic Control, IEEE Transactions on*, IEEE, v. 44, n. 7, p. 1384–1406, 1999.
- OGATA, K. *Engenharia de Controle Moderno*. 4th. ed. [S.l.]: Pearson Brasil, 2003. ISBN 8587918230.
- OLIVEIRA, G. H. C.; AMARAL, W.; LATAWIEC, K. Crhpc using volterra models and orthonormal basis functions: an application to cstr plants. In: *Control Applications, 2003. CCA 2003. Proceedings of 2003 IEEE Conference on*. [S.l.: s.n.], 2003. v. 1, p. 718–723.
- OLIVEIRA, G. H. C.; CAMPELLO, R. J. G. B.; AMARAL, W. C. Fuzzy models within orthonormal basis function framework. In: *Proc. 1999 IEEE Int. Fuzzy Syst. Conf.* [S.l.: s.n.], 1999. v. 2, p. 957–962 vol.2.
- OLIVEIRA, G. H. d. C. *Controle Preditivo para Processos com Incertezas Estruturadas baseado em Séries de Funções Ortonormais*. Tese (Doutorado) — UNICAMP - FEEC, set. 1997.
- OLIVEIRA, J. de; LEMOS, J. Long-range predictive adaptive fuzzy relational control. *Fuzzy Sets and Systems*, v. 70, p. 337 – 357, 1995. Modern Fuzzy Control.
- PHILLIPS, C.; HARBOR, R. *Feedback control systems*. [S.l.]: Prentice Hall, 1991. ISBN 9780133134469.
- PHILLIPS, L. C.; NAGLE, T. H. *Digital control system analysis and design*. 3rd. ed. New Jersey: Prentice Hall, Inc., 1995.
- QIN, S.; BADGWELL, T. An overview of nonlinear model predictive control applications. In: *Nonlinear Model Predictive Control*. [S.l.: s.n.], 2000, (Progress in Systems and Control Theory, v. 26). p. 369–392.

- QIN, S.; BADGWELL, T. A. A survey of industrial model predictive control technology. *Control Engineering Practice*, v. 11, n. 7, p. 733 – 764, 2003.
- QIN, S. J.; BADGWELL, T. A. An overview of industrial model predictive control technology. In: NEW YORK, NY: AMERICAN INSTITUTE OF CHEMICAL ENGINEERS, 1971-C2002. *AIChE Symposium Series*. [S.l.], 1997. v. 93, n. 316, p. 232–256.
- RICHALET, J. et al. Model predictive heuristic control: Applications to industrial processes. *Automatica*, v. 14, n. 5, p. 413–428, 1978. ISSN 0005-1098.
- ROSA, A. da; CAMPELLO, R. J. G. B.; AMARAL, W. C. Exact Search Directions for Optimization of Linear and Nonlinear Models Based on Generalized Orthonormal Functions. *Automatic Control, IEEE Transactions on*, v. 54, n. 12, p. 2757–2772, dec. 2009. ISSN 0018-9286.
- SANDRI, S.; CORREA, C. Lógica Nebulosa. *V Escola de Redes Neurais*, p. c073–c090, 1999.
- SANTOS, J. E. d. O. *Controle Preditivo Não-Linear para Sistemas de Hammerstein*. Tese (Doutorado) — Universidade Federal de Santa Catarina, abr. 2007.
- SCHOTT, K.; BEQUETTE, B. Multiple Model Adaptive Control. In: MURRAY-SMITH, R.; JOHANSEN, T. (Ed.). *Multiple Model Approaches to Modelling and Control*. [S.l.]: Taylor and Francis, 1997. cap. 11.
- SEKI, H. et al. Industrial application of a nonlinear model predictive control to polymerization reactors. *Control Engineering Practice*, v. 9, n. 8, p. 819 – 828, 2001. *Advanced Control of Chemical Processes*.
- SIMOES, M.; SHAW, I. *Controle e Modelagem Fuzzy*. [S.l.]: FAPESP, 2007. ISBN 978-85-212-0416-9.
- SOUZA, A.; PINHEIRO, C. A. M. *Introdução a Modelagem, Análise e Simulação de Sistemas Dinâmicos*. [S.l.]: Interciência, 2008. ISBN 9788571931886.
- TAKAGI, T.; SUGENO, M. Fuzzy Identification of systems and its application to modeling and control. *IEEE Transaction on Systems, Man and Cybernetics*, v. 15, n. 1, p. 116–132, 1985.
- VENDRAMIN, L. *Estudo e Desenvolvimento de Algoritmo para Agrupamento Fuzzy de Dados em cenário centralizado e distribuído*. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação - ICMC - USP, Agosto 2012.
- WAHLBERG, B. System identification using Kautz models. *Automatic Control, IEEE Transactions on, IEEE*, v. 39, n. 6, p. 1276–1282, 1994.
- WANG, L.-X.; MENDEL, J. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *Neural Networks, IEEE Transactions on*, v. 3, n. 5, p. 807–814, Sep 1992. ISSN 1045-9227.
- WANG, L.-X.; MENDEL, J. M. Fuzzy basis functions, universal approximation, and orthogonal least-square learning. *IEEE Transaction on Neural Network*, v. 43, n. 5, p. 807–814, 1992.

WIENER, N. *Extrapolation, Interpolation and Smoothing of Stationary Time Series*. [S.l.]: MIT Press, 1949.

YAGER, R.; FILEV, D. *Essentials of Fuzzy Modeling and Control*. [S.l.]: John Wiley & Sons, 1994. (A Wiley-Interscience publication). ISBN 9780471017615.

ZADEH, L. A. Fuzzy sets. *Information and Control*, v. 8, p. 338–353, 1965.

ZHAO, H. et al. A nonlinear industrial model predictive controller using integrated {PLS} and neural net state-space model. *Control Engineering Practice*, v. 9, n. 2, p. 125 – 133, 2001.

ZWICK, R.; CARLSTEIN, E.; BUDESCU, D. V. Measures of similarity among fuzzy concepts: A comparative analysis. *International Journal of Approximate Reasoning*, v. 1, n. 2, p. 221 – 242, 1987.

## ***APÊNDICE A – Método de Mínimos Quadrados***

O método dos mínimos quadrados foi originalmente proposto de forma independente por Gauss e Legendre. Gauss<sup>1</sup> durante suas observações astronômicas afirmou que:

*"O valor mais provável de grandezas desconhecidas é o que minimiza a soma dos quadrados da diferença entre o valor medido e o valor calculado, ponderando pelo grau de precisão da medida".*

Seja  $\hat{\theta}$  o valor estimado de  $\theta$  e  $\hat{y}$  a saída estimada do sistema, dada por:

$$\hat{y} = U\hat{\theta} \quad (\text{A.1})$$

Sejam os erros entre os vetores de parâmetros desconhecidos e estimados entre a saída do sistema e a saída do modelo definidos respectivamente por  $\tilde{\theta} = \theta - \hat{\theta}$  e  $\tilde{y} = y - \hat{y}$ . Logo, tem-se que:

$$\tilde{y} = U\tilde{\theta} \quad (\text{A.2})$$

A equação (A.2) é denominada equação do erro. O estimador é obtido minimizando:

$$J(\hat{\theta}) = w(1)\tilde{y}_1^2 + w(2)\tilde{y}_2^2 + \dots + w(N)\tilde{y}_N^2$$

sendo  $w(i)$  é a ponderação em cada componente do erro, que é a função da previsão da medida. Esta equação pode ser reescrita como segue:

$$J(\hat{\theta}) = \tilde{y}^T W \tilde{y}$$

sendo  $\tilde{y} = [\tilde{y}_1 \ \tilde{y}_2 \ \dots \ \tilde{y}_N]^T$  e  $W = \text{diag}(w(1) \ w(2) \ \dots \ w(N))$ . Deseja-se:

$$\hat{\theta}_M = \arg \min_{\hat{\theta}} J(\hat{\theta}) = \arg \min_{\hat{\theta}} \|y - U\hat{\theta}\|_w^2$$

---

<sup>1</sup>Johann Carl Friedrich Gauss (Braunschweig, 1777 - Göttingen, 1855) foi um famoso matemático, astrônomo e físico alemão. Era conhecido como o *príncipe dos matemáticos*. Muitos consideram Gauss o maior gênio da história da Matemática. Seu QI foi estimado em cerca de 240

sendo que a função  $J(\hat{\theta})$  pode ser reescrita como:

$$J(\hat{\theta}) = y^T W y - 2y^T W U \hat{\theta} + \hat{\theta}^T U^T W U \hat{\theta}$$

Sabe-se que  $\hat{\theta}_M$  deve ser tal que  $dJ/d\hat{\theta} = 0$ , o que implica:

$$\frac{dJ}{d\hat{\theta}} = -2[y^T W U]^T + 2U^T W U \hat{\theta} = 0$$

$$[U^T W U] \hat{\theta} = U^T W y \quad (\text{A.3})$$

que é denominada Equação Normal e produz o seguinte estimador:

$$\hat{\theta}_M = [U^T W U]^{-1} U^T W y \quad (\text{A.4})$$

A matriz  $[U^T W U]^{-1}$  é denominada matriz Pseudo-Inversa. O estimador (A.4) é denominado Estimador de Markov, Mínimos Quadrados Ponderados (*Weighted Least Squares* - *WLS*) ou ainda Mínimos Quadrados Generalizado. Quando  $W = \sigma^2 I_N$ , sendo  $\sigma$  um escalar e  $I_N$  é a matriz de identidade  $N \times N$ , obtém-se o estimador de Mínimos Quadrados convencional, descrito por:

$$\hat{\theta}_{MQ} = [U^T U]^{-1} U^T y \quad (\text{A.5})$$

Neste caso tem-se que  $\hat{\theta}_{MQ}$  é seguramente um ponto de mínimo pois a matriz Hessiana:

$$\frac{d^2 J}{d\hat{\theta}^2} = 2U^T W U$$

é definida positiva posto que  $W = \sigma^2 I_N$  é definida positiva. Logo,  $\hat{\theta}_{MQ}$  em (A.5) é uma solução ótima. Como é única, é uma solução ótima global. No caso geral do estimador em (A.4) a otimalidade requer apenas que a matriz de ponderação  $W$  seja definida positiva.

## *APÊNDICE B – Código para Implementação do Controlador Preditivo Fuzzy TS-GOBF*

Este apêndice tem como objetivo apresentar o código embarcado no CLP para implementação do Sistema de Controle Preditivo Multimodelos *Fuzzy* TS-GOBF. Os programas foram impressos diretamente da ferramenta de programação do CLP e anexados neste apêndice na sequência:

- Bloco de Função *KJ\_I* - *Function Block* utilizado para o cálculo da resposta livre de cada modelo local;
- Bloco de Função *F* - *Function Block* utilizado para o cálculo da resposta forçada de cada modelo local;
- Bloco de Função *deltaU* - *Function Block* utilizado para o cálculo da ação de controle local;
- Programa Local1 (INIT e CYCLIC) - Programa para a execução dos blocos de funções dos modelos locais;
- Programa *Fuzzy* (INIT e CYCLIC) - Programa utilizado para obtenção dos graus de ativação das regras *fuzzy* e para o cálculo da ação de controle global.
- Programa BFO (CYCLIC) - Programa para o cálculo da saída do modelo GOBF.



```
(
*****
* COPYRIGHT -
*****
* Library: Preditivo
* File: KJ_I.st
* Author: HenriquesJ
* Created: November 29, 2013
*****
* Implementation of library Preditivo
*****)
```

FUNCTION\_BLOCK KJ\_I

```

    I[1,1]      := 1;
    I[1,2]      := 0;
    I[2,1]      := 0;
    I[2,2]      := 1;

    IF (j=1) THEN
        E[1,1]   := A[1,1];
        E[1,2]   := A[1,2];
        E[2,1]   := A[2,1];
        E[2,2]   := A[2,2];

        A_J[1,1] := 0;
        A_J[1,2] := 0;
        A_J[2,1] := 0;
        A_J[2,2] := 0;

        D[1,1]   := 0;   D[1,2]   := 0;
        D[2,1]   := 0;   D[2,2]   := 0;

    END_IF
```

```

FOR x := 1 TO 2 BY 1 DO                                (* Número de Linhas da Matriz A *)
    FOR y := 1 TO 2 BY 1 DO                            (* Número de Colunas da Matriz B *)
        valor := 0;
        FOR z := 1 TO 2 BY 1 DO                       (* Número de Colunas da Matriz A *)
            IF(j<>1) THEN
                valor := valor + A[x,z] * E[z,y];
            END_IF
        END_FOR
    END_FOR
    IF(j<>1) THEN
        D[x,y] := valor ;
    ELSE
        D[x,y] := A[x,y];
    END_IF
END_FOR
END_FOR

FOR x := 1 TO 2 BY 1 DO
    FOR y := 1 TO 2 BY 1 DO
        A_J[x,y] := A_J[x,y]+ D[x,y];
        KJ_I[x,y] := A_J[x,y] - I[x,y];
    END_FOR
END_FOR
IF(j>1) THEN
    E[1,1] := D[1,1];    E[1,2] := D[1,2];
    E[2,1] := D[2,1];    E[2,2] := D[2,2];
END_IF
END_FUNCTION_BLOCK

```

```

(*****
* COPYRIGHT --
*****
* Library: Preditivo
* File: F.st
* Author: Pessoal
* Created: November 29, 2013
*****
* Implementation of library Preditivo
*****)

(* TODO: Add your comment here *)

FUNCTION_BLOCK F
    FOR flag1:=1 TO N BY 1 DO
        F_linha[flag1]      := (Y_L[flag1] - (SetPointCm *W[flag1]));

    END_FOR

    FOR linha := 1 TO Nu BY 1 DO
        valor2:= 0;

        FOR coluna := 1 TO N BY 1 DO
            valor2 := valor2 + ((2* G[linha,coluna])*F_linha[coluna]);

        END_FOR

        f[linha] := valor2;

    END_FOR

END_FUNCTION_BLOCK

```

```
(*****
* COPYRIGHT --
*****
* Library: Preditivo
* File: deltaU.st
* Author: HenriquesJ
* Created: November 29, 2013
*****
* Implementation of library Preditivo
*****)

FUNCTION_BLOCK deltaU
    FOR linha_a := 1 TO Nu BY 1 DO
        valor3:= 0;
        FOR coluna_a := 1 TO Nu BY 1 DO
            valor3 := valor3 + ((Q[linha_a,coluna_a])*f[coluna_a]);
        END_FOR
        DeltaU[linha_a] := valor3;
    END_FOR
END_FUNCTION_BLOCK
```

```
(*****  
* COPYRIG_1HT --  
*****  
* ProG_1ram: Matriz3  
* File: Matriz3Init.st  
* Author: HenriquesJ  
* Created: November 19, 2012  
*****  
* Implementation of proG_1ram Matriz3  
*****)
```

```
PROGRAM _INIT
```

```
(* Matriz A *)
```

```
A[1,1] := 0.9855;
```

```
A[1,2] := 0;
```

```
A[2,1] := 0.0288;
```

```
A[2,2] := 0.9855;
```

```
(* Vetor B*)
```

```
B[1] := 0.1698 ;
```

```
B[2] := -0.1673;
```

```
(* Matriz Identidade 2 x 2 *)
```

```
I[1,1] := 1;    I[1,2] := 0;
```

```
I[2,1] := 0;    I[2,2] := 1;
```

```
(* Vetor C *)
```

```
c_1[1] := 2.5971;    c_1[2] := -5.4999;
```

```
c_2[1] := 3.9225;    c_2[2] := -1.5687;
```

```
c_3[1] := 4.7437;    c_3[2] := 0.9813;
```

```
(* Set Point *)
```

```
W[1] := 1;    W[2] := 1;    W[3] := 1;    W[4] := 1;    W[5] := 1;    W[6] := 1;
```

```
W[7] := 1;    W[8] := 1;    W[9] := 1;    W[10] := 1;    W[11] := 1;    W[12] := 1;
```

```
W[13] := 1;    W[14] := 1;    W[15] := 1;    W[16] := 1;    W[17] := 1;    W[18] := 1;
```

```
W[19] := 1;    W[20] := 1;    W[21] := 1;    W[22] := 1;    W[23] := 1;    W[24] := 1;
```

```
W[25] := 1;    W[26] := 1;    W[27] := 1;    W[28] := 1;    W[29] := 1;    W[30] := 1;
```

```
(* Matriz Q (invertida) -> G_1'*G_1 + alfa*eye(2)*)
```

```
Q_1[1,1]:=-0.0011864;    Q_1[1,2]:=0.0011993;
```

```
Q_1[2,1]:=0.0011993;    Q_1[2,2]:=-0.0012757;
```

```
Q_2[1,1]:=-0.0012073;    Q_2[1,2]:=0.0012064;
```

```
Q_2[2,1]:=0.0012064;    Q_2[2,2]:=-0.0013089;
```

```
Q_3[1,1]:=-0.0012223;    Q_3[1,2]:=0.0012012;
```

```
Q_3[2,1]:=0.0012012;    Q_3[2,2]:=-0.0013359;
```

```
j:= 1;
```

```
d := 2;
```

```
HabilitaControle      :=FALSE;
```

```
END_PROGRAM
```

```
(*****
* COPYRIGHT --
*****
* Program:   Calculo da Resposta Livre, Forçada LOCAL
* File: Matriz3Cyclic.st
* Author: HenriquesJ
* Created: November 19, 2012
*****
* Implementation of program Matriz3
*****)

PROGRAM _CYCLIC

IF(HabilitaControle) THEN

    Counter := Counter + 1;

END_IF

IF HabilitaControle AND (Counter = NumControle) THEN

    Y[d] := NivelGlobal ;

    IF ( d=2) THEN

        SinalControle      := 0;

    END_IF

    j := 1;
```

(\* Horizonte de Predição - Alterar valor do vetor YL \*)

WHILE (j<(N+1)) DO

KJ\_I\_0(j := j, A := A); (\* Chamada do Bloco para Cálculo de KJ - I\*)

FOR x:=1 TO 2 BY 1 DO (\* Saída do Bloco KJ-I \*)

FOR y:=1 TO 2 BY 1 DO

KJ\_W[x,y] := KJ\_I\_0.KJ\_I[x,y];

END\_FOR

END\_FOR

(\* Cálculo da Resposta Livre \*)

(\* Saída Livre - Modelo Local 1\*)

Y\_L\_X\_1 := c\_1[1]\*(KJ\_W[1,1]) + c\_1[1]\*(KJ\_W[1,2]);

Y\_L\_X\_1\_1 := c\_1[2]\*(KJ\_W[2,1]) + c\_1[2]\*(KJ\_W[2,2]);

Y\_L\_1[j] := (Y\_L\_X\_1)\*deltaL[1] + (Y\_L\_X\_1\_1)\*deltaL[2] + Y[d-1] ;  
(\* Saída Livre - Modelo Local 2\*)

Y\_L\_X\_2 := c\_2[1]\*(KJ\_W[1,1]) + c\_2[1]\*(KJ\_W[1,2]);

Y\_L\_X\_2\_1 := c\_2[2]\*(KJ\_W[2,1]) + c\_2[2]\*(KJ\_W[2,2]);

Y\_L\_2[j] := (Y\_L\_X\_2)\*deltaL[1] + (Y\_L\_X\_2\_1)\*deltaL[2] + Y[d-1] ;

(\* Saída Livre - Modelo Local 3\*)

Y\_L\_X\_3 := c\_3[1]\*(KJ\_W[1,1]) + c\_3[1]\*(KJ\_W[1,2]);

Y\_L\_X\_3\_1 := c\_3[2]\*(KJ\_W[2,1]) + c\_3[2]\*(KJ\_W[2,2]);

Y\_L\_3[j] := (Y\_L\_X\_3)\*deltaL[1] + (Y\_L\_X\_3\_1)\*deltaL[2] + Y[d-1] ;



```
        j:= j +1; (* Incremento Horizonte de Predição *)  
  
END_WHILE  
  
        (* Aplicação Referências Nível*)  
IF (d<150) THEN  
  
        SetPointCm := SP1;  
  
ELSIF ((d>150) AND (d<350) )THEN  
  
        SetPointCm := SP2;  
  
ELSIF ((d>350) AND (d<550)) THEN  
  
        SetPointCm := SP3;  
  
ELSIF ((d>550) AND (d<750)) THEN  
  
        SetPointCm := SP4;  
  
ELSIF ((d>750) AND (d<950)) THEN  
  
        SetPointCm := SP5;  
  
ELSIF ((d>950) AND (d<1150)) THEN  
  
        SetPointCm := SP6;  
  
ELSIF ((d>1150) AND (d<1350)) THEN  
  
        SetPointCm := SP7;  
  
ELSIF ((d>1350) AND (d<1550)) THEN  
  
        SetPointCm := SP8;  
  
ELSIF ((d>1550) AND (d<1750)) THEN  
  
        SetPointCm := SP9;  
  
ELSIF ((d>1750) AND (d<1950)) THEN  
  
        SetPointCm := SP10;  
  
END_IF
```

(\* Cálculo da Resposta Forçada - Modelo Local 1\*)

F\_1(Y\_L := Y\_L\_1, N := N , SetPointCm := SetPointCm, \

G:= G\_1, W:= W, Nu := Nu );

f\_1[1] := F\_1.f[1];

f\_1[2] := F\_1.f[2];

f\_1[3] := F\_1.f[3];

f\_1[4] := F\_1.f[4];

f\_1[5] := F\_1.f[5];

(\* Cálculo da Resposta Forçada - Modelo Local 2\*)

F\_2(Y\_L := Y\_L\_2, N := N , SetPointCm := SetPointCm, \

G := G\_2, W:= W, Nu := Nu );

f\_2[1] := F\_2.f[1];    f\_2[2] := F\_2.f[2];

f\_2[3] := F\_2.f[3];    f\_2[4] := F\_2.f[4];

f\_2[5] := F\_2.f[5];

(\* Cálculo da Resposta Forçada - Modelo Local 3\*)

F\_3(Y\_L := Y\_L\_3, N := N , SetPointCm := SetPointCm, \

G := G\_3, W:= W, Nu := Nu );

f\_3[1] := F\_3.f[1];    f\_3[2] := F\_3.f[2];

f\_3[3] := F\_3.f[3];    f\_3[4] := F\_3.f[4];

f\_3[5] := F\_3.f[5];

(\* Calculo do Delta U \*)

deltaU\_1(Q := Q\_1, f := f\_1, Nu := Nu);

U\_Local1:= deltaU\_1.DeltaU[1];(\* Ação de Controle Modelo Local 1 \*)

deltaU\_2(Q := Q\_2, f := f\_2, Nu := Nu);

U\_Local2:= deltaU\_2.DeltaU[1];(\* Ação de Controle Modelo Local 2 \*)

deltaU\_3(Q := Q\_3, f := f\_3, Nu := Nu);

U\_Local3:= deltaU\_3.DeltaU[1];(\* Ação de Controle Modelo Local 3 \*)

END\_IF

END\_PROGRAM

```
(*****
* COPYRIGHT --
*****
* Program: Fuzzy
* File: FuzzyInit.st
* Author: HenriquesJ
* Created: May 02, 2013
*****
* Implementation of program Fuzzy
*****)
```

PROGRAM\_INIT

```
V1_R1      :=  -973.1173;
V2_R1      :=  -973.1174;
V3_R1      :=  -15.4772;
V4_R1      :=   62.0369;

V1_R2      :=   26.2372;
V2_R2      :=   57.8024;
V3_R2      :=   73.0029;
V4_R2      :=   79.0810;

V1_R3      :=   62.6955;
V2_R3      :=   97.5390;
V3_R3      :=  231.4390;
V4_R3      :=  231.4391;
```

SinalControle := 0;

SinalControleGlobal := 0;

SinalControleIncr := 0;

END\_PROGRAM

```
(*****  
  
* COPYRIGHT --  
  
*****  
  
* Program: Fuzzy  
  
* File: FuzzyCyclic.st  
  
* Author: HenriquesJ  
  
* Created: May 02, 2013  
  
*****  
  
* Implementation of program Fuzzy *)  
  
PROGRAM _CYCLIC  
  
IF HabilitaControle AND (Counter = NumControle) THEN  
  
    IF ( Filtro_L1 > V1_R1) AND (Filtro_L1 < V2_R1) THEN  
  
        Alfa_1 := (Filtro_L1 - V1_R1);  
  
        Beta_1 := (V2_R1 - V1_R1);  
  
        Mi_1 := Alfa_1 / Beta_1;  
  
    ELSIF (Filtro_L1 > V2_R1) AND (Filtro_L1 < V3_R1) THEN  
  
        Mi_1 := 1;  
  
    ELSIF (Filtro_L1 > V3_R1) AND (Filtro_L1 < V4_R1) THEN  
  
        Gama_1 := (V4_R1 - Filtro_L1);  
  
        Delta_1 := (V4_R1 - V3_R1);  
  
        Mi_1 := Gama_1 / Delta_1;  
  
    ELSE  
  
        Mi_1 := 0;  
  
    END_IF
```

```
(***** REGRA 2 *****)
```

```
IF ( Filtro_L1 > V1_R2) AND (Filtro_L1 < V2_R2) THEN
```

```
    Alfa_2 := (Filtro_L1 - V1_R2);
```

```
    Beta_2 := (V2_R2 - V1_R2);
```

```
    Mi_2 := Alfa_2 / Beta_2;
```

```
ELSIF (Filtro_L1 > V2_R2) AND (Filtro_L1 < V3_R2) THEN
```

```
    Mi_2 := 1;
```

```
ELSIF (Filtro_L1 > V3_R2) AND (Filtro_L1 < V4_R2) THEN
```

```
    Gama_2 := (V4_R2 - Filtro_L1);
```

```
    Delta_2 := (V4_R2 - V3_R2);
```

```
    Mi_2 := Gama_2 / Delta_2;
```

```
ELSE
```

```
    Mi_2 := 0;
```

```
END_IF
```

```
(***** REGRA 3 *****)
```

```
IF ( Filtro_L1 > V1_R3) AND (Filtro_L1 < V2_R3) THEN
```

```
    Alfa_3 := (Filtro_L1 - V1_R3);
```

```
    Beta_3 := (V2_R3 - V1_R3);
```

```
    Mi_3 := Alfa_3 / Beta_3;
```

```
ELSIF (Filtro_L1 > V2_R3) AND (Filtro_L1 < V3_R3) THEN
```

```
    Mi_3:= 1;
```

```

ELSIF (Filtro_L1 > V3_R3) AND (Filtro_L1 < V4_R3) THEN

    Gama_3      :=      (V4_R3 - Filtro_L1) ;

    Delta_3     := (V4_R3 - V3_R3);

    Mi_3 :=      ( Gama_3      / Delta_3 );

ELSE

    Mi_3 := 0;

END_IF

(***** Saida Global *****)

NumeradorControle := (Mi_1 * U_Local1 + Mi_2 * \
                    U_Local2 + Mi_3 * U_Local3 );

DenominadorControle:= (Mi_1 + Mi_2 + Mi_3 );

SinalControleIncr:= NumeradorControle / DenominadorControle;

SinalControle := SinalControleIncr + SinalControleGlobal;

END_IF

IF (SinalControle > 10) THEN

    SinalControleGlobal      := 10;

ELSIF (SinalControle < 0 ) THEN

    SinalControleGlobal      := 0;

ELSE

    SinalControleGlobal      := SinalControle;

END_IF

SaidaAnalogica := REAL_TO_INT (SinalControleGlobal * 3276.7);

END_PROGRAM

```



```
(*****  
  
* COPYRIGHT --  
  
*****  
  
* Program: Matriz3  
* File: Matriz3Cyclic.st  
* Author: HenriquesJ  
* Created: November 19, 2012  
  
*****  
  
* Implementation of program BFO  
  
*****)  
  
PROGRAM _CYCLIC  
  
IF HabilitaControle AND (Counter = NumControle) THEN  
  
    Filtro_L1_Old[1]    := L[1];  
    Filtro_L1_Old[2]    := L[2];  
  
    FOR linha_b := 1 TO 2 BY 1 DO  
  
        valor4:= 0;  
  
        FOR coluna_b := 1 TO 2 BY 1 DO  
  
            valor4 := valor4 + A[linha_b,coluna_b]*L_old[coluna_b];  
  
        END_FOR  
  
        L_a[linha_b] := valor4;  
  
    END_FOR  
  
    L_b[1] := B[1]*SinalControleGlobal;  
    L_b[2] := B[2]*SinalControleGlobal;
```

```
L[1] := L_a[1] + L_b[1];
L[2] := L_a[2] + L_b[2];

Filtro_L1 := L[1];          (* Saída do Filtro *)

L_old[1] := L[1];
L_old[2] := L[2];

deltaL[1] := L[1] - Filtro_L1_Old[1];
deltaL[2] := L[2] - Filtro_L1_Old[2];

(*SIMULACAO*)

Y_1 := (c_1[1]*L[1]) + (c_1[2]*L[2]);
Y_2 := (c_2[1]*L[1]) + (c_2[2]*L[2]);
Y_3 := (c_3[1]*L[1]) + (c_3[2]*L[2]);
Num_Y := (Mi_1*Y_1) + (Mi_2*Y_2) + (Mi_3*Y_3);
Den_Y := Mi_1 + Mi_2 + Mi_3;
Y_Simulacao := (Num_Y / Den_Y);

IF (d=Numero_Amostras) THEN
    HabilitaControle := FALSE;
END_IF

d := d + 1;

Counter := 0;

END_IF

END_PROGRAM
```