

Fadul Ferrari Rodor

Modelagem de Sistemas Dinâmicos Não Lineares via RBF-GOBF

Itajubá - MG
Novembro de 2017

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

Fadul Ferrari Rodor

Modelagem de Sistemas Dinâmicos Não
Lineares via RBF-GOBF

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

Área de Concentração: Automação e Sistemas Elétri-
cos Industriais

Orientadores: Prof. Dr. Carlos Alberto Murari Pinheiro
Prof. Dr. Jeremias Barbosa Machado

Novembro de 2017

Itajubá

Ao meu amado avô, Ignácio (in memoriam).

Agradecimentos

Agradeço primeiramente a minha família, pelo constante apoio nessa trajetória.

Aos amigos da época de república André, Gabriel, Victor, Brian, Igor e Beto, pela paciência nos meus dias ruins.

Aos amigos Diogo, Caio, Filipe, Fernando, Júlio, Kalife, Gabriela, Rodrigo e Paulo pela companhia nos bons e maus momentos.

Ao professor Pinheiro pelo apoio e paciência nos quase dez anos em que temos trabalhado juntos. Ao professor Jeremias, sempre disposto a indicar um caminho nessa jornada.

*“Life’s but a walking shadow, a poor player
That struts and frets his hour upon the stage
And then is heard no more. It is a tale
Told by an idiot, full of sound and fury.”
(William Shakespeare)*

Resumo

Trata-se neste trabalho trata da modelagem e identificação de sistemas dinâmicos não lineares estáveis representáveis por modelos de Wiener por um estrutura formada por bases de funções ortonormais generalizadas (*Generalized Orthonormal Basis Functions* - GOBF) com funções internas e redes neurais com funções de base radial (*Radial Basis Functions* - RBF). Os modelos GOBF com funções internas são capazes de representar dinâmicas lineares intrincadas com uma parametrização que se vale apenas de valores reais, sejam os polos do sistema a ser representado complexos e/ou reais. Com informações de entrada e saída do sistema a ser identificado é possível obter um modelo GOBF-RBF inicial. Os *clusters* que determinam os parâmetros iniciais das RBFs (centros das funções gaussianas e larguras ou *spreads*) são obtidos pelo método *fuzzy C-means*, o qual é inicializado com um número de centros pré-determinado, obtido pela técnica *subtractive clustering*, garantindo *clusters* com volume e densidade apropriados. São propostas duas técnicas para o ajuste dos parâmetros da estrutura. A primeira delas se baseia em um método de otimização não linear e os gradientes exatos da estrutura. Apresenta-se um procedimento para a obtenção dos cálculos analíticos dos gradientes de saída do modelo GOBF-RBF em relação a seus parâmetros (polos da base ortonormal, centros, larguras e pesos de saída da rede RBF). A segunda proposta se vale de um método metaheurístico chamado otimização por enxame de partículas com comportamento quântico. As metodologias são validadas com suas aplicações em três diferentes sistemas não lineares associados a modelos de processos práticos.

Palavras-chaves: Base de Funções Ortonormais Generalizadas. Sistemas Dinâmicos Não-lineares. Identificação de Sistemas. Redes Neurais com Funções de Base Radial. Otimização por enxame de partículas.

Abstract

This work aims the modeling and identification of stable nonlinear dynamic systems which can be represented by Wiener models in a structure composed by a ladder-structured Generalized Orthonormal Basis Functions (GOBF) and Radial Basis Functions Networks (RBF networks). Ladder-structured GOBF models are capable of represent complex dynamic systems through a parametrization in which only real values are used, been the poles of that system real and/or complex. With input and output data of a given system to be identified is possible to obtain a initial GOBF-RBF model, using clustering methods. The clusters that define the initial RBF systems are obtained through a Fuzzy C-Means clustering in combination with the subtractive clustering technique, which guarantee that the clusters will have suitable volume and density. Two techniques for the parameters adjustment are proposed. The first one relies on a nonlinear optimization method using the analytical gradients of the output structure concerning its parameters (GOBF poles, centers, standard deviations or spreads, and output weights). The second one uses a meta-heuristic algorithm named particle swarm optimization. The methodologies are applied to three different nonlinear dynamic systems.

Keywords: Generalized Orthonormal Basis Functions. Nonlinear Dynamic Systems. System Identification. Radial Basis Function Neural Network, Particle Swarm Optimization.

Lista de ilustrações

Figura 1 – Representação em blocos das Funções de Laguerre.	23
Figura 2 – Representação em blocos das Funções de Kautz.	25
Figura 3 – Representação em blocos de um modelo GOBF.	27
Figura 4 – Representação em blocos da estrutura interna parametrizada por γ	29
Figura 5 – Representação em blocos das funções da base de uma GOBF.	30
Figura 6 – Estrutura de um modelo de Wiener.	34
Figura 7 – Estrutura típica de uma rede neural RBF.	37
Figura 8 – Fluxograma do algoritmo <i>subtractive clustering</i>	43
Figura 9 – Modelo GOBF-RBF.	44
Figura 10 – Fluxograma do processo de otimização não linear.	49
Figura 11 – Sistema de Nível.	58
Figura 12 – Dados de identificação do sistema de nível.	59
Figura 13 – Dados de validação do sistema de nível.	59
Figura 14 – Estrutura do modelo GOBF-RBF para o sistema de nível líquido.	60
Figura 15 – Resposta do modelo selecionado pelo EQM para os dados de validação do sistema de nível utilizando a otimização não linear.	61
Figura 16 – Resposta do modelo selecionado pelo BIC para os dados de validação do sistema de nível utilizando a otimização não linear.	62
Figura 17 – Resposta do modelo com 1 polo e 8 funções na base otimizado pelo QPSO para os dados de validação do sistema de nível.	64
Figura 18 – Comportamento do algoritmo QPSO para o modelo com 1 polo e 8 funções na base para o sistema de nível.	64
Figura 19 – Resposta do modelo com 1 polo e 3 funções na base otimizado pelo QPSO para os dados de validação do sistema de nível.	65
Figura 20 – Comportamento do algoritmo QPSO para o modelo com 1 polo e 3 funções na base para o sistema de nível.	65
Figura 21 – Dados de identificação do sistema NARX.	66
Figura 22 – Dados de validação do sistema NARX.	67
Figura 23 – Resposta do modelo selecionado pelo BIC para os dados de validação do sistema NARX, utilizando a otimização não linear.	68
Figura 24 – Resposta do modelo selecionado pelo BIC para os dados de validação do sistema NARX, utilizando a otimização não linear.	69
Figura 25 – Resposta do modelo com 2 polos e 4 funções na base para os dados de validação do sistema de nível.	70
Figura 26 – Comportamento do algoritmo QPSO para o modelo com 2 polos e 4 funções na base para o sistema de nível.	71

Figura 27 – Resposta do modelo com 3 polos e 1 função na base para os dados de validação do sistema NARX.	71
Figura 28 – Comportamento do algoritmo QPSO para o modelo com 3 polos e 1 função na base para o sistema NARX.	72
Figura 29 – Sistema de Levitação Magnética.	73
Figura 30 – Sistema de levitação magnética real a ser identificado (QUANSER, 2012).	74
Figura 31 – Dados de identificação do sistema de levitação magnética.	75
Figura 32 – Dados de identificação do sistema de levitação magnética.	75
Figura 33 – Resposta do modelo selecionado pelo EQM para os dados de validação do sistema de levitação magnética, utilizando a otimização não linear.	76
Figura 34 – Resposta do modelo selecionado pelo BIC para os dados de validação do sistema de levitação magnética, utilizando a otimização não linear.	77
Figura 35 – Resposta do modelo com 3 polos e 2 funções na base para os dados de validação do sistema de levitação magnética.	78
Figura 36 – Comportamento do algoritmo QPSO para o modelo com 3 polos e 2 funções na base para o sistema de levitação magnética.	79
Figura 37 – Resposta do modelo com 3 polos e 3 funções na base otimizado pelo QPSO para os dados de validação do sistema de levitação magnética.	79
Figura 38 – Comportamento do algoritmo QPSO para o modelo com 3 polos e 3 funções na base para o sistema de levitação magnética.	80
Figura 39 – Histograma das simulações de Monte Carlo para o sistema de nível.	81
Figura 40 – Histograma das simulações de Monte Carlo para o sistema de levitação magnética.	82

Lista de tabelas

Tabela 1 – Modelo com melhor EQM para o sistema de nível líquido.	60
Tabela 2 – Informações do modelo GOBF-RBF para o sistema de nível líquido. . .	60
Tabela 3 – Comparação da resposta do modelo do nível líquido com outras estruturas.	61
Tabela 4 – Modelos GOBF-RBF para o sistema de nível.	62
Tabela 5 – Parâmetros do QPSO para o sistema de nível.	63
Tabela 6 – Modelos GOBF-RBF para o sistema de nível.	63
Tabela 7 – Modelos GOBF-RBF para o sistema NARX.	67
Tabela 8 – Modelo com melhor EQM para o modelo racional NARX.	67
Tabela 9 – Comparação da resposta do modelo NARX com outras estruturas. . .	68
Tabela 10 – Parâmetros do QPSO para o sistema NARX.	69
Tabela 11 – Modelos GOBF-RBF para o sistema de nível.	70
Tabela 12 – Modelos GOBF-RBF para o sistema de levitação magnética.	76
Tabela 13 – Modelo com melhor EQM para o sistema de levitação magnética. . . .	76
Tabela 14 – Parâmetros do QPSO para o sistema de levitação magnética.	77
Tabela 15 – Modelos GOBF-RBF para o sistema de levitação magnética.	78
Tabela 16 – Resultados da simulação de Monte Carlo para o sistema de nível usando a otimização não linear.	81
Tabela 17 – Resultados da simulação de Monte Carlo para o sistema de levitação magnética usando a otimização não linear.	81

Lista de abreviaturas e siglas

EQM	- Erro Quadrático Médio
BFO	- Base de Funções Ortonormais
GOBF	- <i>Generalized Orthonormal Basis Function</i>
NF	- Número de Funções
NP	- Número de Polos
NC	- Número de <i>Clusters</i> na Rede de Base Radial
TP	- Total de Parâmetros do Modelo
FIR	- <i>Finite Impulse Response</i>
NFIR	- <i>Nonlinear Finite Impulse Response</i>
IIR	- <i>Infinite Impulse Response</i>
RBF	- <i>Radial Basis Function</i>
NARMAX	- <i>Nonlinear Autoregressive Moving Average with exogenous inputs</i>
NARX	- <i>Nonlinear Autoregressive with exogenous inputs</i>
SISO	- <i>Single Input Single Output</i>
MLP	- <i>Multi Layer Perceptron</i>
PSO	- <i>Particle Swarm Optimization</i>
QPSO	- <i>Quantum-behaved Particle Swarm Optimization</i>
AIC	- <i>Akaike Information Criterion</i>
BIC	- <i>Bayesian Information Criterion</i>
FPE	- <i>Final Prediction Error</i>

Lista de símbolos

$u(k)$	- Entrada de excitação dos sistema
$\phi(k)$	- Resposta de uma função ortonormal a uma entrada $u(k)$
$\Phi(k)$	- Vetor com as saídas das funções ortonormais
p	- Polo da função de Laguerre
c e b	- Parâmetros da função de Kautz
θ	- Parâmetros de Otimização em um modelo BFO
γ	- Parâmetros da GOBF com funções internas de estrutura <i>ladder</i>
Υ	- Conjunto de parâmetros γ
μ_{ij}	- Valor de ativação de uma regra <i>Fuzzy</i>
$\sigma_{i,j}$	- Desvio-padrão ou <i>spreads</i> da i -ésima função de base radial para a j -ésima amostra
Σ_i	- Matriz de covariâncias da i -ésima função de base radial
$v_{i,j}$	- Centros da i -ésima função de base radial da j -ésima amostra
V	- Matriz formada pelos vetores dos centros das funções de base radial
y	- Saída medida ou simulada
\hat{y}	- Saída estimada
w_i	- Peso da camada de saída da i -ésima função de base radial
W	- Vetor de pesos de saída das funções de base radial

Sumário

1	INTRODUÇÃO	15
1.1	Modelagem de Sistemas Lineares por Bases de Funções Ortonormais	15
1.2	Modelagem de Sistemas Dinâmicos Não Lineares	16
1.3	Organização do Trabalho	17
2	MODELAGEM DE SISTEMAS DINÂMICOS LINEARES POR BASES DE FUNÇÕES ORTONORMAIS	19
2.1	Introdução	19
2.2	Bases de Funções Ortonormais	20
2.3	Base de Funções Ortonormais Generalizadas - GOBFs	26
2.3.1	Funções Internas	28
2.4	Ajuste de parâmetros das GOBF	33
3	MODELAGEM DE SISTEMAS DINÂMICOS NÃO LINEARES POR BASES DE FUNÇÕES ORTONORMAIS	34
3.1	Introdução	34
3.2	Redes Neurais com Funções de Base Radial	35
3.2.1	Os neurônios das redes de base radial	35
3.2.2	Estrutura da Rede Neural RBF	37
3.2.3	Treinamento das redes RBF	38
3.2.4	Definição dos parâmetros iniciais da Rede Neural de Base Radial	39
3.2.5	<i>Subtractive Clustering</i>	41
3.3	Modelos GOBF-RBF	44
3.4	Otimização Não Linear dos Modelos GOBF-RBF	45
3.4.1	Ajuste dos parâmetros do modelo GOBF-RBF final	46
3.4.2	Algoritmo de otimização não linear dos modelos GOBF-RBF	48
3.5	Otimização dos Modelos GOBF-RBF por Enxame de Partículas	49
3.5.1	Algoritmo PSO	50
3.5.2	Algoritmo QPSO	52
3.5.3	Algoritmo de otimização não linear dos modelos GOBF-RBF	55
4	RESULTADOS	57
4.1	Introdução	57
4.2	Exemplos de aplicação	57
4.2.1	Modelo de um sistema de nível	58
4.2.1.1	Otimização não linear	59

4.2.1.2	Otimização pelo QPSO	63
4.2.2	Modelo racional NARX	66
4.2.2.1	Otimização não linear	67
4.2.2.2	Otimização pelo QPSO	69
4.2.3	Modelo de um sistema de levitação magnética	72
4.2.3.1	Otimização não linear	75
4.2.3.2	Otimização pelo QPSO	77
4.3	Simulação de Monte Carlo	80
5	COMENTÁRIOS FINAIS, CONCLUSÕES E CONTRIBUIÇÕES . .	83
	APÊNDICE A – CÁLCULOS ANALÍTICOS DOS GRADIENTES PARA AS FUNÇÕES GENERALIZADAS	85
	APÊNDICE B – SELEÇÃO DE MODELOS - CRITÉRIOS DE IN- FORMAÇÃO	91
	REFERÊNCIAS	92

1 Introdução

A base da ciência é, em sua essência, a observação e a medição. Baseados em observações, os cientistas formulam teorias sobre os problemas estudados. Com essas teorias, desenvolvem experimentos para enfim comprová-las, modificá-las ou, em alguns casos, refutá-las. De forma semelhante pode-se exaltar a importância do desenvolvimento de modelos, pois como importante ramo da ciência, são o resultado da observação, medição e experimentação com a finalidade de obter-se uma representação semântica ou matemática que descreva um comportamento ou fenômeno real. Assim, um modelo representa os aspectos essenciais de um sistema existente (ou a ser construído), o qual agrega conhecimentos a cerca desse sistema de forma utilizável (EYKHOFF, 1974).

Em engenharia, a obtenção de modelos que representem processos com determinada exatidão são necessários na análise e projeto desses. Técnicas de projeto de controladores, previsão de séries temporais e otimização de sistemas são baseadas em modelos de processos, e os resultados dessas soluções está intimamente ligada a uma boa representação de tais sistemas. Os recursos computacionais atualmente disponíveis permitem a análise de sistemas complexos, possibilitando a elaboração de modelos matemáticos eficientes e técnicas de controle que atendam a especificações mais rígidas, garantindo melhor desempenho dos processos associados.

1.1 Modelagem de Sistemas Lineares por Bases de Funções Ortonormais

Aborda-se neste trabalho a modelagem de sistemas dinâmicos por funções ortonormais que formem uma base completa no espaço Lebesgue $L_2[0, \infty)$. Bases de funções ortonormais (*Orthonormal Basis Functions* - OBF) têm sido utilizadas com sucesso para modelar e controlar sistemas dinâmicos (WAHLBERG, 1991; DUMONT; FU, 1993; WAHLBERG, 1994; HOF; HEUBERGER; BOKOR, 1995; CAMPELLO; AMARAL, 2002; MACHADO, 2007; MACHADO, 2011; MACHADO; CAMPELLO; AMARAL, 2013). Esses modelos apresentam características vantajosas para a representação de sistemas dinâmicos, como por exemplo: não é necessário conhecer a priori a estrutura exata do vetor de regressão das informações numéricas do sistema em questão; aumenta-se a capacidade de representação do modelo simplesmente aumentando-se a ordem do mesmo; tolerância a dinâmicas não modeladas; ausência de realimentação de saída, evitando a propagação de erros de estimação (NELLES, 2001); garantia de que a representação de um sistema estável também o será (HEUBERGER; HOF; WAHLBERG, 2005).

Uma base de funções ortonormais pode ser definida de maneira arbitrária através da seleção dos seus polos, devido a sua completude. Porém, caso esses sejam selecionados próximos aos polos do sistema a ser modelado, obtém-se uma modelagem com menos funções na base (HEUBERGER; HOF; WAHLBERG, 2005). As bases mais comuns são as de Laguerre, de Kautz e Generalizada.

As bases de Laguerre são mais apropriadas para a modelagem de sistemas com polos reais ou com parte complexa de valor reduzido, as de Kautz, devido a sua construção baseada em polos complexos conjugados, necessita de um número menor de funções para representar sistemas oscilatórios. Quando a dinâmica do sistema em questão é muito intrincada, com múltiplos modos dominantes, opta-se pelas bases de funções ortonormais generalizadas (*Generalized Orthonormal Basis Functions* - GOBF), uma vez que estas podem representar vários polos distintos. Essa última é obtida a partir da conexão de n filtros *all-pass* de n_b -ésima ordem (cada filtro possui n_b polos, distintos ou não, reais ou complexos). Uma estrutura de especial interesse neste trabalho são as chamadas GOBF com funções internas de estrutura *ladder*, sendo essas parametrizadas apenas por parâmetros reais, independente da natureza dos polos das funções internas.

1.2 Modelagem de Sistemas Dinâmicos Não Lineares

Há alguns anos, a modelagem de sistemas dinâmicos não lineares se caracterizava por um conjunto de soluções específicas e restritas para os problemas em estudo. O advento das tecnologias com redes neurais, modelos *fuzzy* e técnicas de otimização sistematizadas para propostas de estruturas mais modernas, nos permite tratar o problema de identificação de sistemas não lineares de uma maneira muito mais abrangente, mas sempre tendo em vista que a única característica comum entre essa classe é sua unicidade (NELLES, 2001).

Bases de funções ortonormais podem ser utilizadas para representar sistemas dinâmicos estáveis lineares ou não (MACHADO, 2011; MACHADO, 2007; CAMPELLO; FAVIER; AMARAL, 2003; ROSA, 2009), assemelhando-se a estrutura de regressores dos modelos NFIR (*Nonlinear Finite Impulse Response*), contudo sem as desvantagem do enorme número de regressores, o que reflete em um número menor de parâmetros a serem estimados. São ainda muito mais robustas a dinâmicas não modeladas e a estimações pouco precisas do atraso de transporte do sistema (AGUIRRE, 2007a). Tendo essas características em vista, adiciona-se à estrutura das GOBFs os modelos neurais baseados em funções de base radial, a fim de obter-se um modelo GOBF-RBF para a representação de sistemas dinâmicos não lineares.

O conceito de redes com funções de base radial (*radial basis function* - RBF) foi criado por Broomhead e Lowe (1988). Estudos subsequentes foram realizados sobre

sua capacidade de interpolação e aproximação (FREEMAN; SAAD, 1995; LOWE, 1989). Esse tipo de estrutura, diferente de uma rede MLP (*Multilayer Perceptron*), se vale de uma estrutura radial, o que a concede uma capacidade de interpretação melhor frente as redes com múltiplas camadas, bem como novos e mais rápidos métodos de treinamento (HAYKIN, 1999; NELLES, 2001).

O procedimento básico proposto nesta tese para modelagem de sistemas não lineares está exemplificado a seguir: para a obtenção dos modelos GOBF-RBF primeiramente escolhe-se o número e os valores dos parâmetros γ , que definem os polos do modelo GOBF, de maneira aleatória ou com base em algum conhecimento prévio do modelo a ser identificado. A partir dos dados de saída dos filtros das GOBF anteriormente especificadas e a resposta desejada do modelo, aplica-se a técnica de agrupamento *Fuzzy C-Means* (BEZDEK, 1980) em conjunto com o *Subtractive Clustering* (CHIU, 1994b) a fim de se predeterminar o número de *clusters* a serem caracterizados e seus valores centrais.

Com esses agrupamentos é possível então obter os centros e desvios-padrão das RBFs e, aplicando-se o método dos mínimos quadrados, os valores dos pesos de saída dessas. Definido o modelo inicial, pode-se então aplicar um processo de otimização não linear no ajuste dos polos (parâmetros γ) das GOBF, dos centros, das dispersões ou desvios-padrão e dos pesos de saída das redes RBF. Para tanto, são calculados os gradientes exatos da saída com relação aos parâmetros de interesse. Esses gradientes fornecem a direção de busca para os valores ótimos dos parâmetros. São desenvolvidos os cálculos analíticos dos gradientes para a função de saída do modelo GOBF-RBF.

Uma segunda proposta para a obtenção dos parâmetros refinados da estrutura GOBF-RBF em estudo, se dá pelo uso de um algoritmo metaheurístico de otimização chamado de *Particle Swarm Optimization* (PSO) (KENNEDY; EBERHART, 1995). O PSO e suas variantes, tem sido utilizado com sucesso no treinamento de redes neurais com funções de base radial (TSEKOURAS; TSIMIKAS, 2013; CHEN; QIN, 2006; ZHAO *et al.*, 2014). Entre suas variações de maior sucesso, ressalta-se o *quantum-behaved particle swarm optimization* (QPSO) (SUN; FENG; XU, 2004). Destacando seu bom desempenho na localização de mínimos em funções multimodais, uma vez que suas partículas não apresentam o comportamento newtoniano do PSO convencional, mas baseia-se no princípio da incerteza, aumentando sua capacidade de busca (SUN; LAI; WU, 2012).

1.3 Organização do Trabalho

O trabalho está organizado da seguinte maneira:

- **Capítulo 2 - Modelagem de Sistemas Dinâmicos Lineares por Bases de Funções Ortonormais** - O capítulo 2 apresenta a modelagem de sistemas dinâ-

nicos por meio das bases de funções ortonormais, mostrando suas características e vantagens frente a outros métodos. Apresentam-se as principais bases de funções ortonormais utilizadas na modelagem de sistemas, Laguerre, Kautz e as bases de funções generalizadas com funções internas de estrutura *ladder*. Estas últimas são de especial interesse no trabalho aqui apresentado, devido a suas características vantajosas frente às outras abordagens, entre elas a possibilidade de se trabalhar com polos complexos ou não representados apenas por parâmetros reais;

- **Capítulo 3 - Modelagem de Sistemas Dinâmicos Não Lineares por Bases de Funções Ortonormais** - No capítulo 3 apresenta-se a proposta de modelagem de sistemas dinâmicos não lineares por base de funções ortonormais generalizadas com redes neurais com funções de base radial. Uma técnica de agrupamento *fuzzy* é empregada para determinação dos modelos iniciais juntamente com a determinação prévia do número de *clusters* pelo método *subtractive-clustering*. Utiliza-se então algoritmos de otimização não linear baseados no cálculo analítico dos gradientes dos modelos com relação a seus parâmetros, realizando um ajuste fino no modelo, otimizando os polos das GOBF, os centros, larguras (dispersões ou *spreads*) e pesos de saída das redes RBF. Uma segunda proposta para a realização do ajuste fino do modelo é apresentada utilizando técnicas baseadas no enxame de partículas (PSO);
- **Capítulo 4 - Resultados** - No capítulo 4 são apresentados exemplos de modelagem de sistemas não lineares empregando-se as metodologias propostas (otimização não linear e por enxame de partículas), bem como uma análise do efeito da inicialização dos polos sobre a metodologia envolvendo a otimização não linear;
- **Capítulo 5 - Conclusão** No capítulo 5 são apresentadas as conclusões e contribuições do trabalho realizado e perspectivas de trabalhos futuros.

2 Modelagem de Sistemas Dinâmicos Lineares por Bases de Funções Ortonormais

2.1 Introdução

A representação matemática de sistemas dinâmicos é importante em várias áreas da ciência e tecnologia. Modelos para análise, otimização, simulação e controle de sistemas dinâmicos requerem exatidão e processos de identificação eficientes. Uma abordagem de particular relevância é a modelagem de sistemas dinâmicos usando base de funções ortonormais (*Orthonormal Basis Functions* - OBF) (SCHETZEN, 1980; OLIVEIRA *et al.*, 2000; HEUBERGER; HOF; WAHLBERG, 2005). As funções ortonormais mais utilizadas são as de Laguerre e de Kautz (BOKOR; SCHIPP, 1998; WAHLBERG; MAKILA, 1996; BROOME, 1965), as quais são mais apropriadas para representar sistemas com dinâmicas dominantes de primeira ou segunda ordem. Sistemas complexos e de dinâmicas complexas são modelados usando as Bases de Funções Ortonormais Generalizadas (*Generalized Orthonormal Basis Functions* - GOBF) (HEUBERGER; HOF; WAHLBERG, 2005; HOF; HEUBERGER; BOKOR, 1995; NINNESS; GUSTAFSSON, 1997), ao custo de uma parametrização mais complexa. O modelo de GOBF aqui analisado apresenta vantagens sobre outras funções ortonormais, uma vez que apenas parâmetros reais são utilizados na sua representação, independente dos polos serem reais ou complexos, permitindo a representação de sistemas com vários modos dominantes e/ou ordem elevada por modelos mais simples, sem necessidade de especificar as características dinâmicas dos polos a priori.

Modelos formados por bases de funções ortonormais apresentam vantagens como, por exemplo, a possibilidade de incorporar conhecimentos prévios sobre as dinâmicas dominantes do sistema a ser modelado (NINNESS; GUSTAFSSON, 1997). A inclusão dessas informações pode reduzir e simplificar os problemas de identificação e controle associados ao sistema. Modelos baseados em funções ortonormais exibem tolerância a dinâmicas não modeladas e são menos sensíveis aos parâmetros estimados (NELLES, 2001).

Outro ponto a se evidenciar é o fato de modelos OBF não envolverem realimentação da saída, assim, erros de predição não são propagados no modelo, o que, em geral, resulta em representações mais exatas (NELLES, 2001). Devido à completude da base, também é possível aumentar a sua exatidão simplesmente aumentando o número de funções na base (HEUBERGER; HOF; WAHLBERG, 2005).

Os parâmetros dos modelos relacionados às bases ortonormais são os polos das

funções ortonormais e os coeficientes da expansão das séries dessas funções. Os valores dos coeficientes da expansão dependem da escolha dos polos, sendo esses polos parâmetros de escolha livre e sua correta seleção corresponde a um passo importante no processo de identificação de modelos OBF. Quando os polos são selecionados de maneira adequada, a velocidade de convergência da série ortonormal pode aumentar consideravelmente, permitindo reduções de sua ordem sem afetar a exatidão (HOF; HEUBERGER; BOKOR, 1995). Na verdade, caso os polos da base sejam selecionados próximos aos polos dominantes do sistema, uma aproximação precisa pode ser obtida com poucos coeficientes. A seleção desses polos pode ser feita manualmente pelo projetista baseando-se em conhecimentos prévios sobre o sistema ou de maneira automática a partir dos dados de entrada e saída e algum método de otimização (HEUBERGER; HOF; WAHLBERG, 2005; NELLES, 2001).

Em (MACHADO; CAMPELLO; AMARAL, 2013), uma metodologia baseada na informação do gradiente foi utilizada para otimizar e identificar os parâmetros dos modelos das OBF. No entanto, apesar de identificar os parâmetros dos modelos OBFs com exatidão, é necessário determinar o número de polos e funções da base previamente. Em (MACHADO, 2014), uma metodologia para determinar os parâmetros da GOBF, bem como o número de funções da base utilizando algoritmos genéticos foi proposta.

2.2 Bases de Funções Ortonormais

Modelos por bases de funções ortonormais (OBF) podem ser vistos como uma generalização dos modelos FIR, sendo estes um tipo especial de modelo OBF (NELLES, 2001). Pode-se escrever um modelo FIR por

$$y(k) = g_1 z^{-1} u(k) + g_2 z^{-2} u(k) + \dots + g_i z^{-i} u(k) + v(k) \quad (2.1)$$

na qual, $y(k)$ é a saída do filtro, $u(k)$ a entrada e $v(k)$ o ruído branco. Logo, a saída desse modelo pode ser vista como uma combinação linear da entrada $u(k)$ associada com suas amostras anteriores simbolizadas por $z^{-1}, z^{-2}, \dots, z^m$, sendo que todos esses termos possuem polos localizados no centro do círculo unitário. Como a resposta ao impulso de todos esses termos são ortonormais, diz-se que eles formam uma base ortonormal.

Dois sinais discretos $u_1(k)$ e $u_2(k)$ são ditos ortogonais se (NELLES, 2001):

$$\sum_{k=-\infty}^{\infty} u_1(k)u_2(k) = 0 \quad (2.2)$$

$$\sum_{k=-\infty}^{\infty} u_1(k)^2 = \text{constante} \quad e \quad \sum_{k=-\infty}^{\infty} u_2(k)^2 = \text{constante} \quad (2.3)$$

Dois sinais discretos $u_1(k)$ e $u_2(k)$ são ditos ortonormais se:

$$\sum_{k=-\infty}^{\infty} u_1(k)u_2(k) = 0 \quad (2.4)$$

$$\sum_{k=-\infty}^{\infty} u_1(k)^2 = 1 \quad e \quad \sum_{k=-\infty}^{\infty} u_2(k)^2 = 1. \quad (2.5)$$

É sabido que a ordem de modelos FIR para a representação de alguns sistemas deve ser alta (NELLES, 2001). Isso se explica sob a óptica das OBFs, uma vez que as funções da base FIR nada mais são que impulsos defasados, o que pode resultar em uma representação não muito realista da saída esperada. Se uma outra função de base que respeite as definições dadas anteriormente sobre ortonormalidade for escolhida, é possível representar sistemas com um número reduzido de funções de base. Assim, partindo do modelo FIR pode-se substituir os termos triviais z^{-i} por filtros mais generalizados e complexos ϕ . O modelo OBF pode ser dado por

$$\hat{y}(k) = g_1\phi_1(z)u(k) + g_2\phi_2(z)u(k) + \dots + g_i\phi_i(z)u(k) + v(k). \quad (2.6)$$

Assim, a resposta impulsiva do processo, $h(j)$, pode ser modelada através de uma série de funções ortonormais:

$$h(j) = \sum_{i=1}^{\infty} g_i\phi_i(j). \quad (2.7)$$

A resposta ao impulso $h(j)$ de um sistema dinâmico linear para uma aproximação com n_f funções ortonormais pode ser dada como:

$$\begin{aligned} \hat{y}(k) &= \sum_{j=0}^{\infty} h(j)u(k-j) \\ &= \sum_{j=0}^{\infty} \sum_{i=1}^{n_f} g_i\phi_i(j)u(k-j) \\ &= \sum_{i=1}^{n_f} g_i \sum_{j=0}^{\infty} \phi_i(j)u(k-j). \end{aligned}$$

Representando a equação (2.6) de maneira compacta:

$$\hat{y}(k) = \sum_{i=1}^{n_f} g_i\psi_i(k), \quad (2.8)$$

na qual $\psi_i(k)$ representa a convolução da entrada $u(k)$ com a i -ésima função ortonormal ϕ_i no instante k .

As OBFs são bases completas no espaço de funções quadraticamente somáveis no intervalo $[0, \infty)$, sendo que suas funções possuem a capacidade de aproximar com exatidão arbitrária, através de uma combinação linear de finitas funções da base, qualquer função linear no espaço de Lebesgue (HEUBERGER; HOF; WAHLBERG, 2005; CAMPELLO; OLIVEIRA; AMARAL, 2007).

A representação de sistemas dinâmicos lineares e estáveis por modelos baseados em OBFs apresentam vantagens quando comparados a outros tipos de modelagem:

- É garantido que a representação de um sistema estável também será estável;
- Um aumento no número de funções do modelo acarreta em um conseqüente aumento na capacidade de aproximação, o que pode ser feito adaptativamente;
- A utilização de OBFs com polos próximos aos polos dominantes do sistema que ela representa acarreta em uma quantidade menor de termos que em um modelo FIR para uma mesma exatidão;
- Ausência de realimentação de saída, o que evita a propagação de erros na predição de modelos e, conseqüentemente, sistemas mais exatos;
- Não requer a determinação prévia da ordem do modelo nem do seu atraso de transporte, eliminando a etapa de determinação de regressores;
- Modelos baseados em OBF apresentam tolerância à dinâmicas não modeladas e sensibilidade reduzida à dinâmicas não dominantes;
- Capacidade de lidar de forma robusta com atrasos de transporte;
- Desacoplamento natural das saídas em sistemas multivariáveis.

A modelagem por base de funções ortonormais admite a representação do modelo em espaço de estados. Dado um conjunto de polos é possível representar o sistema por

$$\begin{bmatrix} x(k+1) \\ \hat{y}(k) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}. \quad (2.9)$$

O modelo apresentado na equação (2.8) pode ser representado no espaço de estados como:

$$\begin{aligned} \Psi(k+1) &= A_f \Psi(k) + B_f u(k) \\ \hat{y}(k) &= C_f \Psi(k) \end{aligned} \quad (2.10)$$

na qual as matrizes A_f e B_f dependem dos parâmetros do modelo, C_f representa os termos da expansão em série da saída das funções ortonormais e $\Psi(k) = [\psi_1(k) \dots \psi_{n_f}(k)]^T$ é o vetor de estados ortonormais de ordem n_f , sendo que são coincidentes com as saídas dos filtros correspondentes às funções ortonormais de ordem equivalente. Caso desejado, este modelo pode incluir a parte estática que modela a saída do sistema a ser modelado.

Existem diferentes possibilidades na escolha dos filtros, porém três são as mais usuais. Os filtros utilizando funções de Laguerre permitem incorporar a dinâmica de sistemas com um polo real para processos de comportamento aperiódicos. Os filtros de Kautz incorporam a dinâmica de processos com comportamentos oscilatórios apresentando um par de polos complexos conjugados. Por fim, as bases de funções ortonormais generalizadas (GOBF), que incluem os filtros de Laguerre e Kautz como casos especiais, permitem um número arbitrário de polos reais e/ou complexos conjugados.

A base de Laguerre é dada por:

$$\Phi_i(z) = \frac{\sqrt{1-p^2}}{z-p} \left(\frac{1-pz}{z-p} \right)^{i-1}, \quad i = 1, 2, \dots \quad (2.11)$$

sendo que $p = \{p : p \in \mathbb{R} \text{ e } |p| < 1\}$ é o polo que parametriza as funções ortonormais e i a ordem da função. Como pode-se notar por (2.11), os modelos de Laguerre de ordem i podem ser escritos como modelos em cascatas de ordem $i - 1$, conforme ilustrado na Figura 1.

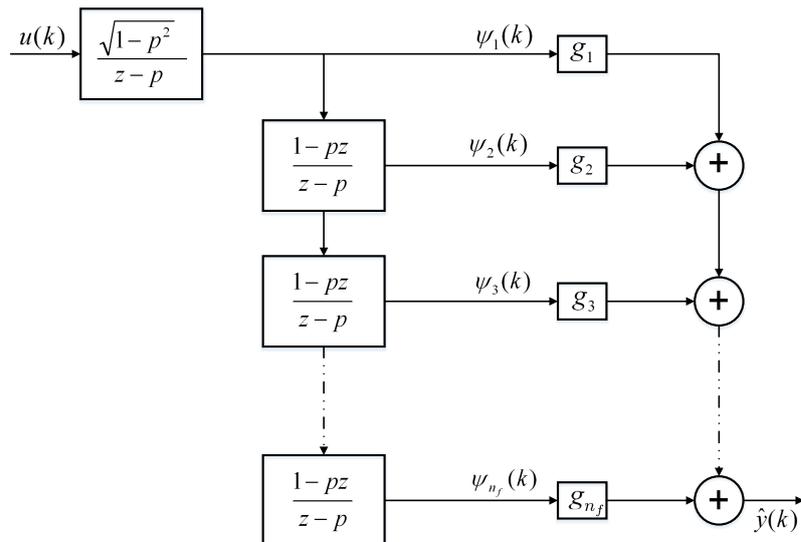


Figura 1 – Representação em blocos das Funções de Laguerre.

Pode-se representar os modelos de Laguerre de acordo com (2.10), sendo as matrizes A_f , B_f e C_f dadas por

$$A_f = \begin{bmatrix} p & 0 & 0 & \dots & 0 \\ 1 - p^2 & p & 0 & \dots & 0 \\ (-p)(1 - p^2) & 1 - p^2 & p & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (-p)^{n_f-2}(1 - p^2) & (-p)^{n_f-3}(1 - p^2) & \dots & p \end{bmatrix} \quad (2.12)$$

$$B_f = \sqrt{1 - p^2} \begin{bmatrix} 1 & -p & (-p)^2 & \dots & (-p)^{n_f-1} \end{bmatrix} \quad (2.13)$$

na qual p é o polo de Laguerre, n_f é o número de filtros de Laguerre. E g_i , na Figura 1, são os coeficientes da expansão da série.

As funções de Kautz (WAHLBERG, 1994) são parametrizadas pelo par de polos conjugados $\beta = \alpha \pm jv$ segundo a equação (2.14).

$$\begin{aligned} \Phi_{2i-1}(z) &= \frac{\sqrt{(1-c^2)(z-b)}}{z^2+b(c-1)z-c} \left(\frac{-cz^2+b(c-1)z+1}{z^2+b(c-1)z-c} \right)^{i-1} \\ \Phi_{2i}(z) &= \frac{\sqrt{(1-b^2)(1-c^2)}}{z^2+b(c-1)z-c} \left(\frac{-cz^2+b(c-1)z+1}{z^2+b(c-1)z-c} \right)^{i-1} \end{aligned} \quad (2.14)$$

na qual

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

sendo $0 < b < 1$ e $-1 < c < 0$. As funções de Kautz podem ser representadas de forma recursiva, como apresentado na Figura 2.

A descrição na forma de espaço de estados de Kautz foi desenvolvida por Mbarek, Messaoud e Favier (2003) e é dada por

$$\begin{aligned} \Psi(k+1) &= A_1\Psi(k) + A_2\Psi(k-1) + BU_2(k+1) \\ \hat{y}(k) &= C\Psi(k+1) \end{aligned} \quad (2.15)$$

na qual

$$U_2(k+1) = \begin{bmatrix} u(k) & u(k+1) \end{bmatrix}^T$$

$$A_1 = \begin{bmatrix} \alpha_{1,1} & 0 & \dots & 0 \\ \alpha_{2,1} & \alpha_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n_f,1} & \alpha_{n_f,2} & \dots & \alpha_{n_f,n_f} \end{bmatrix}$$

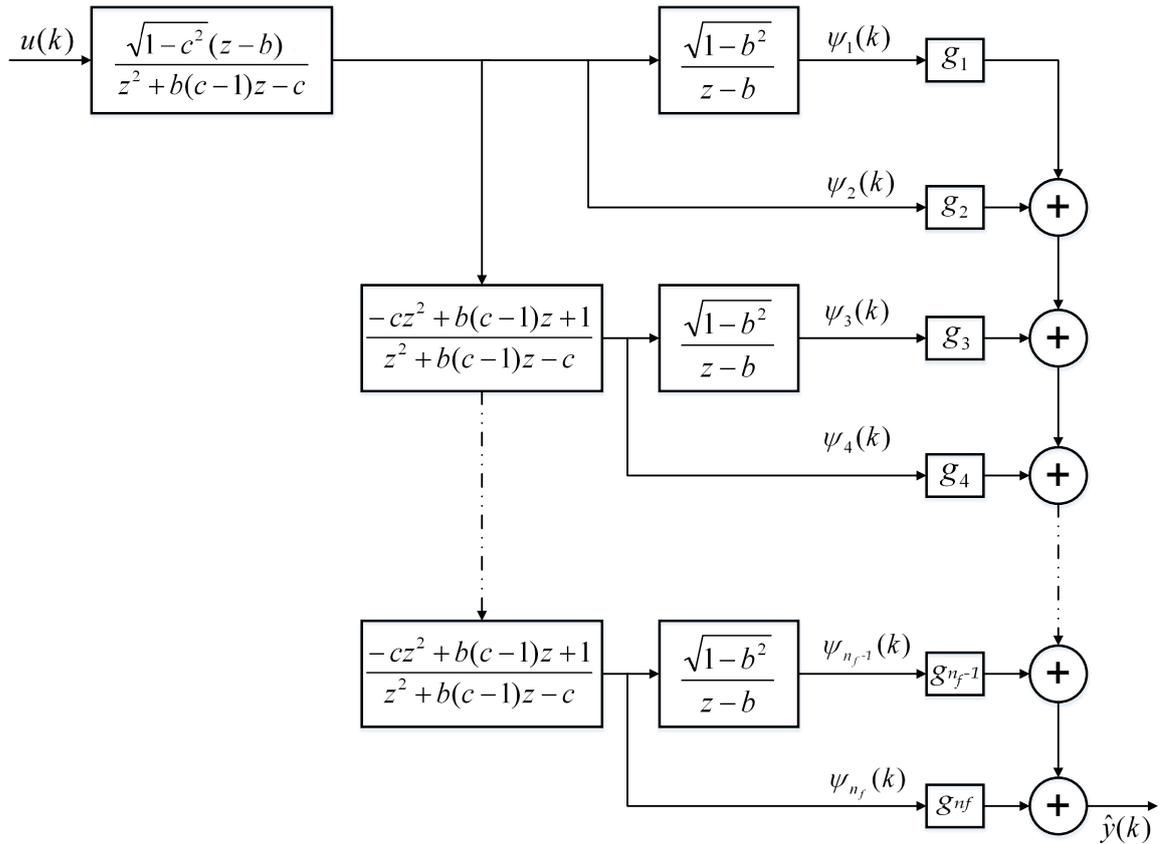


Figura 2 – Representação em blocos das Funções de Kautz.

$$A_2 = \begin{bmatrix} \beta_{1,1} & 0 & \dots & 0 \\ \beta_{2,1} & \beta_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{n_f,1} & \beta_{n_f,2} & \dots & \beta_{n_f,n_f} \end{bmatrix}$$

$$B = \sqrt{1-c^2} \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} \\ \sigma_{2,1} & \sigma_{2,2} \\ \vdots & \vdots \\ \sigma_{n_f,1} & \sigma_{n_f,2} \end{bmatrix}$$

$$C = [g_1 \ g_2 \ \dots \ g_{n_f}]$$

Os elementos da matriz A_1 são definidos da seguinte maneira:

- Se $i = 1$

$$\alpha_{1,1} = -b(c-1)$$

- Se $i > 1$

$$\begin{aligned}\alpha_{i,i} &= -b(c-1) \\ \alpha_{i,(i-2)} &= -c\alpha_{i,i} + b(c-1) \\ &\vdots \\ \alpha_{i,j} &= -c\alpha_{i,(j+2)} \\ &\vdots \\ \alpha_{i,1} &= -c\alpha_{i,3}.\end{aligned}$$

Os elementos da matriz A_2 são definidos da seguinte maneira:

- Se $i = 1$

$$\beta_{1,1} = c$$

- Se $i > 1$

$$\begin{aligned}\beta_{i,i} &= c \\ \beta_{i,(i-2)} &= -c\beta_{i,i} + 1 \\ &\vdots \\ \beta_{i,j} &= -c\beta_{i,(j+2)} \\ &\vdots \\ \beta_{i,1} &= -c\beta_{i,3}.\end{aligned}$$

Os elementos da matriz B são definidos da seguinte maneira:

$$\sigma_{i,1} = (-c)^{\binom{i-1}{2}}(-b), \quad \text{para } i = 1, \dots, nf.$$

$$\sigma_{i,1} = (-c)^{\binom{i-1}{2}}, \quad \text{para } i = 1, \dots, nf.$$

Quando o sistema a ser modelado necessitar de mais que dois polos distintos para ser representado com exatidão, deve-se utilizar modelos por base de funções ortonormais generalizadas nas quais as funções ortonormais podem apresentar n_b polos distintos.

2.3 Base de Funções Ortonormais Generalizadas - GOBFs

A classe das funções ortonormais generalizadas é obtida pela conexão de n filtros *all-pass* de n_b -ésima ordem. Cada filtro terá n_b polos distintos ou não, reais ou complexos (HEUBERGER; HOF; WAHLBERG, 2005; NINNESS; GUSTAFSSON, 1997). A Figura 3 apresenta a estrutura em blocos de um modelo GOBF, no qual os blocos representados por G_b são as funções ortonormais que compõem a base e C_n^T (com $i = 1, \dots, n$), denotando os correspondentes subvetores dos coeficientes da expansão, os quais são os pesos das saídas dos filtros antes de se somarem e formarem a saída final $\hat{y}(k)$ do modelo.

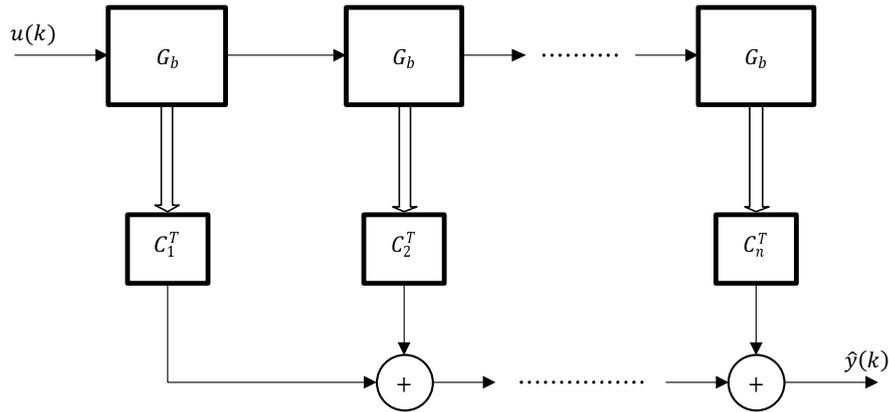


Figura 3 – Representação em blocos de um modelo GOBF.

Um caso especial de GOBF corresponde em conectar diferentes filtros *all-pass* de primeira e segunda ordem, no qual cada filtro pode apresentar polo(s) distinto(s). Essas funções são também conhecidas por funções de Takenaka-Malmquist (HEUBERGER; HOF; WAHLBERG, 2005), dadas por

$$F_\kappa(z) = \frac{\sqrt{1 - |\xi_\kappa|^2}}{z - \xi_\kappa} \prod_{i=1}^{\kappa-1} \left[\frac{1 - \xi_i^* z}{z - \xi_i} \right] \quad (2.16)$$

em que $\xi_i, i = 1, \dots, \kappa$ denotam os polos do modelo GOBF. Contudo, este tipo de modelagem apresenta algumas complicações, principalmente quando as características dinâmicas do sistema a ser modelado são desconhecidas, entre elas:

- A parametrização não é única, já que qualquer permutação de polos resultará em modelos idênticos;
- É necessário especificar a priori quantos serão os polos reais e os polos complexos conjugados;
- A saída dos filtros com polos complexos é um valor complexo, contudo esse contexto pode ser contornada com uma transformação proposta por Ninness e Gustafsson (1997);
- Para a determinação dos parâmetros no caso de par de polos complexos conjugados (partes reais e imaginárias), os parâmetros são interdependentes, uma vez que os polos devem estar dentro do círculo unitário (HEUBERGER; HOF; WAHLBERG, 2005).

Felizmente, existe outra parametrização que não apresenta as complicações descritas acima. Esta representação é conhecida como base de funções ortonormais com funções internas, na qual as funções de Laguerre e de Kautz são casos especiais (HEUBERGER; HOF; WAHLBERG, 2005).

2.3.1 Funções Internas

Uma função de transferência racional $H(z)$ é dita “interna” caso seja estável e satisfaça a equação (2.17), sendo assim um filtro *all-pass*.

$$H(z)H(1/z) = 1 \quad (2.17)$$

Os blocos *all-pass* mais simples são:

- Filtros *all-pass* de primeira ordem com função de transferência

$$H_i(z) = \frac{z - a_i}{1 - a_i z}, \quad -1 < a_i < 1, \quad (2.18)$$

com realização ortonormal no espaço de estados dada por

$$\begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix} = \left[\begin{array}{c|c} a_i & \sqrt{1 - a_i^2} \\ \hline \sqrt{1 - a_i^2} & -a_i \end{array} \right]. \quad (2.19)$$

Esta representação de modelos com polo real é equivalente à representação com funções de Laguerre.

- Filtros *all-pass* de segunda ordem com função de transferência

$$H_i(z) = \frac{-c_j z^2 + b_j(c_j - 1)z + 1}{z^2 + b_j(c_j - 1)z - c_j}, \quad -1 < b_j < 1, \quad -1 < c_j < 1, \quad (2.20)$$

com realização ortonormal no espaço de estados dada por

$$\begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix} = \left[\begin{array}{cc|c} b_j & b_j \sqrt{1 - b_j^2} & \sqrt{1 - b_j^2} \sqrt{1 - c_j^2} \\ \hline \sqrt{1 - b_j^2} & -b_j c_j & -b_j \sqrt{1 - c_j^2} \\ 0 & \sqrt{1 - c_j^2} & -c_j \end{array} \right]. \quad (2.21)$$

Sendo que esta realização não é única. Esta representação de modelos dois polos reais é equivalente à representação com funções de Kautz.

- Uma seção de filtros *all-pass* com polo na origem e duas entradas e duas saídas

$$\begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix} = \left[\begin{array}{c|cc} 0 & 0 & 1 \\ \hline \sqrt{1 - \gamma_j^2} & -\gamma_j & 0 \\ \gamma_j & \sqrt{1 - \gamma_j^2} & -0 \end{array} \right] \quad (2.22)$$

na qual $-1 < \gamma_j < 1$.

A conexão em cascata de n_b funções internas de primeira ordem gera outras funções internas de ordem n_b (HEUBERGER; HOF; WAHLBERG, 2005), mas quando envolvem polos complexos conjugados seus parâmetros geralmente se tornam complexos ou requerem alguma transformação para que se tornem reais. Para evitar essas transformações utiliza-se a última função interna apresentada anteriormente, esta representação para modelos GOBF garante que os polos sejam parametrizados sempre por parâmetros reais independente da natureza dos polos. Esta representação foi proposta originalmente por Gray e Markel (1975), considerando inicialmente como uma realização em espaço de estados para uma função interna com somente um polo com duas entradas e duas saídas cujo modelo em estado pode ser reescrito por (HEUBERGER; HOF; WAHLBERG, 2005; GRAY; MARKEL, 1975)

$$\begin{bmatrix} \psi_i(k+1) \\ y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \sqrt{1-\gamma^2} & -\gamma & 0 \\ \gamma & \sqrt{1-\gamma^2} & 0 \end{bmatrix} \begin{bmatrix} \psi_i(k) \\ u_1(k) \\ u_2(k) \end{bmatrix} \quad (2.23)$$

com $-1 < \gamma < 1$. O modelo (2.23) pode ser representado através do diagrama de blocos da Figura 4.

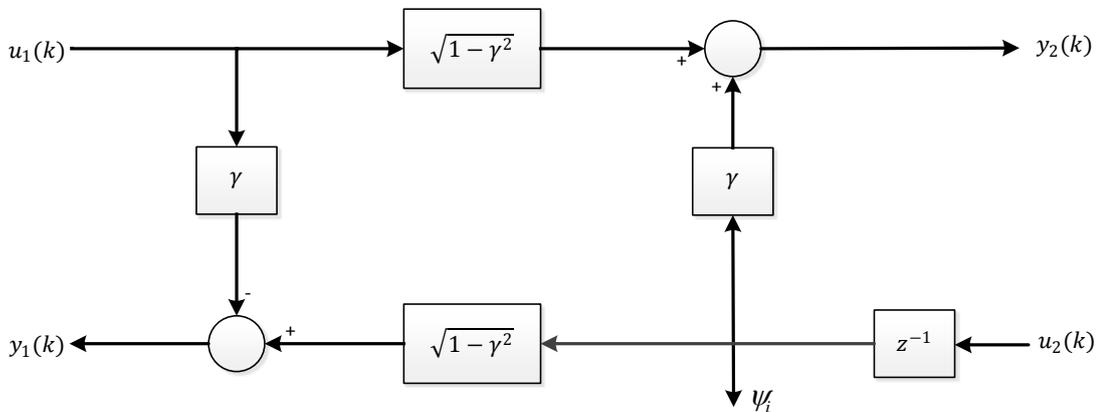


Figura 4 – Representação em blocos da estrutura interna parametrizada por γ .

Essa representação é derivada do modelo dado por (2.22), uma vez que a conexão da saída $y_2(k)$ com a entrada $u_2(k)$ as torna equivalentes. Esta estrutura favorece a conexão em cascata das funções de primeira ordem e fornece uma metodologia para obtenção de modelos GOBF parametrizados apenas por parâmetros reais (GRAY; MARKEL, 1975; HEUBERGER; HOF; WAHLBERG, 2005).

Esta estrutura é normalizada, gerando uma base ortonormal. Bases com somente um parâmetro γ são equivalentes às bases com funções de Laguerre e bases com dois pa-

râmetros são equivalentes às bases com funções de Kautz (HEUBERGER; HOF; WAHLBERG, 2005). A ligação dos n_b blocos com distintos parâmetros γ_b forma uma base ortonormal generalizada com n_b polos, que podem ou não serem distintos, e se dá através da estrutura apresentada na Figura 5.

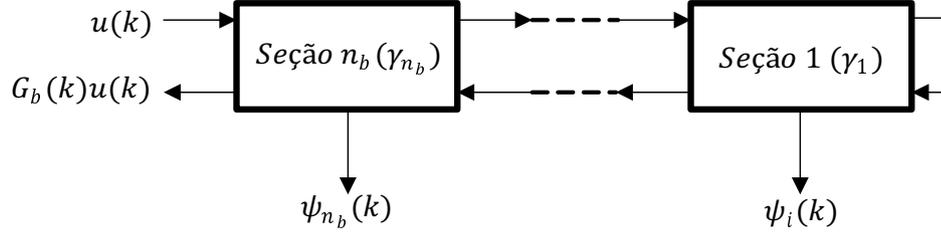


Figura 5 – Representação em blocos das funções da base de uma GOBF.

As estruturas GOBF são constituídas, como mostrado na Figura 3, pela ligação em cascata de n blocos G_b formados pela base de funções ortonormais com polos reais e/ou complexos conjugados, de maneira que o número total de funções na base ortonormal resultante seja $n_f = n_b n$. Neste trabalho utiliza-se a representação em espaço de estados para computar as saídas das GOBFs com funções internas. O modelo em espaço de estados de um único bloco G_b com n_b funções internas, parametrizadas por γ_i com $i = 1, \dots, n_b$ é dado por

$$\begin{aligned}\Psi_b(k+1) &= A_b \Psi_b(k) + B_b u(k) \\ \hat{y}_b(k) &= C_b \Psi_b(k)\end{aligned}\quad (2.24)$$

na qual $\Psi_b(k) = [\psi_1(k) \dots \psi_{n_b}(k)]$ representam as saídas dos filtros (estados) que compõem o bloco G_b da GOBF. As matrizes A_b e B_b dependem diretamente dos parâmetros γ_i , como descrito por Machado, Campello e Amaral (2013) e C_b é o respectivo subvetor dos coeficientes da expansão em séries do modelo GOBF.

As matrizes A_b e B_b dependem diretamente dos parâmetros γ_i , com $i = 1, \dots, n_b$. A matriz A_b é definida por:

$$A_b = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n_b} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n_b} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n_b} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n_b,1} & a_{n_b,2} & a_{n_b,3} & \dots & a_{n_b,n_b} \end{bmatrix} . \quad (2.25)$$

Os termos $a_{i,j}$, com i sendo a linha e j a coluna a qual o elemento pertence, são dados por:

- Para $i = 1$, os elementos são:

$$\begin{aligned} a_{1,j} &= \gamma_1, & \text{para } j = 1; \\ a_{1,j} &= \gamma_j \prod_{l=1}^{j-1} \sqrt{1 - \gamma_l^2}, & \text{para } j = 2, \dots, n_b. \end{aligned}$$

- Para $i = 2, \dots, n_b$, os elementos são:

$$\begin{aligned} a_{i,j} &= \sqrt{1 - \gamma_j^2}, & \text{para } j = i - 1; \\ a_{i,j} &= -\gamma_{j-1}\gamma_j, & \text{para } j = i; \\ a_{1,j} &= -\gamma_{i-1}\gamma_j \prod_{l=i}^{j-1} \sqrt{1 - \gamma_l^2}, & \text{para } j = i + 1, \dots, n_b; \\ a_{i,j} &= 0, & \text{para os demais elementos.} \end{aligned}$$

A matriz B_b do modelo é definida por:

$$B_b = \begin{bmatrix} b_{1,1} \\ b_{2,1} \\ b_{3,1} \\ \vdots \\ b_{n_b,1} \end{bmatrix} \quad (2.26)$$

com seus elementos dados por:

- para $i = 1$, o elemento será:

$$b_{1,1} = \prod_{l=1}^{n_b} \sqrt{1 - \gamma_l^2}.$$

- Para $i = 2, \dots, n_b$, os elementos são:

$$b_{i,1} = -\gamma_{i-1} \prod_{l=i}^{n_b} \sqrt{1 - \gamma_l^2}.$$

A matriz C_b é dada por:

$$C_b = \begin{bmatrix} g_1 & g_2 & \dots & g_{n_b} \end{bmatrix}$$

na qual g_i , com $i = 1, \dots, n_b$, são os coeficientes da expansão em série do modelo GOBF.

O modelo genérico da GOBF, composto do cascadeamento de n blocos G_b , pode ser representado na forma de espaço de estados como mostrado em (2.27), na qual $\Psi(k)$ e composto pela saída de $n_b n$ filtros ortonormais

$$\begin{aligned} \Psi(k+1) &= A_f \Psi(k) + B_f u(k) \\ \hat{y}(k) &= C_f \Psi(k). \end{aligned} \quad (2.27)$$

A matriz (A_f) do modelo (2.27) é definida por

$$A_f = \begin{bmatrix} A_b & 0 & 0 & \cdots & 0 \\ A_I & A_b & 0 & \cdots & 0 \\ (-\gamma_{n_b})A_I & A_I & A_b & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (-\gamma_{n_b})^{n-2}A_I & (-\gamma_{n_b})^{n-3}A_I & (-\gamma_{n_b})^{n-4}A_I & \cdots & A_b \end{bmatrix} \quad (2.28)$$

sendo as submatrizes A_b definidas como (2.25). As submatrizes A_I têm seus elementos $a_{I_{i,j}}$, na qual i é sua linha e j sua coluna, definidos por:

- Para $i = 1, \dots, n_b$ e $j \neq n_b$, o elemento $a_{I_{i,j}}$ será:

$$a_{I_{i,j}} = 0;$$

- para $j = n_b$, os elementos $a_{I_{i,j}}$ serão:

Se $i = 1$

$$a_{I_{i,j}} = (1 - \gamma_{n_b}^2) \prod_{l=1}^{n_b-1} \sqrt{1 - \gamma_l^2}.$$

Se $i = 2, \dots, n_b$

$$a_{I_{i,j}} = (1 - \gamma_{n_b}^2)(-\gamma_{i-1}) \prod_{l=i}^{n_b-1} \sqrt{1 - \gamma_l^2}.$$

A matriz B_f presente no modelo (2.27) é formada como mostrado em (2.29).

$$B_f = \begin{bmatrix} B_b \\ (-\gamma_{n_b})B_b \\ (-\gamma_{n_b})^2B_b \\ \vdots \\ (-\gamma_{n_b})^{n-1}B_b \end{bmatrix} \quad (2.29)$$

A matriz C_f é definida por:

$$C_f = \begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,n_b} & \cdots & g_{1,n_b n} \end{bmatrix}$$

na qual $g_{1,j}$, com $j = 1, \dots, n_b n$, são os coeficientes da expansão em série do modelo GOBF.

A representação em espaço de estados do modelo GOBF com funções internas apresentada permite que sejam obtidos os polos do modelo através da obtenção dos autovalores da matriz A_f . Uma possível conversão do modelo em espaço de estados para função de transferência foi dado por [Heuberger, Hof e Bosgra \(1995\)](#) e é representada por (2.30), característica essa interessante, uma vez que o modelo foi parametrizado apenas pelos parâmetros reais γ independentemente das características dos polos do sistema, de forma que os polos do sistema são determinados pelos autovetores da matriz A_f .

$$G_k(z) = C_f(zI - A_f)^{-1}B_f. \quad (2.30)$$

2.4 Ajuste de parâmetros das GOBF

Uma vez apresentada a modelagem de sistemas dinâmicos lineares por base de funções ortonormais, o passo que segue se trata da obtenção dos parâmetros que a constituem. Os parâmetros a serem determinados são os polos das funções ortonormais (γ_l) e os coeficientes da expansão em série de funções (C_f). Como discutido em trabalhos anteriores ([OLIVEIRA, 1997](#); [CAMPELLO; AMARAL, 2002](#); [CAMPELLO; OLIVEIRA; AMARAL, 2007](#); [MACHADO; CAMPELLO; AMARAL, 2013](#)), uma vez determinados os polos, pode-se determinar os termos da expansão em série pelo método dos mínimos quadrados.

A seleção dos polos é de relativa complexidade, uma vez que em geral depende de um conhecimento prévio do sistema a ser modelado ou de métodos de otimização que minimizem o erro do modelo estimado. Em trabalhos como ([ROSA, 2009](#); [ROSA; CAMPELLO; AMARAL, 2009](#)) é apresentada uma metodologia de otimização para os modelos de Kautz e GOBF com funções de Takenaka-Malmquist através de buscas exatas dadas pelos cálculos dos gradientes de saída do modelo com relação a seus parâmetros. [Machado \(2014\)](#) propôs uma técnica envolvendo algoritmo genético para determinação dos parâmetros de GOBFs com funções internas, juntamente com uma métrica baseada no critério de Akaike ([AKAIKE, 1974](#)) para a seleção de um modelo parcimonioso. Em ([MACHADO, 2011](#)) os cálculos analíticos dos gradientes para as funções de Kautz, de Laguerre e GOBF com funções internas foram realizados e a otimização dos modelos foi feita com base em técnicas de otimização não lineares ([LUENBERGER, 2003](#)). Esses resultados servem como base para os desenvolvimentos propostos neste trabalho, sendo os cálculos para as GOBFs com funções internas de especial interesse. Os desenvolvimentos desses gradientes analíticos encontram-se no apêndice [A](#).

3 Modelagem de Sistemas Dinâmicos Não Lineares por Bases de Funções Ortonormais

3.1 Introdução

Os sistemas dinâmicos encontrados na prática são, na maioria dos casos, não lineares. Em muitas situações uma representação linear é suficiente para fins de controle ou outras análise. Quando as técnicas convencionais não são suficientes para caracterizar tais sistemas com qualidade é necessário se valer de representações não lineares. Muitas vezes, ao custo inevitável do aumento da complexidade do modelo e algoritmos de identificação a serem utilizados (AGUIRRE, 2007b).

Este capítulo apresenta a modelagem de sistemas dinâmicos SISO (*Single Input Single Output*), discretos, não lineares sob o prisma das GOBFs aliadas a RBFs, em uma estrutura semelhante aos já conhecidos modelos de Wiener (LJUNG, 1999), como o apresentado na Figura 6.

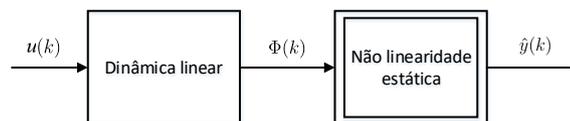


Figura 6 – Estrutura de um modelo de Wiener.

Os modelos não lineares analisados são descritos por

$$\hat{y}(k) = \mathcal{H}(\Phi(k)), \quad (3.1)$$

cuja $\hat{y}(k)$ é a saída estimada ou prevista no instante k e $\Phi(k)$ é o vetor de regressores, que neste trabalho são as saídas dos filtros das GOBFs. O operador \mathcal{H} é um operador estático não linear genérico que realiza uma transformação nas saídas dos filtros. Assim, a dinâmica do sistema fica concentrada no vetor $\Phi(k)$.

Neste trabalho utiliza-se uma rede RBF como operador \mathcal{H} (CAMPELLO; OLIVEIRA; AMARAL, 2007). Redes de base radial foram introduzidas por Broomhead e Lowe (1988) tendo em vista as características de aproximador universal dessa classe de funções (POWELL, 1987; LOWE, 1989). Elas têm sido utilizadas com sucesso para a representação de sistemas dinâmicos não lineares desde sua concepção (CHEN *et al.*, 1990a;

CHEN *et al.*, 1990b; ABBAS *et al.*, 2015; GAN *et al.*, 2015). Uma de suas características é a capacidade de aproximar funções com exatidão arbitrária simplesmente aumentando-se o número de funções na rede (NELLES, 2001).

Neste capítulo é proposta uma nova metodologia, baseada em técnicas de otimização não linear, para a obtenção de modelos GOBF-RBF sem conhecimentos a priori das características estáticas ou dinâmicas do sistema a ser modelado. O primeiro passo do método é a obtenção de um modelo inicial GOBF-RBF utilizando a técnica de agrupamento *Fuzzy C-Means* (BEZDEK, 1980; BABUSKA; VEEN; KAYMAK, 2002) em conjunto com o *Subtractive Clustering* (CHIU, 1994b) para predeterminar o número de *clusters* a serem caracterizados e seus respectivos centros. Na sequência, através de técnicas de otimização não linear, otimizam-se os polos (parâmetros γ) das GOBF, os centros, as dispersões ou desvios-padrão e os pesos de saída das redes RBF, para tanto, são obtidos os gradientes exatos da saída com relação a esses parâmetros.

Uma segunda proposta, partindo-se do mesmo modelo inicial utilizado pela otimização não linear, se baseia no uso de algoritmos baseados em enxame de partículas, otimizando-se os polos (parâmetros γ) das GOBF, os centros, as dispersões ou desvios-padrão das redes RBF.

3.2 Redes Neurais com Funções de Base Radial

Em contraste com as redes MPL (*Multi Layer Perceptron*) convencionais, as redes de base radial (RBF - *Radial Basis Functions*) se valem de um mecanismo de construção radial. Esse fato garante aos parâmetros de sua camada oculta uma melhor interpretação àquela dada as MLP's, e, portanto, oferecem novas e mais rápidas metodologias de treinamento. Broomhead e Lowe (1988) foram os primeiros a discutir as RBFs no contexto de redes neurais. Seu potencial de interpolação e suas propriedades de generalização foram discutidas em (LOWE, 1989).

3.2.1 Os neurônios das redes de base radial

A estrutura do neurônio de uma rede RBF pode ser explicada em duas partes. Na primeira a distância do vetor de dados de entrada $\underline{u} = [u_1 \ u_2 \ \cdots \ u_m]^T$ ao vetor de centros $\underline{v}_i = [v_{i,1} \ v_{i,2} \ \cdots \ v_{i,m}]^T$ das funções de base radial com relação a matriz $\underline{\Sigma}_i$ é calculada. Na segunda parte, essa distância, dada por um escalar x , é transformada por uma função de ativação não linear $g(x)$. Escolhas típicas para a função de ativação são a função Gaussiana, dada por

$$g(x) = \exp\left(-\frac{1}{2}x^2\right), \quad (3.2)$$

e a função multi-quadrática inversa, que possui um parâmetro livre adicional a , dada por

$$g(x) = \frac{1}{\sqrt{x^2 + a^2}}. \quad (3.3)$$

Ao longo deste trabalho será usada a função Gaussiana como função de ativação, por ser a mais comumente utilizada na representação de sistemas dinâmicos. Considerando vetores de dados de entrada \underline{u} , os centros da função radial \underline{v}_i e a matriz com o inverso das respectivas variâncias na diagonal principal $\underline{\Sigma}_i$, a representação da norma das distâncias é expressa por (3.4), e as funções de base radial por (3.5).

$$x_i = \|\underline{u} - \underline{v}_i\|_{\underline{\Sigma}_i} = \sqrt{(\underline{u} - \underline{v}_i)^T \underline{\Sigma}_i (\underline{u} - \underline{v}_i)} \quad (3.4)$$

$$G_i(\underline{u}) = \exp\left(-\frac{1}{2}\|\underline{u} - \underline{v}_i\|_{\underline{\Sigma}_i}^2\right) \quad (3.5)$$

na qual

$$\|\underline{u} - \underline{v}_i\|_{\underline{\Sigma}_i} = \sqrt{\sum_{j=1}^m \left(\frac{u_j - v_{i,j}}{\sigma_{i,j}}\right)^2} = \sqrt{\left(\frac{u_1 - v_{i,1}}{\sigma_{i,1}}\right)^2 + \dots + \left(\frac{u_m - v_{i,m}}{\sigma_{i,m}}\right)^2} \quad (3.6)$$

$$\underline{\Sigma}_i = \begin{bmatrix} 1/\sigma_{i,1}^2 & 0 & \dots & 0 \\ 0 & 1/\sigma_{i,2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/\sigma_{i,m}^2 \end{bmatrix}. \quad (3.7)$$

Então, existem três possibilidades na escolha da matriz $\underline{\Sigma}_i$:

- Desvios-padrão idênticos em todas as dimensões, o que levaria a uma função de base radial com contornos circulares;
- Desvios-padrão diferentes para cada dimensão, resultando em uma função de base simétrica mas com contornos elípticos;
- O uso de uma matriz de covariância completa, o que permitiria rotações nas funções de base.

3.2.2 Estrutura da Rede Neural RBF

Assim como as redes MLP, uma rede RBF é um aproximador universal (HAYKIN, 1999). Sua estrutura difere de uma rede MLP devido ao fato de possuir apenas uma camada oculta, enquanto a MLP possui camadas ocultas de acordo com sua especificação. Uma estrutura RBF com múltiplas camadas ocultas não faria muito sentido, visto que uma de suas principais vantagens é a possibilidade de parametrizar a camada oculta com conhecimentos prévios do sistema a ser modelado; uma rede RBF com duas camadas ocultas, por exemplo, apresentaria problemas na parametrização da segunda camada, uma vez que a entrada dessa (a saída da primeira camada) não estaria limitada a um espaço conhecido e interpretável (NELLES, 2001). Sendo assim, uma rede RBF possui três camadas: A camada de entrada, a camada oculta e a camada de saída. A Figura 7 ilustra a estrutura típica de uma rede RBF.

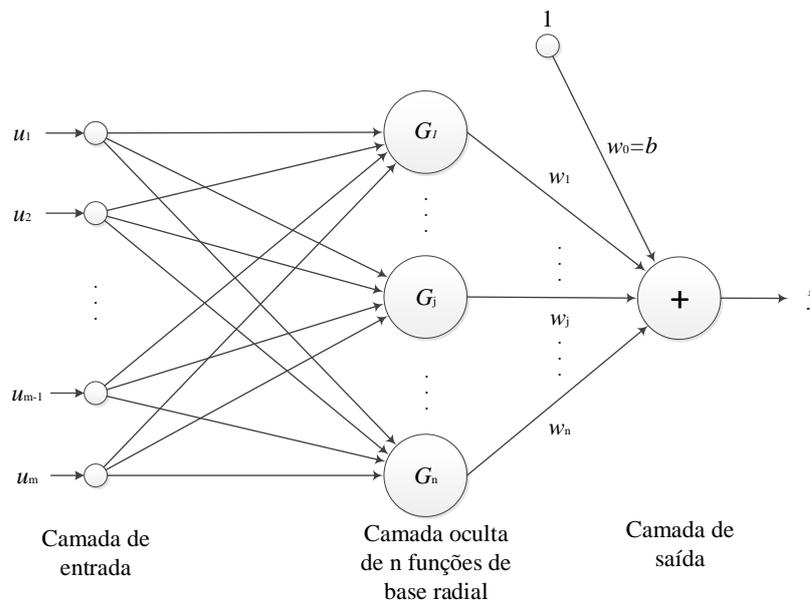


Figura 7 – Estrutura típica de uma rede neural RBF.

A camada de entrada é responsável por fornecer os nós que conectam as informações das entradas da rede com a sua estrutura interna.

A camada oculta é constituída de funções de ativação de base radial. Ela realiza uma transformação não linear do espaço de entrada de dados da modelagem RBF. Os valores dos centros e dos desvios padrão das funções nessa camada determinam o comportamento da rede, como apresentado na seção anterior.

A camada de saída da rede possui pesos que ponderam as informações de saídas de cada função de base radial, agregando-as linearmente.

Assim, a saída de uma rede RBF pode ser escrita como

$$\hat{y} = \sum_{i=0}^n w_i G_i (\|\underline{u} - \underline{v}_i\|_{\underline{\Sigma}_i}) \quad \text{com } G_0(\cdot) = 1, \quad (3.8)$$

cujos pesos da camada de saída são dados por w_i . Os parâmetros da camada oculta são o vetor de centros \underline{v}_i , que representa a posição da i -ésima função de base radial e a matriz $\underline{\Sigma}_i$, que representa as larguras e rotações dessa função.

Assim, uma rede RBF consiste de três parâmetros:

- Os pesos da camada de saída (w_i), que são parâmetros lineares e determinam as cotas das funções de base radial, bem como seu valor de *offset*;
- Os desvios-padrão, que são parâmetros não lineares da camada oculta e determinam as larguras (e possivelmente as rotações) das funções de base radial;
- Os centros (v_i), que são parâmetros não lineares da camada oculta e determinam as posições das funções de base radial.

Enquanto a determinação das ponderações da camada de saída pode ser classificada como um problema básico de regressão linear, a determinação dos valores dos centros e das larguras das funções constitui uma tarefa mais complexa, visto que a rede possui uma alta sensibilidade a escolha das larguras (desvios-padrão pequenos causam afundamentos nas respostas, caso sejam grandes, sobressaltos e muito grandes, sua capacidade de extrapolação cai a zero), evidenciando a necessidade de um treinamento adequado.

3.2.3 Treinamento das redes RBF

Segundo Haykin (1999) as estratégias para ajuste dos parâmetros de redes RBF são tipicamente classificadas como, empíricas, auto-organizadas e supervisionadas. Um exemplo de estratégia empírica é a seleção aleatória dos centros das funções de base radial, sendo os valores das larguras estimadas por

$$\sigma_i^* = \frac{\beta d_{max}}{\sqrt{n}} \quad (3.9)$$

sendo β é um fator positivo, d_{max} é a maior distância entre todos os pares de centros das RBFs e n é o número de RBFs utilizadas.

Moody e Darken (1989) propuseram uma aproximação diferenciada para os valores das larguras dada por

$$\sigma_i^* = \frac{1}{r} \left(\sqrt{\sum_{j=1}^r d_{i,j}^2} \right), \quad (3.10)$$

na qual $d_{i,j} = \|v_i - v_j\|$ para $j \neq i$, e r é o número de centros na vizinhança de v_i .

Os procedimentos citados para seleção dos valores das larguras garantem distribuições uniformes para as funções com a finalidade de promover uma boa capacidade de generalização da rede. Os pesos da camada de saída são determinados pelo método dos mínimos quadrados, no qual são empregadas as informações das saídas das funções de base radial para um conjunto de dados de entrada de treinamento da rede, e as informações desejadas na saída da rede.

As estratégias auto-organizadas são constituídas de dois estágios:

- Determinação dos valores dos centros e das larguras das funções de base radial, geralmente via métodos de agrupamento de dados;
- Cálculo dos valores dos pesos lineares da camada de saída da rede, sendo que em geral se utiliza o método dos mínimos quadrados.

Entre os métodos de agrupamento (clustering) usualmente empregados pode-se citar o K-médias, C-means e outras técnicas baseadas em lógica Fuzzy, como o Fuzzy C-means (BEZDEK, 1980), Gustafson-Kessel (GUSTAFSON; KESSEL, 1979), Gath-Geva (GATH; GEVA, 1989).

As estratégias supervisionadas utilizam procedimentos de aprendizagem que ajustam alguns ou todos os parâmetros (centros, desvios padrão e pesos) de uma determinada rede RBF. Geralmente a informação relacionada com o erro entre o valor da saída da rede e o valor desejado é utilizada no processo de treinamento. Um método usualmente utilizado refere-se ao gradiente descendente (KARAYIANNIS, 1997).

3.2.4 Definição dos parâmetros iniciais da Rede Neural de Base Radial

Como visto, é possível definir a matriz $\underline{\Sigma}_i$ de três maneiras: com desvios idênticos, desvios diferentes e com uma matriz de covariância completa. Opta-se neste trabalho pelo uso de desvios diferentes, garantindo uma melhor capacidade de representação que os desvios idênticos e evita-se problemas de *overfitting* e longos tempos de treinamento, uma vez que o número de parâmetros cresce quadraticamente com a dimensão do espaço de entrada (NELLES, 2001).

Para inicialização dos centros e das dispersões (*spreads* ou desvios-padrão) da RBF utilizou-se o algoritmo *Fuzzy C-means* (BEZDEK, 1980). O algoritmo usa a norma euclidiana existente entre um dado e os centros dos agrupamentos para inferir um determinado grau de pertinência que é inversamente proporcional a essa distância (JANG; SUN; MI-ZUTANI, 2007), realizando, assim, um processo iterativo de minimização de uma função

de custo definida por

$$J = \sum_{k=1}^n \sum_{i=1}^c \mu_{ik}^m \|x_k - v_i\|^2 \quad (3.11)$$

na qual n é o número de dados utilizados, c é o número de agrupamentos adotados, x_k é o k -ésimo dados do conjunto de treinamento utilizado, v_i é o i -ésimo centro dos agrupamentos, μ_{ik} é o grau de pertinência do k -ésimo dados no i -ésimo agrupamento e m é uma constante (tipicamente com o valor $m = 2$) (CHIU, 1994b). O centro v_i de cada agrupamento i ($i = 1, \dots, c$) pode ser calculado pela equação (3.12).

$$v_i = \frac{\sum_{k=1}^n \mu_{ik}^m x_k}{\sum_{k=1}^n \mu_{ik}^m} \quad (3.12)$$

O grau de pertinência μ_{ik} associado a um determinado agrupamento é dado por (3.13).

$$\mu_{ik} = \frac{1}{\sum_{j=1}^c \frac{\|x_k - v_i\|^{2/(m-1)}}{\|x_k - v_j\|}} \quad (3.13)$$

O algoritmo, uma vez inicializado com o número de *clusters* e o valor inicial da posição dos centros v_i , $i = 1, 2, \dots, c$, tende a convergir para um mínimo ou um ponto de sela da função de custo (BEZDEK, 1981).

O algoritmo que segue descreve os passos do processo de forma genérica (FAUSTINO, 2011):

1. Determinar os valores de c, m, ε e t_{max} ;
2. Gerar a partição *fuzzy* de forma aleatória ou com algum valor pré-determinado;
3. Atribuir valor 0 ao contador de iterações t ;
4. Atribuir $J^{(0)} = 0$;
5. Incrementar o contador de iterações t ;
6. Calcular os centros v_i segundo a equação (3.12);
7. Calcular os graus de pertinência μ_{ik} segundo a equação (3.13);
8. Calcular a função objetivo por meio da equação (3.11);
9. Calcular $\delta = J^{(t+1)} - J^{(t)}$;

10. Incrementar o contador de interações t ;
11. Se o critério de parada do algoritmo for falso, retorna ao passo 5, senão, finaliza-se o algoritmo

O critério de parada é dado quando $\delta \leq \varepsilon$ ou o número máximo de iterações (t_{max}) for alcançado.

A qualidade da solução, como a maioria dos problemas de agrupamento, depende fortemente das escolhas do número de centros e de seus valores iniciais (CHIU, 1994a). No trabalho de Yager e Filev (1992) foi sugerido um método (conhecido como *Mountain Method*) para a determinação dos números de centros e seus valores, que, apesar da qualidade dos resultados apresentados, padece de um esforço computacional elevado, uma vez que o mesmo cresce exponencialmente com a dimensão do problema. Uma forma modificada do algoritmo (conhecido como *Subtractive Clustering*) foi proposta em (CHIU, 1994a).

3.2.5 *Subtractive Clustering*

O método *Subtractive Clustering* assume cada dado como um potencial centro de agrupamento, calculando essa probabilidade baseado na densidade de dados na vizinhança. Considere um conjunto de n amostras $\{x_1, x_2, \dots, x_n\}$ em um espaço M dimensional. Sem perda de generalidade, pode-se assumir que as amostras tenham sido normalizadas em cada dimensão de forma que a faixa de cada uma seja igual, em outras palavras, as amostras sejam limitadas por um hiper-cubo. Considere cada dado como um centro de *cluster* em potencial, e que esse potencial para uma amostra x_i seja quantizado por

$$P_i = \sum_{j=1}^n e^{-\alpha \|x_i - x_j\|^2} \quad (3.14)$$

sendo

$$\alpha = \frac{4}{r_a^2}$$

e r_a seja um constante positiva. Logo, a medida do potencial de cada amostra é uma função da distancia dele com relação a todas as demais amostras. Uma amostra com muitas amostras na vizinhança tem um alto valor de potencial. A constante r_a é efetivamente o raio que define a vizinhança, dados fora desse raio tem pouca influência no potencial do ponto em questão.

Essa medida difere do proposta por Yager e Filev (1992) de duas maneiras: o potencial é associado ao dado real ao invés de um dado do *grid* como sugerido no trabalho deles; a influência de uma amostra da vizinhança cai exponencialmente com o quadrado da distância, ao invés da distância.

Após o potencial de cada amostra ser computado, seleciona-se a amostra com maior potencial como centro do primeiro *cluster*, assim, seja x_1^* a localização do centro do primeiro *cluster* e P_1^* seu valor de potencial, a partir daí revisa-se o potencial de cada amostra x_i pela formula

$$P_i = P_i - P_1^* e^{-\beta \|x_i - x_1\|^2}, \quad (3.15)$$

na qual

$$\beta = \frac{4}{r_b^2}$$

com r_b sendo uma constante positiva. Assim, subtrai-se um potencial de cada amostra dependendo da sua distância com relação ao centro do primeiro *cluster*. As amostras próximas ao centro do primeiro *cluster* terão seu potencial largamente reduzido e, portanto, dificilmente serão selecionados como centros dos próximos *clusters*. A constante r_b define o raio da vizinhança a qual o potencial será reduzido, sendo seu valor maior que r_a a fim de evitar sobreposições de centros próximos (em geral $r_b = 1,5r_a$).

Quando o potencial de todas as amostras for revisado de acordo com a equação (3.15), seleciona-se a amostra com o maior potencial como centro do segundo *cluster*. Após a obtenção do centro do k -ésimo *cluster*, revisa-se o potencial de cada amostra por

$$P_i = P_i - P_k^* e^{-\beta \|x_i - x_k\|^2}, \quad (3.16)$$

na qual x_k^* é a localização do centro do k -ésimo *cluster* e P_k^* seu valor de potencial.

No procedimento de [Yager e Filev \(1992\)](#), o processo de aquisição de novos *cluster* e a revisão dos potenciais repete-se até que

$$P_k^* < \varepsilon P_1^* \quad (3.17)$$

sendo ε um valor pequeno e sua seleção é de grande importância para o método, uma vez que um ε muito grande poucas amostras serão aceitas como centros dos *clusters*, se ε for muito pequeno, muitas amostras serão selecionadas como centros. É difícil definir um valor único para esse parâmetro que funcione com todos os padrões de dados, [Chiu \(1994a\)](#) sugere que seu valor esteja entre 0,15 e 0,5, logo um desenvolvimento adicional para aceitar ou rejeitar um centro de *cluster* foi proposto por ([CHIUI, 1994b](#)). Segue sua implementação em pseudo-código:

Algorithm 1 Algoritmo *Subtractive Clustering*

```

1: if  $P_k^* > \bar{\varepsilon}P_1^*$  then
2:   | Aceite  $x_k^*$  como um centro de cluster e continue
3: else
4:   | if  $P_k^* < \varepsilon P_1^*$  then
5:     | Rejeite  $x_k^*$  e finalize o processo de agrupamento
6:   | else
7:     | Seja  $d_{min}$  a menor distância entre  $x_k^*$  e todos os centros dos clusters
8:     | anteriormente encontrados
9:     | if  $\left(\frac{d_{min}}{r_a} + \frac{P_k^*}{P_1^*}\right) \geq 1$  then
10:    | | Aceite  $x_k^*$  como um centro de cluster e continue
11:    | | else
12:    | | Rejeite  $x_k^*$  e defina o potencial em  $x_k^*$  como 0
13:    | | Selecione a amostra com o próximo maior potencial como o novo  $x_k^*$  e reteste
14:    | | end if
15:   | end if
16: end if
    
```

O fluxograma abaixo sumariza o processo de obtenção dos *clusters* iniciais pelo *subtractive clustering*.

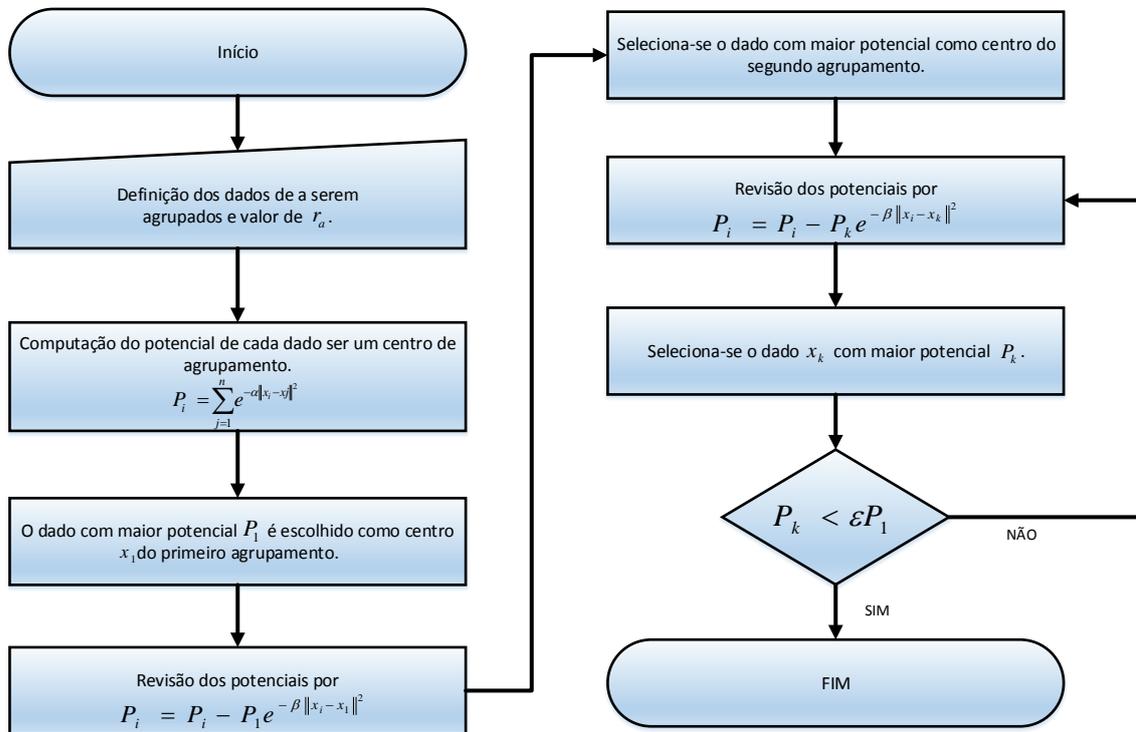


Figura 8 – Fluxograma do algoritmo *subtractive clustering*.

Com os dados dos *clusters* obtidos é possível determinar os valores dos centros das gaussianas, que são os valores centrais dos *clusters*. Os valores das larguras das funções podem ser estimados pela equação (3.18).

$$\sigma_i^* = \sqrt{\sum_{i=1}^m x_i^2 - \frac{\frac{1}{m} \left(\sum_{j=1}^m x_j \right)^2}{m-1}} \quad (3.18)$$

Os pesos da camada de saída são obtidos pelo método dos mínimos quadrados, na qual são empregadas as informações das saídas das funções de base radial para um conjunto de dados de entrada de treinamento da rede e as informações desejadas na saída da rede.

3.3 Modelos GOBF-RBF

A Figura 9 apresenta a estrutura genérica do modelo GOBF-RBF em estudo.

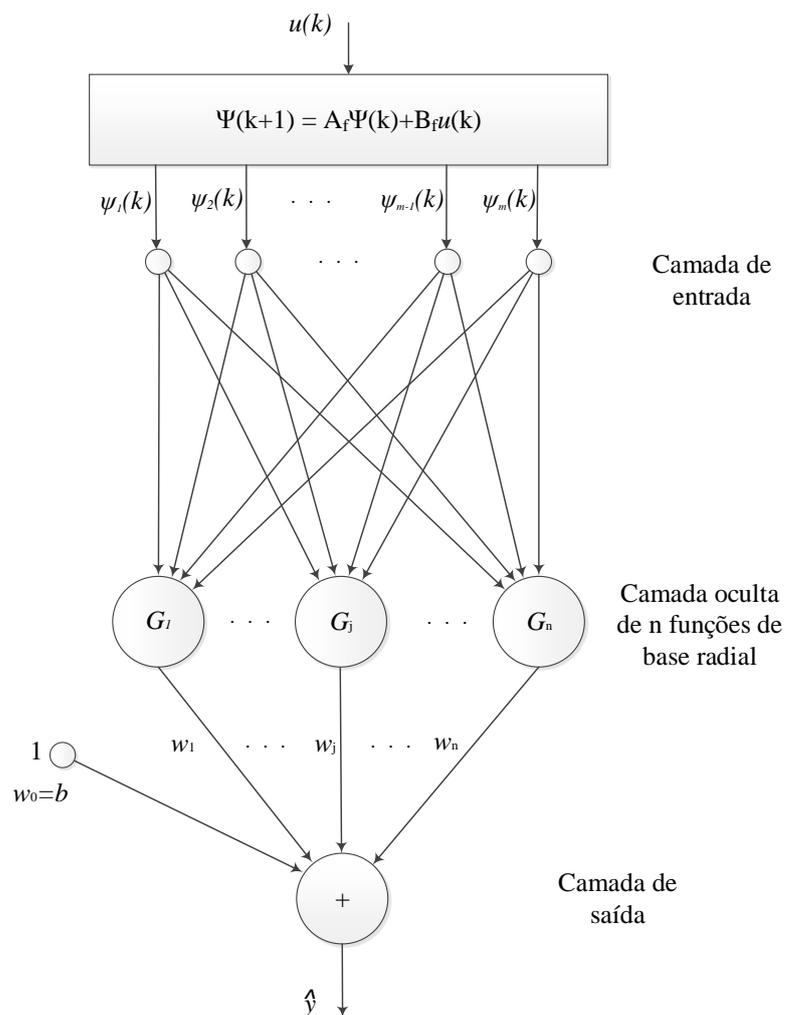


Figura 9 – Modelo GOBF-RBF.

A saída global $\hat{y}(k)$ é obtida da função de saída da rede RBF (NELLES, 2001; HAYKIN, 1999),

$$\hat{y}(k) = \sum_{i=0}^n w_i G_i(\|\Psi(k) - \underline{v}_i\|_{\underline{\Sigma}_i}), \quad (3.19)$$

na qual G_i é o valor de saída da i -ésima função de base radial como definido em (3.5), com w_i , \underline{v}_i e $\underline{\Sigma}_i$ sendo, os pesos da camada de saída, os centros das funções radiais e a matriz cuja a diagonal principal apresenta o inverso das variâncias, respectivamente, e $\Psi(k)$ o vetor de entrada com todas as saídas dos filtros das GOBFs. A proposta deste trabalho é combinar uma estrutura GOBF com uma RBF, em que a primeira modele a dinâmica linear do sistema de interesse, e a segunda estrutura mapeie as não linearidades inerentes ao processo a ser modelado.

Para determinar o modelo GOBF-RBF é necessário definir os parâmetros da rede RBF a partir de uma determinação inicial do vetor Υ que define os polos da GOBF, e consequentemente as saídas individuais dos filtros dessa. Computadas as saídas dos filtros $\Psi(k)$ para um conjunto de entradas $u(k)$, aplica-se a técnica de agrupamento apresentada juntamente com o método dos mínimos quadrados para definição dos parâmetros iniciais da RBF. Uma vez determinado o modelo GOBF-RBF inicial, uma técnica de otimização não linear é aplicada no sentido de refinar os valores dos parâmetros iniciais da modelagem em questão.

O número total de parâmetros a serem otimizados é dado por

$$\text{Número de Parâmetros} = 2.NC.NF.NP + NP + NC + 1, \quad (3.20)$$

na qual NF é o número de funções ortonormais dos modelos GOBF-RBF, NP é o número de polos das funções e NC é o número de *clusters* ou centros das funções radiais.

3.4 Otimização Não Linear dos Modelos GOBF-RBF

Com o objetivo otimizar os parâmetros dos modelos GOBF-RBF utiliza-se o algoritmo de Levenberg-Marquardt (LEVENBERG, 1944; MARQUARDT, 1963; NOCEDAL; WRIGHT, 1999; NELLES, 2001). Esse método foi selecionado tendo em vista que os modelos a serem otimizados não envolvem restrições que não sejam os limites superiores e inferiores dos parâmetros (MACHADO, 2011). O algoritmo de Levenberg-Marquardt é uma técnica padrão para resolver problemas de mínimos quadrados não lineares, ele pode ser interpretado como uma combinação de dois métodos, o do gradiente descendente e o de Gauss-Newton (NELLES, 2001): atua como o método do gradiente descendente quando a solução ótima está distante, que de forma geral é mais lento mas com convergência garantida e como o algoritmo de Gauss-Newton quando próximo de pontos ótimos,

suplantando o problema de convergência que este último apresenta quando distante da solução, garantindo, por fim, uma convergência mais rápida.

3.4.1 Ajuste dos parâmetros do modelo GOBF-RBF final

Uma vez obtido o modelo GOBF-RBF inicial pelos passos descritos anteriormente, realiza-se um ajuste fino no modelo pela otimização dos parâmetros da estrutura GOBF-RBF proposta utilizando-se o método de otimização não linear apresentado na seção anterior, minimizando-se o erro quadrático entre a saída do modelo e a saída medida ou simulada para todas as amostras consideradas. A equação (3.21) apresenta a função de custo a ser minimizada:

$$\min_{\theta} J = \frac{1}{2} \sum_{k=1}^N (\hat{y}(k) - y(k))^2, \quad (3.21)$$

com

- $\hat{y}(k)$ - saída global estimada
- $y(k)$ - saída medida do sistema
- N - número de amostras do sistema
- θ - parâmetros de otimização.

A minimização é realizada com relação aos parâmetros $\theta = [\Upsilon \ \underline{v}_i \ \underline{\Sigma}_i \ W]^T$, sendo que $\Upsilon = [\gamma_1 \ \dots \ \gamma_{n_b}]$ é o vetor com os parâmetros γ relacionados ao(s) n_b polo(s) do modelo GOBF, \underline{v}_i é o vetor com o(s) centro(s) da i -ésima função de base radial, $\underline{\Sigma}_i$ é a matriz cuja a diagonal principal contém o inverso das variâncias da i -ésima função de base radial como em (3.7), e, finalmente, W é o vetor com os pesos (incluindo o *offset*) da rede com base de funções radial proposta.

Analisa-se então os gradientes para determinar a direção de busca dos valores ótimos de cada parâmetros, sendo necessário conhecer o gradiente da função de custo com relação àquele parâmetro, para Υ tem-se

$$\nabla_{\Upsilon} J = \left[\frac{\partial J}{\partial \gamma_1} \ \dots \ \frac{\partial J}{\partial \gamma_{n_b}} \right]^T. \quad (3.22)$$

O gradiente da função de custo com relação ao parâmetro γ_l é dado por

$$\frac{\partial J}{\partial \gamma_l} = \sum_{k=1}^N (y(k) - \hat{y}(k)) \frac{\partial \hat{y}}{\partial \gamma_l}. \quad (3.23)$$

A saída do model GOBF-RBF é dada por

$$\hat{y}(k) = \sum_{i=0}^n w_i G_i(\|\Psi(k) - \underline{v}_i\| \underline{\Sigma}_i). \quad (3.24)$$

Calculando-se a derivada de (3.24) com relação a γ_l tem-se

$$\frac{\partial \hat{y}}{\partial \gamma_l} = \sum_{i=0}^n \left(w_i \frac{v_i \frac{\partial \Psi(k)}{\partial \gamma_l} - \Psi(k) \frac{\partial \Psi(k)}{\partial \gamma_l}}{\Sigma_i^2} G_i(\|\Psi(k) - \underline{v}_i\|_{\Sigma_i}) \right). \quad (3.25)$$

Os cálculos de $\frac{\partial \Psi(k)}{\partial \gamma_l}$ são dados por Machado (2011) e podem ser encontrados no apêndice A.

Para o cálculo de otimização da função de custo J com relação centros das funções de base radial tem-se

$$\frac{\partial J}{\partial v_{i,j}} = \sum_{k=1}^N (y(k) - \hat{y}(k)) \frac{\partial \hat{y}}{\partial v_{i,j}}. \quad (3.26)$$

Abrindo o vetor \underline{v}_i em (3.24) para analisar as contribuições individuais dos centros para cada entrada na função de base radial multivariada tem-se

$$\hat{y}(k) = \sum_{i=0}^n w_i \exp \left(-\frac{1}{2} \|\Psi(k) - [v_{i,1} \ v_{i,2} \ \cdots \ v_{i,m}]^T\|_{\Sigma_i}^2 \right). \quad (3.27)$$

Desenvolvendo-se a derivada de (3.27) com relação a $v_{i,j}$, vem:

$$\frac{\partial \hat{y}}{\partial v_{i,j}} = w_i \frac{(\Psi(k) - v_{i,j})}{\sigma_{i,j}^2} G_i(\|\Psi(k) - \underline{v}_i\|_{\Sigma_i}). \quad (3.28)$$

Para o cálculo de otimização da função de custo J com relação desvios-padrão das funções de base radial tem-se

$$\frac{\partial J}{\partial \sigma_{i,j}} = \sum_{k=1}^N (y(k) - \hat{y}(k)) \frac{\partial \hat{y}}{\partial \sigma_{i,j}}. \quad (3.29)$$

Reescrevendo-se (3.24) para explicitar os parâmetros $\sigma_{i,j}$ de Σ_i , com $i = 1, \dots, n$, na qual n é o número de funções de bases radiais e $j = 1, \dots, m$, com m sendo o número de entradas de cada função de base radial

$$\hat{y}(k) = \sum_{i=0}^n w_i \exp \left(-\frac{1}{2} \|\Psi(k) - \underline{v}_i\|^2 \begin{bmatrix} 1/\sigma_{i,1}^2 & 0 & \cdots & 0 \\ 0 & 1/\sigma_{i,2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/\sigma_{i,m}^2 \end{bmatrix} \right). \quad (3.30)$$

Desenvolvendo-se a derivada de (3.30) com relação a $\sigma_{i,j}$, vem:

$$\frac{\partial \hat{y}}{\partial \sigma_{i,j}} = w_i \frac{(\Psi(k) - v_{i,j})^2}{\sigma_{i,j}^3} G_i(\|\Psi(k) - \underline{v}_i\|_{\Sigma_i}). \quad (3.31)$$

Para o cálculo de otimização da função de custo J com relação aos pesos da camada de saída da rede RBF tem-se

$$\frac{\partial J}{\partial w_i} = \sum_{k=1}^N (y(k) - \hat{y}(k)) \frac{\partial \hat{y}}{\partial w_i}. \quad (3.32)$$

Calculando-se a derivada de (3.24) com relação a w_i , tem-se

$$\frac{\partial \hat{y}}{\partial w_i} = G_i(\|\Psi(k) - v_i\|_{\Sigma_i}) \quad \text{com } G_0(\cdot) = 1. \quad (3.33)$$

3.4.2 Algoritmo de otimização não linear dos modelos GOBF-RBF

Para a obtenção dos modelos GOBF-RBF dos dados de entrada e saída de sistemas dinâmicos não lineares uma sequência de passos foi proposta. O algoritmo que segue descreve esses passos:

1. Dado um conjunto de dados de entradas e saídas do sistema, determine os valores iniciais e o número de parâmetros γ_l (de maneira aleatória ou baseado em conhecimentos prévios do sistema), e calcule as saídas Ψ dos filtros ortonormais;
2. Dadas as saídas dos filtros e as saídas do sistema a ser modelado, aplica-se o método *subtractive clustering* para determinar o número de *clusters* e seus valores centrais iniciais para a aplicação do método de clusterização *Fuzzy C-means*, obtendo-se por fim os valores finais dos centros dos *clusters* e o grau de pertinência relacionado a cada dado, que indica a qual agrupamento ele pertence;
3. De posse dos *clusters*, determinam-se os desvios-padrão por (3.18) e por fim os pesos de saída da rede de base radial são obtidos pelo método dos mínimos quadrados, obtendo-se assim a estrutura inicial do sistema a ser modelado;
4. Calcular o gradiente $\nabla_{\theta} J = \left[\nabla_{\Upsilon}^T J \quad \nabla_{v_i}^T J \quad \nabla_{\Sigma_i}^T J \quad \nabla_W^T J \right]$ utilizando as equações (3.25), (3.28), (3.31) e (3.33), nas quais Υ é o vetor com os parâmetros relativos ao(s) polo(s) do modelo, v_i é o vetor que contém os centros de uma função radial, Σ_i é a matriz cuja a diagonal principal possui os valores das variâncias como definido em (3.7) e W é o vetor de pesos de saída da rede de base radial;
5. Use os valores de $\nabla_{\theta} J$ no algoritmo de otimização não linear para atualizar o valor dos parâmetros θ , incluindo o intervalo de factibilidade dos parâmetros γ do modelo GOBF;
6. Volte ao passo 4 até que um critério de parada seja atingido, seja ele o número máximo de passos do algoritmo ou a exatidão mínima alcançada.

Os passos descritos anteriormente podem ser sumarizados no fluxograma que segue abaixo.

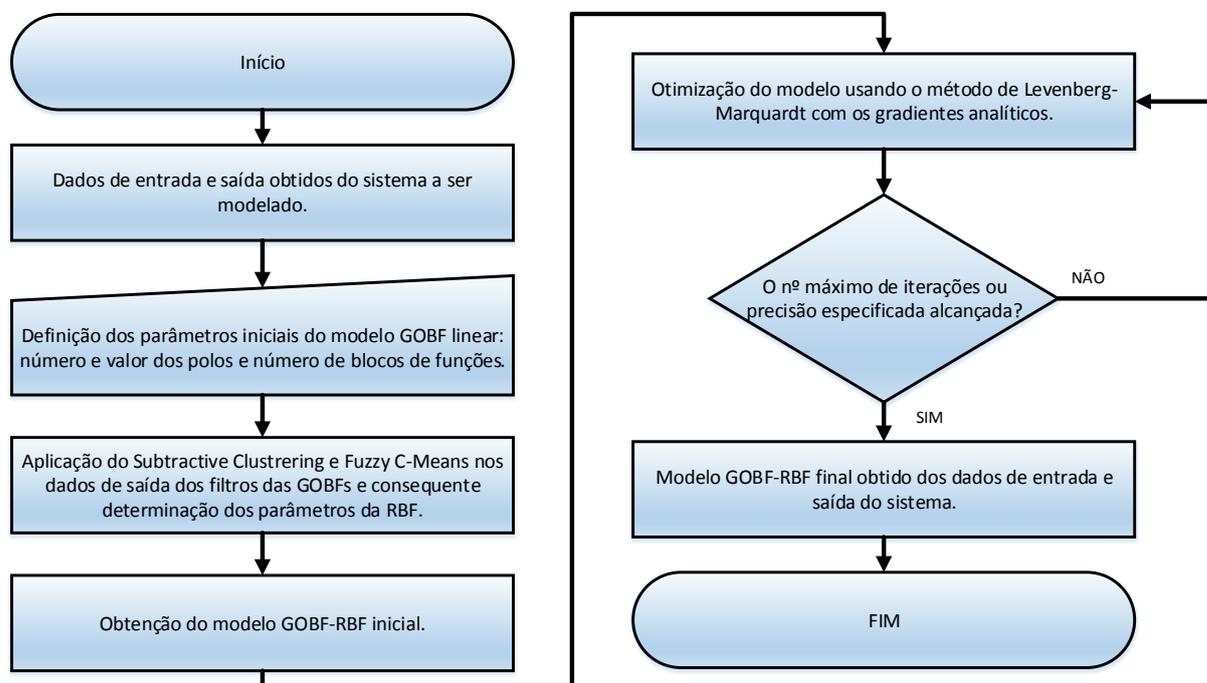


Figura 10 – Fluxograma do processo de otimização não linear.

3.5 Otimização dos Modelos GOBF-RBF por Enxame de Partículas

Otimização por enxame de partículas (*Particle Swarm Optimization* - PSO) é uma técnica de otimização pertencente a classe de algoritmos metaheurísticos conhecidos como *Swarm Intelligence* (SI). Um sistema SI típico é composto basicamente por dois elementos, uma população que interage localmente entre si e o ambiente em que essa população está inserida. O comportamento dessa população é ditado por regras simples, não havendo uma estrutura de controle centralizada comandando-o. A interação entre os agentes é, em geral, local, apresentando ainda um certo grau de aleatoriedade. Ainda assim, há um comportamento global inteligente que surge devido as interações locais dos indivíduos, que “aprendem” com comportamento dos seus pares.

O PSO surgiu como uma técnica de otimização baseada em populações desenvolvida por Kennedy e Eberhart (1995). Sua motivação principal é o comportamento social de bandos de pássaros e cardume de peixes, compartilhando similaridades diversas com técnicas baseadas em algoritmos evolutivos como os algoritmos genéticos (*Genetic Algorithms* - GA) (HOLLAND, 1975), porém não apresenta operadores evolutivos como *crossover* e *mutação* (BACK; FOGEL; MICHALEWICZ, 2000).

A partir de uma população inicial definida aleatoriamente, o processo de otimização do PSO é realizado pela atualização de diversas propriedades desses indivíduos a cada iteração. As potenciais soluções, conhecidas como partículas, se movem pelo universo de busca seguindo sua própria experiência e a melhor solução encontrada, até aquela geração, pelas demais partículas.

Uma versão do PSO que tem ganhado atenção nos últimos anos é a chamada *quantum-behaved particle swarm optimization* (QPSO) (SUN; FENG; XU, 2004). Esse algoritmo baseia-se na mecânica quântica e na análise da trajetória do PSO (CLERC; KENNEDY, 2002). A análise da trajetória das partículas do PSO mostraram que essas oscilam e convergem para um atrator local, ou seja, o comportamento dinâmico de cada partícula do PSO é limitado, limitando também seu o espaço de busca. No QPSO, as partículas assumem um comportamento quântico, sendo atraídas por um potencial quântico, centrado no seu atrator local (SUN; FENG; XU, 2004).

O QPSO tem sido usado com sucesso em problemas de otimização de diversas áreas da ciência como: otimização multiobjetivo (OMKAR *et al.*, 2009), treinamento de redes neurais (LI *et al.*, 2007), identificação de sistemas (GAO; Z.-Q.; TONG, 2008), planejamento de trajetória em sistemas autônomos (TAO *et al.*, 2017), entre outras.

Como segunda proposta para a definição dos parâmetros da estrutura GOBF-RBF em estudo, propõem-se o uso dessa classe de algoritmos. Uma das vantagens que pode ser apontada é o fato de não ser necessário muitos conhecimentos acerca do sistema a ser otimizado (a função objetivo e a lista de parâmetros, bem como seus limites, são informação suficiente para o algoritmo). Os parâmetros a serem otimizados são os polos (parâmetros γ) das GOBF, os centros e as dispersões ou desvios-padrão. Os pesos de saída da RBF podem ser obtidos diretamente pelo método dos mínimos quadrados.

3.5.1 Algoritmo PSO

Suponha um universo com M partículas, cada uma com volume infinitesimal e com apenas três propriedades: seu vetor de posição atual, seu vetor de velocidade e o vetor de melhor posição individual. Supondo que cada partícula possua N propriedades, esses vetores também serão N -dimensionais. Logo, na n -ésima iteração, os vetores de propriedades da partícula i ($1 \leq i \leq M$) são

1. O vetor de posição atual $X_{i,n} = (X_{i,n}^1, X_{i,n}^2, \dots, X_{i,n}^j, \dots, X_{i,n}^N)$, no qual cada componente do vetor representa uma variável de decisão do problema e $1 \leq j \leq N$;
2. O vetor de velocidade $V_{i,n} = (V_{i,n}^1, V_{i,n}^2, \dots, V_{i,n}^j, \dots, V_{i,n}^N)$, que denota o incremento da posição atual e $1 \leq j \leq N$;

3. O vetor de melhor posição individual $P_{i,n} = (P_{i,n}^1, P_{i,n}^2, \dots, P_{i,n}^j, \dots, P_{i,n}^N)$, que denota o incremento da posição atual e $1 \leq j \leq N$ (chamado de *pbest* no trabalho original de Kennedy e Eberhart (1995));

O valor inicial de $X_{i,0}$ é geralmente definido aleatoriamente dentro do universo de excursão das partículas, definido por $[X_{min}^j, X_{max}^j](1 \leq j \leq N)$, na qual X_{min}^j e X_{max}^j são, respectivamente, os limites inferior e superior da posição das partículas na j -ésima dimensão. Da mesma forma, os vetores de velocidade $V_{i,0}$ podem ser inicializados de maneira aleatória escolhendo-se suas componentes dentro no intervalo $[-V_{max}^j, V_{max}^j](1 \leq j \leq N)$, sendo V_{max}^j o limite superior das velocidades na j -ésima dimensão. E $P_{i,n}$ pode ser inicializado com o vetor de posição inicial.

Durante a n -ésima iteração, avalia-se a função objetivo para o vetor de melhor posição individual de cada partícula. O melhor resultado é armazenado em $P_{g,n}$ (conhecido com *gBest* no algoritmo original de Kennedy e Eberhart (1995)), no qual g denota o indexador da melhor partícula. Esse valor é armazenado no vetor de melhor posição global ($G_n = G_n^1, G_n^2, \dots, G_n^N$). A atualização de $P_{i,n}$ é dada por

$$P_{i,n} = \begin{cases} X_{i,n} & \text{se } f(X_{i,n}) < f(P_{i,n-1}), \\ P_{i,n-1} & \text{se } f(X_{i,n}) \geq f(P_{i,n-1}) \end{cases} \quad (3.34)$$

logo, G_n é definido por

$$G_n = P_{g,n}, \text{ onde } g = \arg \min_{(1 \leq i \leq M)} [f(P_{i,n})]. \quad (3.35)$$

As velocidades são atualizadas por

$$V_{i,n+1}^j = V_{i,n}^j + c_1 r_{i,n}^j (P_{i,n}^j - X_{i,n}^j) + c_2 R_{i,n}^j (G_n^j - X_{i,n}^j), \quad (3.36)$$

e as posições por

$$X_{i,n+1}^j = X_{i,n}^j + V_{i,n+1}^j \quad (3.37)$$

nas quais, $i = 1, 2, \dots, M$, $j = 1, 2, \dots, N$, c_1 representa o parâmetro de aceleração cognitivo e c_2 representa o parâmetro de aceleração social, no algoritmo PSO original, são escolhidos como 2. Os parâmetros $r_{i,n}^j$ e $R_{i,n}^j$ são duas sequências diferentes de números aleatórios distribuídos uniformemente entre $[0, 1]$. Cada componente do vetor de velocidades $V_{i,n}^j$ é normalmente restrita ao intervalo $[-V_{max}^j, V_{max}^j]$.

Ao longo dos anos, várias implementações alternativas ao PSO original foram propostas. A primeira revisão do algoritmo, conhecida como PSO com fator de inércia (PSO-In), foi proposta por Shi e Eberhart (1998). Nessa versão, a velocidade das partículas é atualizada de acordo com

$$V_{i,n+1}^j = w V_{i,n}^j + c_1 r_{i,n}^j (P_{i,n}^j - X_{i,n}^j) + c_2 R_{i,n}^j (G_n^j - X_{i,n}^j) \quad (3.38)$$

na qual w é chamado de fator de inércia e seu valor determina o comportamento do algoritmo. Para valores pequenos ($w < 0,8$), o PSO se comporta como um algoritmo de busca local, com rápida convergência. Para $w > 1,2$, o PSO tende a ter um comportamento de busca global, ao custo de um maior número de interações e a possibilidade de que o mesmo não convirja. Uma alternativa é a variação linear desse parâmetro de acordo com o número de iterações,

$$w = w_{max} - \frac{(w_{max} - w_{min})}{n_{max}}n, \quad (3.39)$$

no qual w_{max} e w_{min} correspondem aos valores máximos e mínimos, respectivamente, do fator de inércia, n_{max} é o número máximo de iterações e n é a iteração atual.

Outra versão modificada foi apresentada por Clerc (1999). Essa modificação insere um fator de constrição χ para limitar o ajuste de velocidades das partículas. Nessa versão o vetor de velocidade é dado por

$$V_{i,n+1}^j = \chi[V_{i,n}^j + c_1r_{i,n}^j(P_{i,n}^j - X_{i,n}^j) + c_2R_{i,n}^j(G_n^j - X_{i,n}^j)] \quad (3.40)$$

na qual χ é definido por

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \quad (3.41)$$

de maneira que $\varphi = c_1 + c_2$ e $\varphi > 4$. Essa versão é normalmente conhecida como otimização por enxame de partículas com constrição (PSO-Co).

O algoritmo 2 apresenta o pseudocódigo para as três formulações descritas acima.

3.5.2 Algoritmo QPSO

A análise da trajetória das partículas nos algoritmos PSO foi estudada por Clerc e Kennedy (2002). Foi provado que, considerando os melhores locais e globais como posições fixas e os números aleatórios constantes, o comportamento do algoritmo é linear. O sistema tende a convergir para melhor ponto global em $t \rightarrow \infty$.

Contudo, o comportamento social de organismos é muito mais complexo do que seria possível emular em um algoritmo linear, na prática o comportamento da evolução do pensamento humano, por exemplo, se comporta de maneira incerta, e com certo nível de abstração, se aproxima de partículas com comportamento quântico (SUN; XU; FENG, 2004).

Baseando-se nessa análise, Sun, Feng e Xu (2004) introduziram a teoria quântica no PSO e propuseram um algoritmo PSO no qual as partículas tem comportamento quântico (*Quantum-behaved PSO* - QPSO)

O modelo quântico do PSO pode sobrepujar alguns dos problemas comuns relacionados ao uso do PSO convencional (SUN; XU; FENG, 2004):

Algorithm 2 Algoritmo PSO

```

1: Inicialize a posição atual  $X_{i,0}^j$ , a velocidade  $V_{i,0}^j$  e a melhor posição individual  $P_{i,0}^j$ 
   (fazendo  $P_{i,0}^j = X_{i,0}^j$ ) para cada partícula, avalie a função objetivo para essa população,
   e encontre a melhor posição global  $G_0$ 
2:  $n = 0$ 
3: while A condição de saída não for cumprida do
4:    $n = n + 1$ 
5:   Selecione um valor para  $w$  (para o PSO-In)
6:   for  $i = 1$  até  $M$  do
7:     for  $j = 1$  até  $N$  do
8:        $V_{i,n+1}^j = V_{i,n}^j + c_1 r_{i,n}^j (P_{i,n}^j - X_{i,n}^j) + c_2 R_{i,n}^j (G_n^j - X_{i,n}^j)$  (para o PSO original)
9:       (ou  $V_{i,n+1}^j = w V_{i,n}^j + c_1 r_{i,n}^j (P_{i,n}^j - X_{i,n}^j) + c_2 R_{i,n}^j (G_n^j - X_{i,n}^j)$  (para o PSO-In))
10:      (ou  $V_{i,n+1}^j = \chi [V_{i,n}^j + c_1 r_{i,n}^j (P_{i,n}^j - X_{i,n}^j) + c_2 R_{i,n}^j (G_n^j - X_{i,n}^j)]$  (para o PSO-Co))
11:      if  $V_{i,n+1}^j > V_{max}$  then
12:         $V_{i,n+1}^j = V_{max}$ 
13:      end if
14:      if  $V_{i,n+1}^j < -V_{max}$  then
15:         $V_{i,n+1}^j = -V_{max}$ 
16:      end if
17:       $X_{i,n+1}^j = X_{i,n}^j + V_{i,n+1}^j$ 
18:    end for
19:    Avalie a função objetivo para  $X_{i,n+1}$ 
20:    Atualize  $P_{i,n}$  e  $G_n$ 
21:  end for
22: end while
    
```

- O sistema quântico é um sistema não linear complexo, no qual o princípio da superposição de estados é válido, de maneira que sistemas quânticos possuem muito mais estados que um sistema linear;
- Os sistemas quânticos são baseados no princípio da incerteza, logo, diferem dos sistemas estocásticos clássicos. Uma partícula quântica não tem trajetória definida, pode aparecer em qualquer posição com certa distribuição de probabilidade;
- As partículas nos algoritmos PSO convencionais estão limitadas a um espaço de busca, garantindo que todo enxame não divirja, convergindo para um ótimo local ou global. No QPSO, as partículas podem aparecer em qualquer posição do espaço de busca, mesmo que essa seja bem distante do melhor global atual, com certa probabilidade.

No modelo quântico do PSO, o estado de uma partícula pode ser definido por uma função de onda $\Psi(X_t)$, ao invés de sua posição e velocidade. Assim, a probabilidade de uma determinada partícula aparecer na posição X da função densidade de probabilidade $|\Psi(X, t)|$, depende do campo potencial no qual a partícula se encontra (LIU; SUN; XU, 2006).

O procedimento de funcionamento do algoritmo QPSO é similar ao PSO, mas possui equações de evolução diferentes, uma vez que não há um vetor de velocidades para cada partícula. Os desenvolvimentos formais do procedimento, suas fundamentações e provas encontram-se detalhadas em (SUN; LAI; WU, 2012). As posições das partículas são definidas por

$$X_{i,n+1}^j = \begin{cases} p_{i,n}^j + \alpha |X_{i,n}^j - p_{i,n}^j| \ln 1/u_{i,n+1}^j & \text{se } k < 0,5 \\ p_{i,n}^j - \alpha |X_{i,n}^j - p_{i,n}^j| \ln 1/u_{i,n+1}^j & \text{se } k \geq 0,5 \end{cases} \quad (3.42)$$

na qual

$$p_{i,n}^j = \varphi_{i,n}^j P_{i,n}^j + (1 - \varphi_{i,n}^j) G_n^j \quad (3.43)$$

é o atrator da partícula $X_{i,n}^j$, α é o coeficiente de contração-expansão, k , φ e u são números aleatórios distribuídos uniformemente no intervalo $[0, 1]$.

O algoritmo encontra-se sumarizado no pseudocódigo abaixo.

Algorithm 3 Algoritmo QPSO

```

1: Inicialize as posições atuais  $X_{i,0}^j$  e as melhores posições individuais  $P_{i,0}^j$  de cada partícula, avalie a função objetivo e encontre a melhor posição global  $G_0$ 
2:  $n = 0$ 
3: while A condição de saída não for cumprida do
4:     Seleccione um valor para  $\alpha$ 
5:     for  $i = 1$  até  $M$  do
6:         Avalie a função objetivo  $f(X_{i,n})$ 
7:         Atualize  $P_{i,n}$  e  $G_n$ 
8:         for  $j = 1$  até  $N$  do
9:              $\varphi_{i,n}^j =$  número aleatório  $[0, 1]$ 
10:             $p_{i,n}^j = \varphi_{i,n}^j P_{i,n}^j + (1 - \varphi_{i,n}^j) G_n^j$ 
11:             $u_{i,n+1} =$  número aleatório  $[0, 1]$ 
12:             $k =$  número aleatório  $[0, 1]$ 
13:            if  $k < 0,5$  then
14:                 $X_{i,n+1}^j = p_{i,n}^j + \alpha |X_{i,n}^j - p_{i,n}^j| \ln 1/u_{i,n+1}^j$ 
15:            else
16:                 $X_{i,n+1}^j = p_{i,n}^j - \alpha |X_{i,n}^j - p_{i,n}^j| \ln 1/u_{i,n+1}^j$ 
17:            end if
18:        end for
19:    end for
20:     $n = n + 1$ 
21: end while
    
```

Sempre que aplicando algoritmos para a solução de problemas, uma das etapas mais críticas é a seleção dos parâmetros desse algoritmo. Fica claro que, para o algoritmo QPSO proposto, além do tamanho da população, α é o parâmetro mais crítico.

Sun *et al.* (2012) provaram que selecionando $\alpha \leq e^\zeta$, na qual ζ é a constante *Euler-Mascheroni* ($\zeta \approx 0,5772156649$), é possível prevenir “explosões” populacionais e garantir a convergência do enxame. Mas não há garantia de que essa escolha seja capaz de fornecer uma performance satisfatória ao QPSO em aplicações práticas.

Quando aplicando o QPSO em problemas práticos, pode-se escolher um valor fixo para o parâmetro α ou decrementa-lo linearmente durante a progressão do algoritmo. Esse decremento é ajustado por

$$\alpha = \frac{(\alpha_1 - \alpha_2)(n_{max} - n)}{n_{max}} + \alpha_2, \quad (3.44)$$

na qual α_1 e α_2 ($\alpha_1 > \alpha_2$) são os valores inicial e final de α , respectivamente, n é o número da iteração atual e n_{max} é o número máximo de iterações. Sun *et al.* (2012) realizaram estudos empíricos variando os parâmetros de seleção α e aplicaram o algoritmo na função de Rosenbrock (unimodal), Rastrigin (multimodal) e Griewank (multimodal), obtendo, para o algoritmo QPSO descrito, o melhor resultado quando α varia linearmente de $1,0 \rightarrow 0,8$.

3.5.3 Algoritmo de otimização não linear dos modelos GOBF-RBF

De forma semelhante à proposta de otimização dos parâmetros via técnicas de otimização não lineares, apresenta-se abaixo a sequência de passos para esse refinamento usando o PSO ou o QPSO:

1. Dado um conjunto de dados de entradas e saídas do sistema, determine os valores iniciais e o número de parâmetros γ_l (de maneira aleatória ou baseado em conhecimentos prévios do sistema) e calcule as saídas Ψ dos filtros ortonormais;
2. Dadas as saídas dos filtros e as saídas do sistema a ser modelado, aplica-se o método *subtractive clustering* para determinar o número de *clusters* e seus valores centrais iniciais para a aplicação do método de clusterização *Fuzzy C-means*, obtendo-se por fim os valores finais dos centros dos *clusters* e o grau de pertinência relacionado a cada dado, que indica a qual agrupamento ele pertence;
3. De posse dos *clusters*, determinam-se os desvios-padrão por (3.18) e por fim os pesos de saída da rede de base radial são obtidos pelo método dos mínimos quadrados, obtendo-se assim a estrutura inicial do sistema a ser modelado;
4. Defina os parâmetros do algoritmo de otimização. Caso utilizando o PSO: c_1 , c_2 , V_{max} , X_{min}^j , X_{max}^j e w (PSO-In); caso utilizando o QPSO: α_1 e α_2 ou apenas α (caso esteja usando um valor fixo); e as condições de parada para ambos os casos, o número máximo de iterações n e a precisão mínima para termino;

5. Aplique o algoritmo (PSO ou QPSO), levando em consideração o intervalo de factibilidade dos parâmetros γ do modelo GOBF;
6. Volte ao passo 4 até que um critério de parada seja atingido, seja ele o número máximo de passos do algoritmo ou a exatidão mínima alcançada.

Fica claro que o processo de obtenção da estrutura e valores iniciais do modelo é o mesmo para as duas propostas, diferenciando-se apenas na técnica aplicada para o refinamento de seus parâmetros.

4 Resultados

4.1 Introdução

Com o objetivo de demonstrar os resultados da metodologia proposta para a modelagem de sistemas dinâmicos não lineares, utilizando tanto o processo de otimização não linear quanto o algoritmo PSO, esta seção apresenta exemplos de aplicações dos modelos GOBF-RBF em sistemas reais e simulados.

4.2 Exemplos de aplicação

Uma vez que não se tem conhecimento prévio sobre a dinâmica dos sistemas a serem modelados, empregaram-se GOBFs com diferentes combinações de número de polos e de funções, a fim de se encontrar, dentro dos modelos candidatos obtidos, o que melhor represente o sistema com a abordagem proposta.

O processo de identificação se dá como apresentado anteriormente, iniciando com a determinação dos parâmetros γ das GOBF e as saídas dos filtros e conseguinte obtenção do número de centros e suas posições iniciais pelo algoritmo *subtractive clustering*. Aplica-se o método *Fuzzy C-means* para o agrupamento dos dados de entrada e saída da rede neural, definindo-se então os centros, desvios padrão e pesos de saída. Após a obtenção do modelo inicial, procede-se para a etapa de refinamento utilizando a técnica de otimização não linear com os gradientes do modelo. Posteriormente aplica-se a segunda abordagem utilizando o QPSO e os modelos resultantes e a resposta do algoritmo são analisados.

Para todos os exemplos, foi realizada uma etapa de normalização para garantir que valores de maior amplitude não exerçam influência sobre a convergência dos algoritmos (LIMA; PINHEIRO; SANTOS, 2014). Aplicando-se a equação (4.1), garante-se que os conjuntos de dados estejam na faixa compreendida entre $[0, 1]$.

$$Z^*(t) = \frac{Z(t) - \min(Z)}{\max(Z) - \min(Z)}. \quad (4.1)$$

Apresenta-se ainda, para cada exemplo, uma análise dos vários modelos obtidos do ponto de vista da parcimoniedade. Uma vez que o número de parâmetros dessa estrutura pode ser elevado.

Finalmente, simulações de Monte-Carlo são desenvolvidas e seus resultados apresentados. Essa análise tem por objetivo validar a capacidade de convergência da metodologia proposta frente a inicializações aleatórias dos polos.

4.2.1 Modelo de um sistema de nível

O exemplo que segue faz a modelagem e identificação de um processo real em escala laboratorial de transferência de líquido entre dois tanques por uma bomba, como ilustrado na Figura 11. O processo é descrito por [Lindskog \(1996\)](#), [Hellendoorn e Driankov \(1997\)](#). A saída y é o nível de água no tanque (centímetros) e a entrada u é a tensão (Volts) aplicada ao sistema que bombeia a água para o tanque superior.

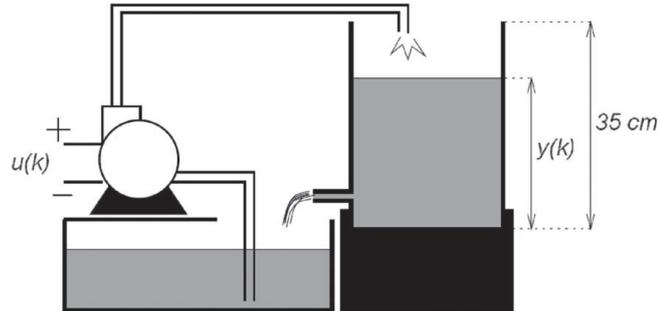


Figura 11 – Sistema de Nível.

A vazão de entrada (q_{in}) de um determinado fluido no processo de nível é definida por

$$q_{in} = kV(t)$$

na qual $V(t)$ é a tensão de acionamento aplicada a servo-bomba do processo, k é uma constante de proporcionalidade relacionada com a tensão $u(t)$ de comando do *driver* da bomba de recalque.

A vazão de saída (q_{out}) do sistema é dada por

$$q_{out} = a_{out}\sqrt{2gy(t)}$$

na qual a_{out} é a área da tubulação de saída, g é a aceleração da gravidade e $y(t)$ é o nível do líquido no tanque no instante t . Assim, a equação que define o nível do tanque em um instante de tempo t é dada por

$$y(t) = y(0) + \frac{1}{A} \int_0^t (q_{in}(\tau) - q_{out}(\tau)) d\tau \quad (4.2)$$

na qual A é a área transversal do reservatório. O sistema em estudo é considerado não linear uma vez que sua vazão de saída é função de \sqrt{y} e a saturação do nível mínimo em y se dá devido a diferença de altura entre a posição do orifício de escoamento e a base do tanque.

Dois conjuntos de dados foram obtidos por [Lindskog \(1996\)](#), um para identificação e outro para validação, cada um contendo 1000 pares de dados de entrada e saída amostrados a uma taxa de 1 segundo. Os dados utilizados no processo de identificação podem ser observados na Figura 12.

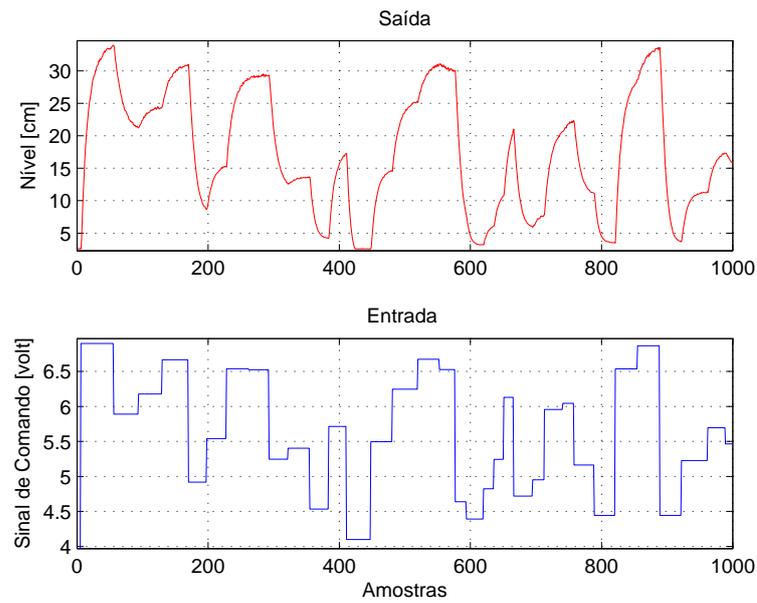


Figura 12 – Dados de identificação do sistema de nível.

Os dados utilizados no processo de validação podem ser observados na Figura 13

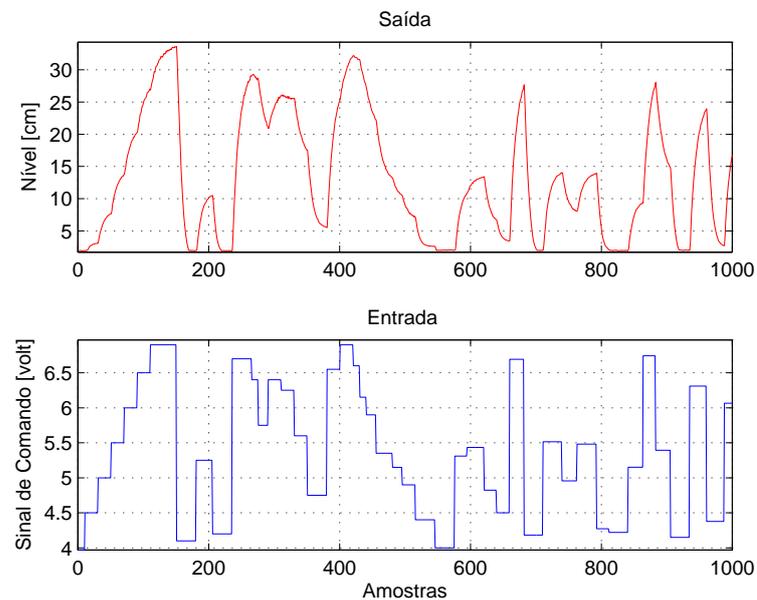


Figura 13 – Dados de validação do sistema de nível.

4.2.1.1 Otimização não linear

A Tabela 1 apresenta os resultados para o melhor modelo encontrado, trazendo também os dados dos parâmetros γ inicial (antes da aplicação da otimização não linear) e final, bem como o valor do erro quadrático médio (EQM) para o modelo utilizando os dados de validação.

Tabela 1 – Modelo com melhor EQM para o sistema de nível líquido.

Υ inicial	Υ final	EQM
$\Upsilon = [0, 5000]$	$\Upsilon = [0, 8167]$	0,3089

O valor do polo inicial associado ao valor de Υ inicial é $p_1 = 0,5000$ e o valor do polo final, referente ao vetor Υ apresentado na Tabela 1 após o processo de otimização é $p_1 = 0,8167$. São apresentados na Tabela 2 o número de funções ortonormais (NF) dos modelos GOBF-RBF, o número de polos (NP) das funções e o número de *clusters* ou centros das funções radiais (NC) e o número total de parâmetros otimizados (TP).

Tabela 2 – Informações do modelo GOBF-RBF para o sistema de nível líquido.

NF	NP	NC	TP
8	1	4	70

A estrutura do modelo proposto para o sistema em estudo pode ser dado, de forma semelhante ao da Figura 9, pela Figura 14.

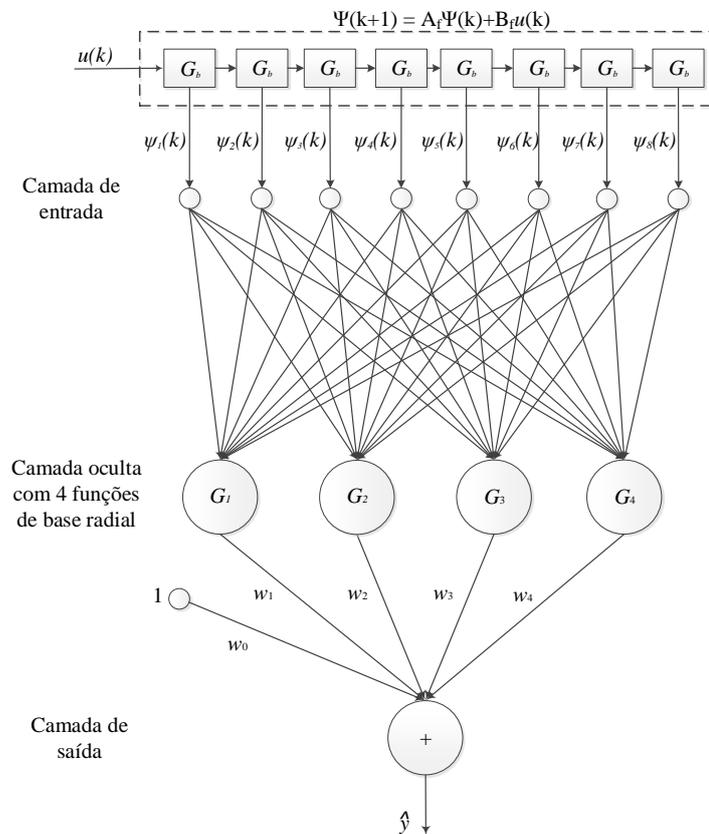


Figura 14 – Estrutura do modelo GOBF-RBF para o sistema de nível líquido.

A Figura 15 apresenta a saída do modelo GOBF-RBF apresentado nas tabelas anteriores para os dados de validação.

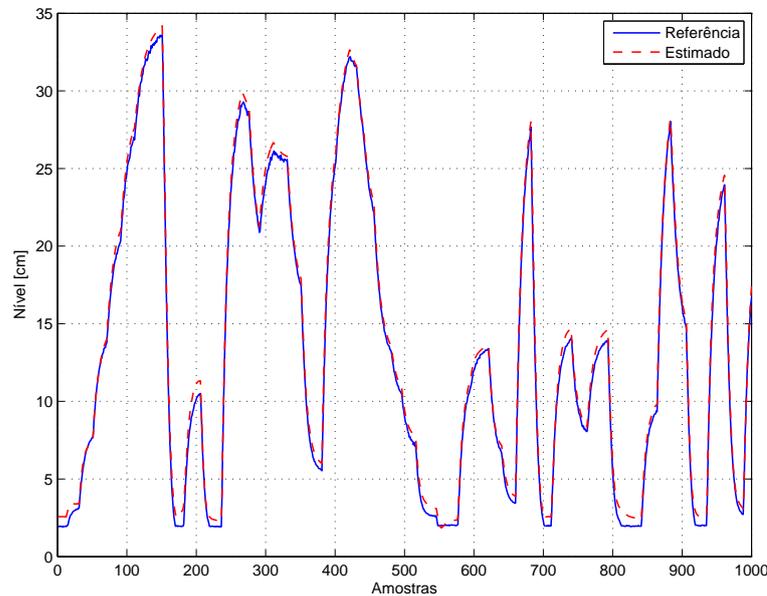


Figura 15 – Resposta do modelo selecionado pelo EQM para os dados de validação do sistema de nível utilizando a otimização não linear.

A tabela 3 apresenta os resultados obtidos para o sistema de nível líquido obtido por Machado (2011) para outras estruturas que também utilizam GOBFs.

Tabela 3 – Comparação da resposta do modelo do nível líquido com outras estruturas.

Tipo de modelo	EQM
GOBF-RBF	0,3089
Volterra-GOBF (MACHADO, 2011)	0,8877
Volterra-GOBF Direções Independentes (MACHADO, 2011)	0,7793

Como pode ser observado pela Figura 15 e pela análise do EQM do modelo obtido para os dados de validação, a metodologia proposta foi capaz de obter um modelo para representar o sistema não linear de nível líquido com uma boa exatidão.

Nota-se, no entanto, que a quantidade de parâmetros do modelo é muito elevada. O que pode ser problemático em algumas aplicações. Porém, observando as respostas dos demais modelos obtidos (Tabela 4), existem modelos de boa qualidade que poderiam acabar sendo descartados caso o critério de escolha se baseasse unicamente no EQM.

Com isso em mente, faz-se uma análise do ponto de vista dos chamados critérios de informação (mais detalhes podem ser encontradas no Apêndice B). Calcula-se o critério de informação de Akaike (AIC) e de Bayes (BIC) para uma análise mais aprofundada das estruturas obtidas. O modelo com menor valor de critério de informação seria o mais

indicado para representar o sistema tendo em vista sua exatidão e parcimoniedade. Os resultados são apresentados na Tabela 4.

Tabela 4 – Modelos GOBF-RBF para o sistema de nível.

NF	NP	NC	TP	EQM	AIC	BIC
3	1	3	23	0.3614	$-9.7180 \cdot 10^2$	$-8.5792 \cdot 10^2$
4	1	3	29	0.3720	$-9.3091 \cdot 10^2$	$-7.8858 \cdot 10^2$
6	1	3	41	0.3714	$-9.0858 \cdot 10^2$	$-7.0736 \cdot 10^2$
8	1	4	70	0.3089	$-1.0347 \cdot 10^3$	$-6.9115 \cdot 10^2$
2	2	3	30	0.3445	$-1.0055 \cdot 10^3$	$-8.5730 \cdot 10^2$

Como observado na Tabela 4, a escolha do modelo pelo EQM diverge do apontado pelo BIC, mas é o mesmo escolhido pelo AIC. A resposta do modelo indicado pelo BIC para os dados de validação podem ser observados na Figura 16.

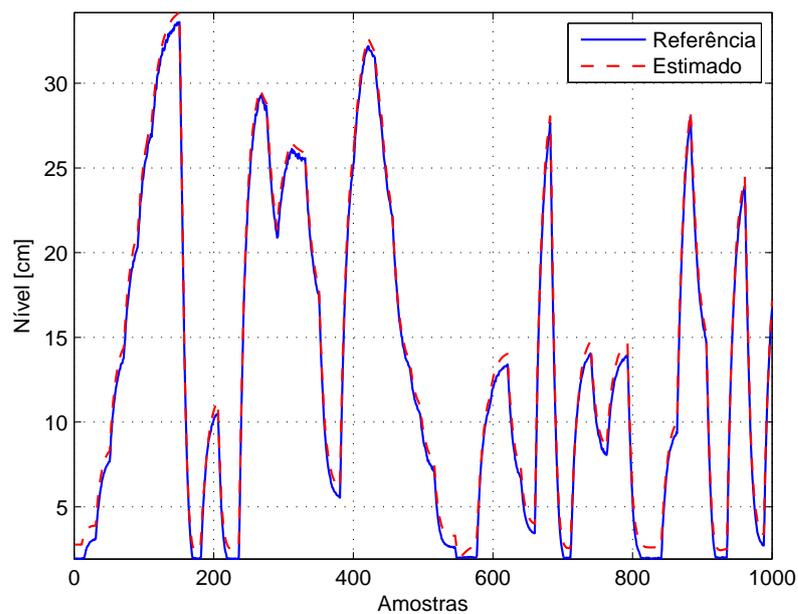


Figura 16 – Resposta do modelo selecionado pelo BIC para os dados de validação do sistema de nível utilizando a otimização não linear.

A resposta do modelo com menor BIC, apesar de maior valor de EQM, representa o sistema satisfatoriamente com menos da metade do número de parâmetros. Vale salientar que o uso dos critérios de informação não é a única abordagem para a escolha de estruturas (NELLES, 2001; AGUIRRE, 2007b), e que para um mesmo conjunto de modelos é possível que critérios de informação diferentes não concordem.

4.2.1.2 Otimização pelo QPSO

O procedimento de otimização pelo QPSO foi realizado nas duas estruturas selecionadas na seção anterior (a de melhor EQM e de menor BIC). É importante frisar que, mesmo variando os parâmetros e aumentando o número máximo de iterações para valores acima de 10000, o algoritmo PSO convencional (incluindo os modelos alternativos apresentados no capítulo anterior) não foi capaz de obter resultados satisfatórios. Uma das possíveis causas é o grande número de parâmetros a serem otimizados e o fato das funções objetivo serem multimodais. Porém, o algoritmo QPSO foi capaz de encontrar resultados satisfatórios, sem a necessidade de recorrer para estruturas *multi-swarms* (BLACKWELL; BRANKE, 2004) ou técnicas multiobjetivo (COELLO; LECHUNGA, 2002).

Os parâmetros utilizados no algoritmo encontram-se na Tabela 5.

Tabela 5 – Parâmetros do QPSO para o sistema de nível.

n° de partículas	n° máx. de passos	α	precisão
50	2000	1,2 \rightarrow 0,8	10 ⁻⁵

A Tabela 6 apresenta o EQM, AIC e BIC para esses modelos.

Tabela 6 – Modelos GOBF-RBF para o sistema de nível.

NF	NP	NC	TP	EQM	AIC	BIC
8	1	4	70	0,6048	-3,6285.10 ²	-1,9311.10 ¹
3	1	3	23	0,3794	-9,2328.10 ²	-8,1040.10 ²

A Figura 17 apresenta o comportamento do modelo de melhor EQM obtido pela otimização não linear para os dados de validação.

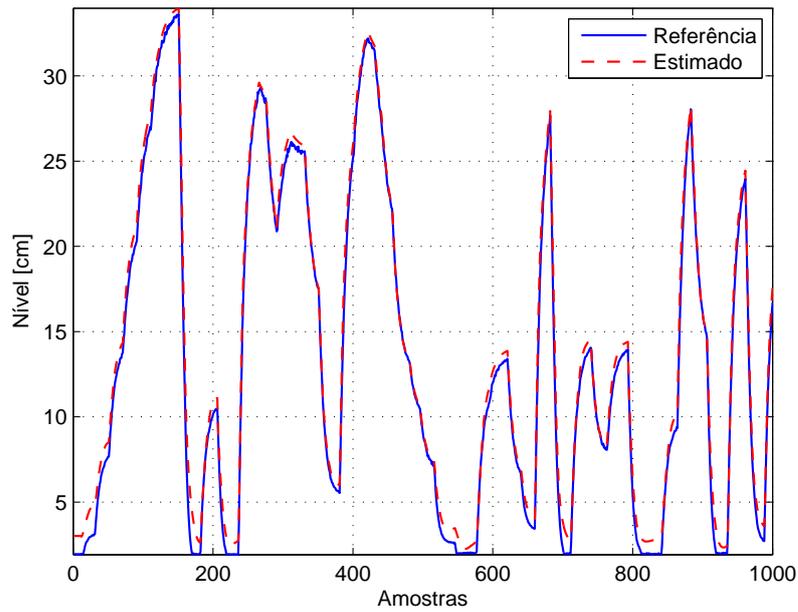


Figura 17 – Resposta do modelo com 1 polo e 8 funções na base otimizado pelo QPSO para os dados de validação do sistema de nível.

O comportamento do algoritmo ao longo das iterações pode ser observado na Figura 18.

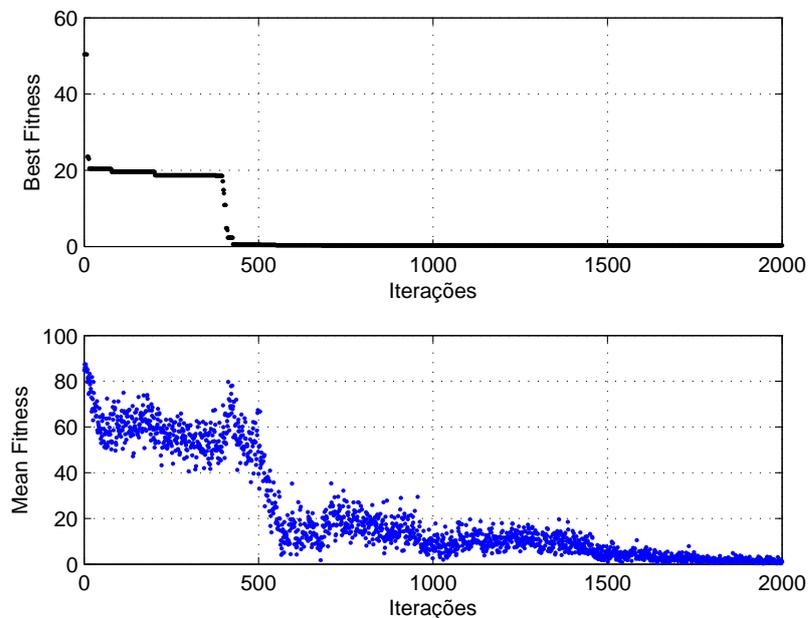


Figura 18 – Comportamento do algoritmo QPSO para o modelo com 1 polo e 8 funções na base para o sistema de nível.

A Figura 19 apresenta o comportamento do modelo de melhor BIC para os dados de validação.

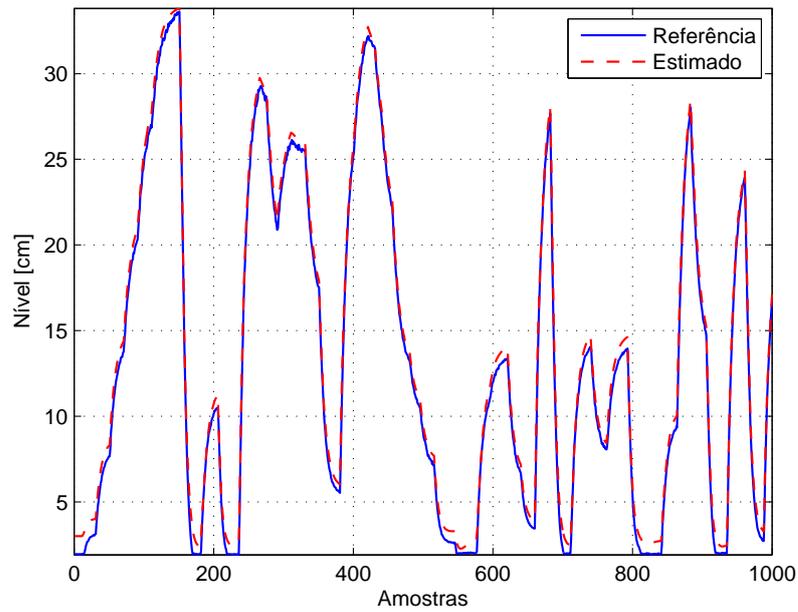


Figura 19 – Resposta do modelo com 1 polo e 3 funções na base otimizado pelo QPSO para os dados de validação do sistema de nível.

O comportamento do algoritmo ao longo das iterações pode ser observado na Figura 20.

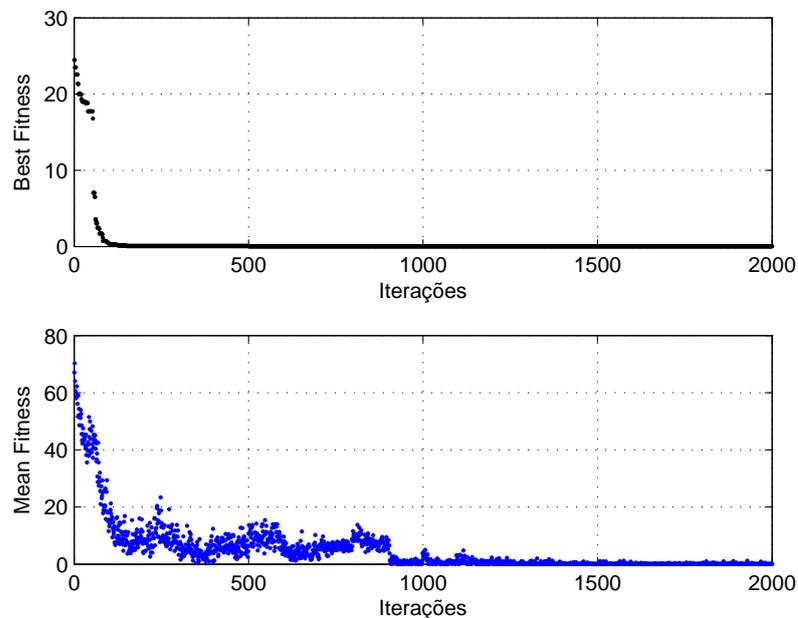


Figura 20 – Comportamento do algoritmo QPSO para o modelo com 1 polo e 3 funções na base para o sistema de nível.

Como pode-se observar pelos resultados, a metodologia baseada no QPSO obteve uma resposta satisfatória, porém inferior à obtida pela otimização não linear. A resposta

obtida para a estrutura com mais parâmetros (selecionada pelo EQM) apresentou-se muito inferior a estrutura selecionada pelo BIC, mesmo que em ambos os casos o *mean fitness* tenha se estabilizado dentro da janela de iterações proposta. Pode-se indicar como motivo para tal comportamento, o elevado número de parâmetros a serem otimizados. É possível que com mais iterações ou mais partículas obtenha-se uma resposta mais próxima à obtida pela otimização não linear.

4.2.2 Modelo racional NARX

No exemplo aqui apresentado utiliza-se um sistema não linear que exibe comportamentos dinâmicos distintos em diferentes regiões da zona de operação (LAZAR, 2001).

$$y(k) = \frac{2,5y(k-1)y(k-2)}{1 + y(k-1)^2 + y(k-2)^2} + 0,3 \cos(0,5(y(k-1) + y(k-2))) + 1,2u(k-1). \quad (4.3)$$

O processo de identificação e a posterior validação do modelo obtido foi feito com base em um conjunto de dados obtidos da simulação do sistema pela obtenção de 300 pares de amostras de entrada e saída, com entradas variando entre 0 e 2 em degraus aleatórios de distribuição uniforme e duração de 10 amostras. A Figura 21 apresenta os dados de entrada e saída utilizados no processo de estimação do modelo. De forma semelhante ao exemplo anterior, variou-se o número de polos e blocos de funções ortonormais utilizadas com a finalidade de obter-se um modelo de qualidade sem conhecimento prévio das características dinâmicas deste.

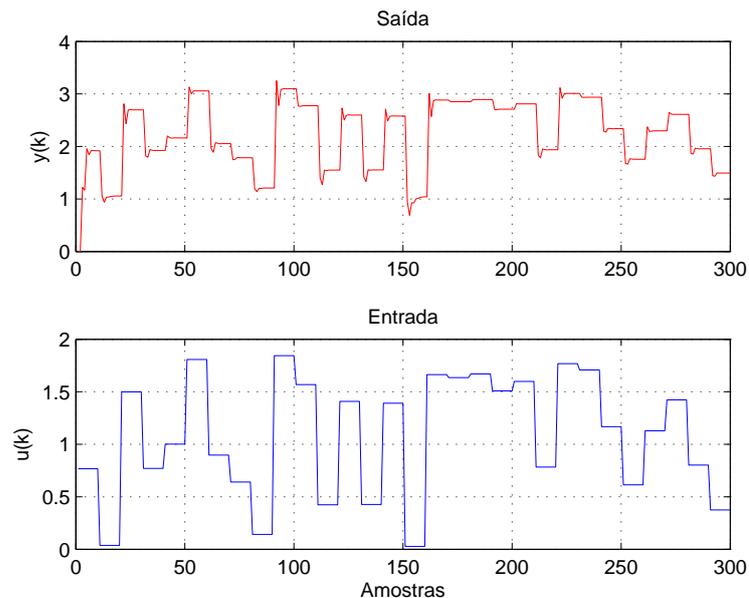


Figura 21 – Dados de identificação do sistema NARX.

Os dados utilizados no processo de validação pode ser observados na Figura 22.

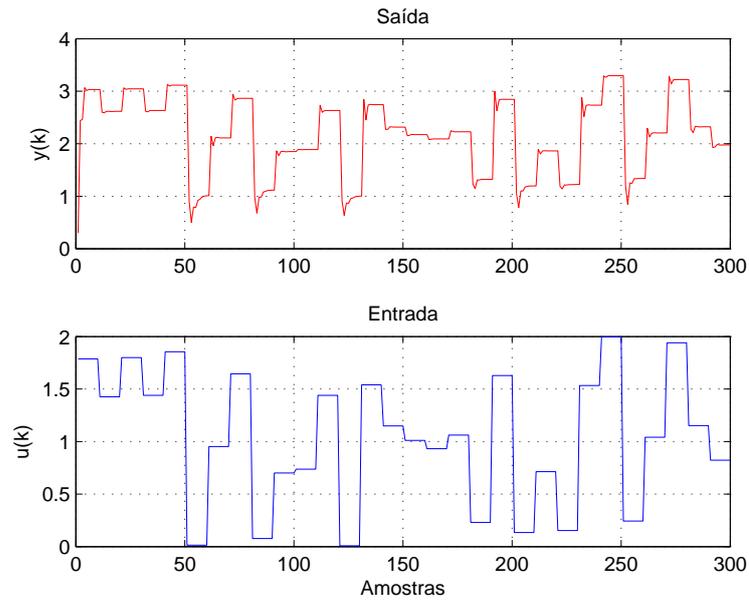


Figura 22 – Dados de validação do sistema NARX.

4.2.2.1 Otimização não linear

O mesmo procedimento apresentado no exemplo anterior foi aplicado para o sistema NARX. A Tabela 7 apresenta o número de funções ortonormais (NF), o número de polos (NP), o número de centros (*clusters*) da rede neural aplicada e o número de parâmetros totais estimados e otimizados do modelo (TP), o EQM, AIC e BIC para os melhores modelos obtidos (EQM's obtidos a partir dos dados de validação).

Tabela 7 – Modelos GOBF-RBF para o sistema NARX.

NF	NP	NC	TP	EQM	AIC	BIC
1	1	2	8	$9,0105 \cdot 10^{-3}$	$-1,3968 \cdot 10^3$	$-1,3549 \cdot 10^3$
1	3	3	25	$5,7032 \cdot 10^{-3}$	$-1,5000 \cdot 10^3$	$-1,3689 \cdot 10^3$
4	2	16	275	$5,2868 \cdot 10^{-4}$	$-1,7135 \cdot 10^3$	$-2,7138 \cdot 10^2$

A Tabela 8 apresenta os valores iniciais e finais de γ_i para o modelo com melhor EQM obtido.

Tabela 8 – Modelo com melhor EQM para o modelo racional NARX.

Υ inicial	Υ final	EQM
$\Upsilon = [0,5000 \quad 0,5000]$	$\Upsilon = [0,5913 \quad 0,0656]$	$5,2868 \cdot 10^{-4}$

O valor dos polos iniciais associados aos valores do vetor Υ inicial são $p_1 = -0,5931$ e $p_2 = 0,8431$, o valor dos polos finais, referentes ao vetor Υ apresentado na Tabela 8 após o processo de otimização são $p_1 = -0,1005$ e $p_2 = 0,6530$.

A Figura 23 apresenta a saída do modelo GOBF-RBF apresentado nas tabelas anteriores para os dados de validação do modelo racional NARX para o modelo com melhor EQM.

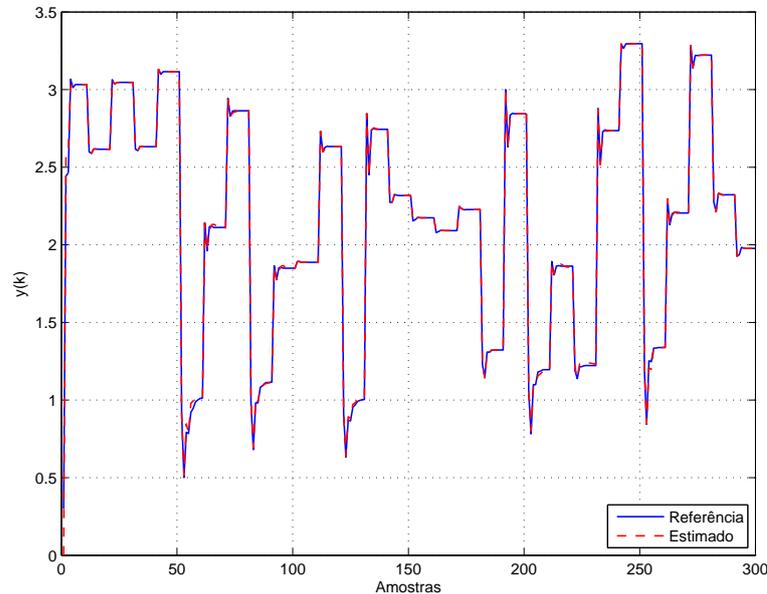


Figura 23 – Resposta do modelo selecionado pelo BIC para os dados de validação do sistema NARX, utilizando a otimização não linear.

A tabela 9 apresenta os resultados obtidos por Machado (2011) para o sistema NARX utilizando outras estruturas com GOBFs.

Tabela 9 – Comparação da resposta do modelo NAXR com outras estruturas.

Tipo de modelo	EQM
GOBF-RBF	$5,2868 \cdot 10^{-4}$
Volterra-GOBF (MACHADO, 2011)	$1,0730 \cdot 10^{-3}$
Volterra-GOBF Direções Independentes (MACHADO, 2011)	$8,3425 \cdot 10^{-4}$

Como observado pela resposta do modelo com melhor EQM e pela comparação apresentada na Tabela 9, a metodologia proposta foi capaz de obter uma representação com boa exatidão do sistema em estudo.

Realizando-se a mesma análise feita para o exemplo anterior relativa a parcimoniedade dos modelos, a Figura 24 apresenta a saída do modelo GOBF-RBF com menor BIC para os dados de validação do modelo racional NARX.

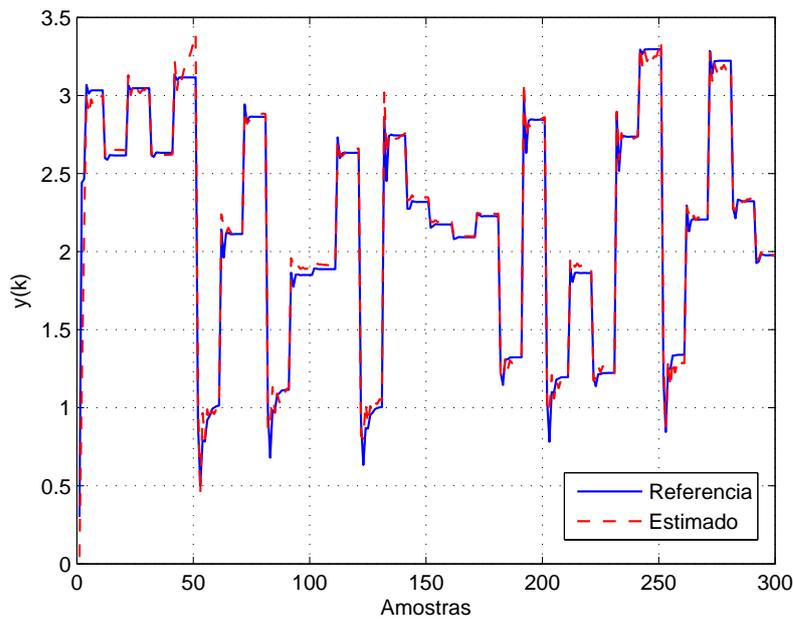


Figura 24 – Resposta do modelo selecionado pelo BIC para os dados de validação do sistema NARX, utilizando a otimização não linear.

Como pode ser notado, a dinâmica do modelo com menor BIC apresenta alguns problemas, mas ainda assim é capaz de representar o sistema com relativa exatidão. Conclui-se que a seleção do modelo tendo em vista sua parcimoniedade deve ser analisada com cautela. É importante considerar o objetivo da modelagem, uma vez que determinadas dinâmicas não representadas corretamente podem ser cruciais para o projeto em desenvolvimento.

4.2.2.2 Otimização pelo QPSO

O procedimento de otimização pelo QPSO foi realizado nas duas estruturas selecionadas na seção anterior (a de melhor EQM e de melhor BIC). Os parâmetros utilizados no algoritmo encontram-se na Tabela 10.

Tabela 10 – Parâmetros do QPSO para o sistema NARX.

nº de partículas	nº máx. de passos	α	precisão
50	5000	1,2 \rightarrow 0,8	10 ⁻⁵

A Tabela 11 apresenta o EQM, AIC e BIC para esses modelos.

Tabela 11 – Modelos GOBF-RBF para o sistema de nível.

NF	NP	NC	TP	EQM	AIC	BIC
1	3	3	25	$8,5268 \cdot 10^{-3}$	$-1,3794 \cdot 10^3$	$-1,2868 \cdot 10^3$
4	2	16	275	$2,4160 \cdot 10^{-2}$	$-1.0089 \cdot 10^3$	$-8,0891 \cdot 10^2$

A Figura 25 apresenta o comportamento do modelo de melhor EQM obtido pela otimização não linear para os dados de validação.

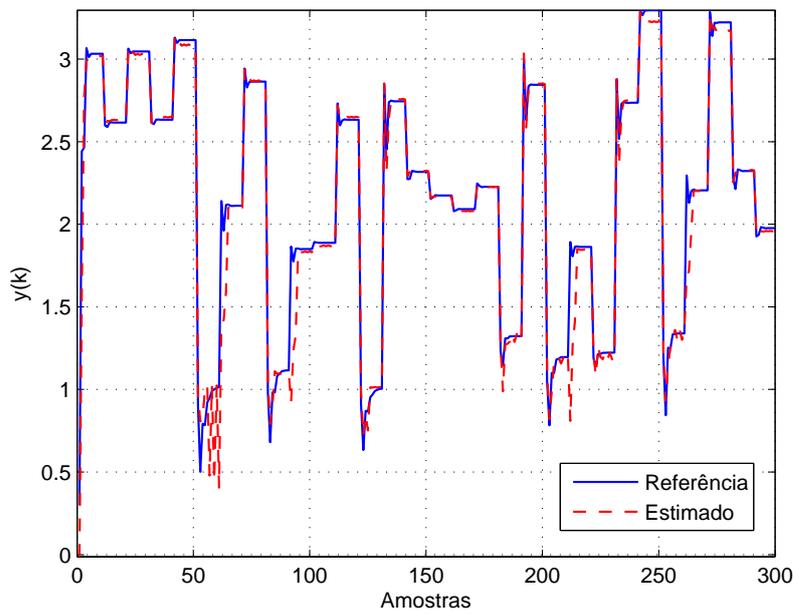


Figura 25 – Resposta do modelo com 2 polos e 4 funções na base para os dados de validação do sistema de nível.

O comportamento do algoritmo ao longo das iterações pode ser observado na Figura 26.

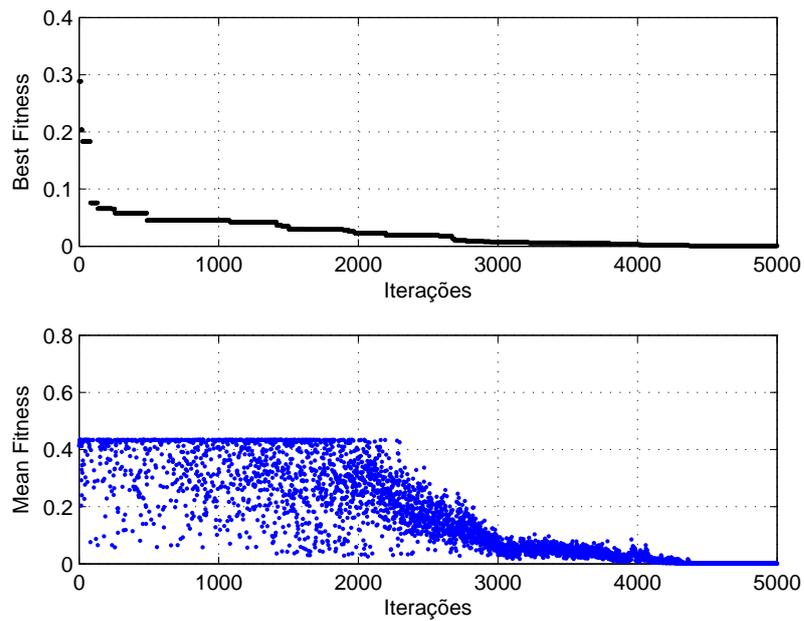


Figura 26 – Comportamento do algoritmo QPSO para o modelo com 2 polos e 4 funções na base para o sistema de nível.

A Figura 27 apresenta o comportamento do modelo de melhor BIC para os dados de validação.

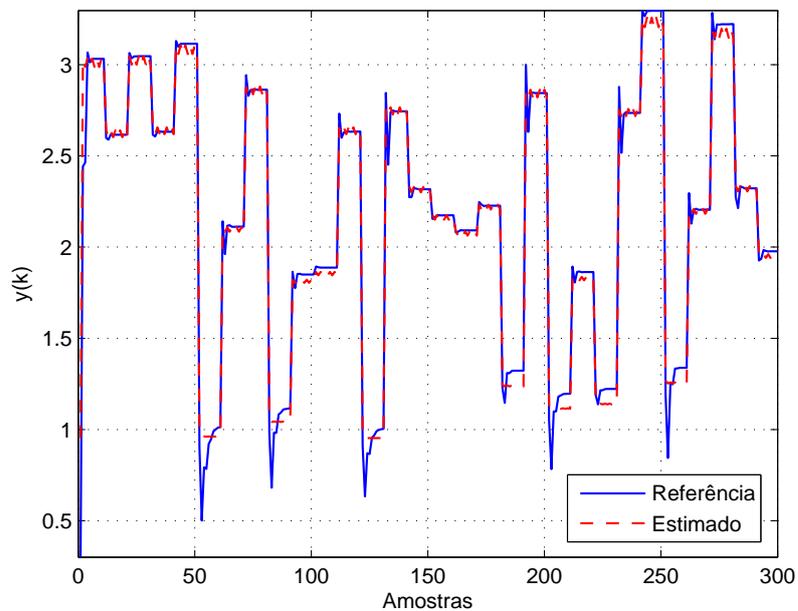


Figura 27 – Resposta do modelo com 3 polos e 1 função na base para os dados de validação do sistema NARX.

O comportamento do algoritmo ao longo das iterações pode ser observado na Figura 28.

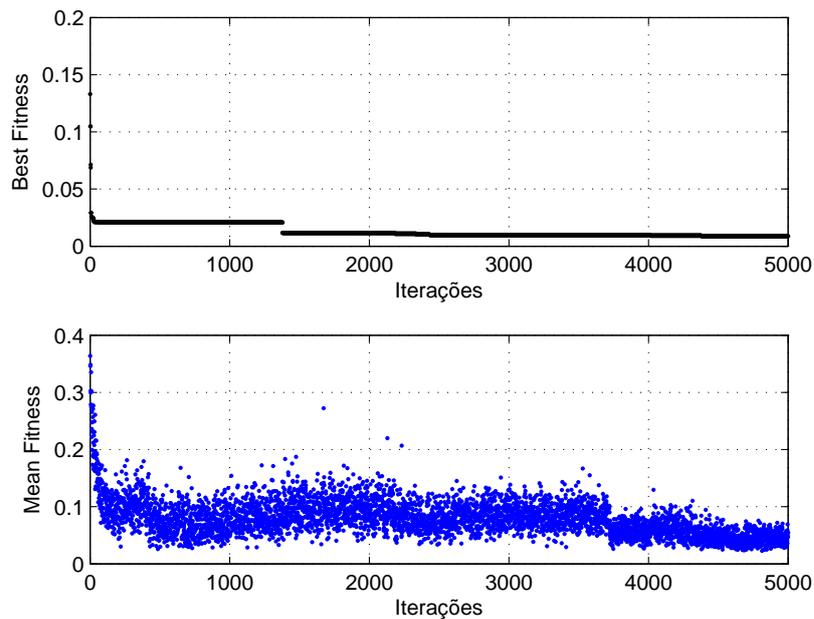


Figura 28 – Comportamento do algoritmo QPSO para o modelo com 3 polos e 1 função na base para o sistema NARX.

Os resultados obtidos pelo QPSO para o modelo NARX foram insatisfatórios. No entanto, algumas conclusões sobre o processo de otimização podem ser realizadas. Apesar da enorme quantidade de parâmetros da estrutura com melhor EQM, o QPSO foi capaz de obter um modelo de boa qualidade para os dados de treinamento, porém a resposta do modelo para os dados de validação diverge justamente em dinâmicas não contempladas nos dados de treinamento, o que indicaria um sobre-treinamento da rede.

Na estrutura com melhor BIC pela otimização não linear, nota-se que o QPSO não foi capaz de parametrizar a dinâmica do sistema. Analisando-se o comportamento do *mean fitness*, uma possibilidade seria aumentar o número máximo de passos do algoritmo, porém, a fim de se manter a isonomia para uma comparação coerente dos procedimentos, optou-se por limitar os passos como apresentado.

4.2.3 Modelo de um sistema de levitação magnética

Sistemas de levitação magnética são utilizados para levantar objetos com características magnéticas sem contato mecânico. Essa tecnologia é aplicada com sucesso em trens de levitação magnética, nos quais, devido à ausência de atrito, tem-se uma redução de perdas de energia por fricção no processo de transporte (AHMAD; JAVAID, 2010). Além de trens de alta velocidade (LIM *et al.*, 2013; SHU; MEISINGER, 2011), esta tecnologia também é aplicada em sistemas de posicionamento de alta precisão (CHEUNG *et al.*, 2009) e outras áreas. Os processos associados a sistemas de levitação magnética constituem modelos com características não lineares, instáveis em malha aberta e de dinâmicas

rápidas. Assim, se faz necessário a obtenção de modelos com exatidões adequadas com o propósito de simular sistemas reais de levitação e projetar malhas de controle eficientes associadas aos processos correspondentes (SUSTER; JADLOVSKA, 2012). A maioria das técnicas convencionais de modelagem ou identificação de sistemas utiliza procedimentos de linearização com a finalidade de facilitar a obtenção dos modelos matemáticos associados. Entretanto, pode haver perdas de informações de algumas características reais do processo.

A Figura 29 ilustra um sistema de levitação magnética, no qual m é a massa do corpo levitado (por exemplo, uma esfera de metal), x_b é a distância do corpo até o eletroímã de levitação, i_c é a corrente na bobina do imã, F_m a força magnética exercida sobre a esfera pelo campo gerado pela bobina do processo.

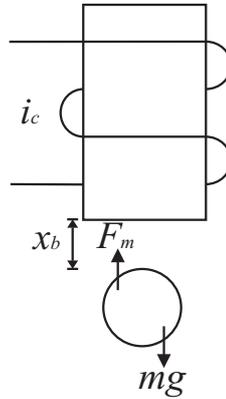


Figura 29 – Sistema de Levitação Magnética.

Aplicando a segunda lei de Newton tem-se

$$F_{ext} = F_g - F_m. \quad (4.4)$$

no qual F_{ext} é a resultante das forças aplicadas sobre o corpo, F_g é a força da gravidade e F_m é a força eletromagnética exercida sobre a esfera pelo campo magnético criado pela bobina. A força magnética F_m é dada por (4.5), cuja K_m é uma constante de proporcionalidade.

$$F_m = \frac{K_m i_c(t)^2}{2x_b^2} \quad (4.5)$$

A força exercida pela gravidade é dada por

$$F_g = mg. \quad (4.6)$$

Combinando as equações anteriores tem-se a equação diferencial não linear de modelagem expressa por (4.7). Nota-se no modelo resultante, que a informação da aceleração

do corpo levitado depende do quadrado do valor da corrente que energiza a bobina do eletroímã, e do inverso do valor quadrático da distância entre o objeto levitado e a borda da bobina de campo do processo. No modelo matemático apresentado não foram consideradas outras características não lineares que ocorrem na prática, como a saturação do campo magnético gerado pela bobina em função da corrente de energização, assim como dispersões do campo em relação à geometria do solenoide utilizado. Estas características adicionais não possuem modelagens triviais, limitando a representação do modelo expresso por (4.7).

$$m\ddot{x}_b = mg - \frac{K_m i_c(t)^2}{2x_b^2} \quad (4.7)$$

Neste contexto, foi realizada a aquisição de dados com uma taxa de amostragem de 0,1 segundos de um sistema de levitação magnética real em escala laboratorial. Para tanto, este foi estabilizado e o *setpoint* (posição desejada) foi utilizado como entrada para fins de identificação e a posição medida da esfera de metal como variável de saída. Foram obtidos dois conjuntos de dados, um para identificação e um segundo reservado para posterior validação. A Figura 30 apresenta o sistema real a ser modelado.



Figura 30 – Sistema de levitação magnética real a ser identificado (QUANSER, 2012).

O conjunto de dados utilizados para identificação pode ser observado na Figura 31.

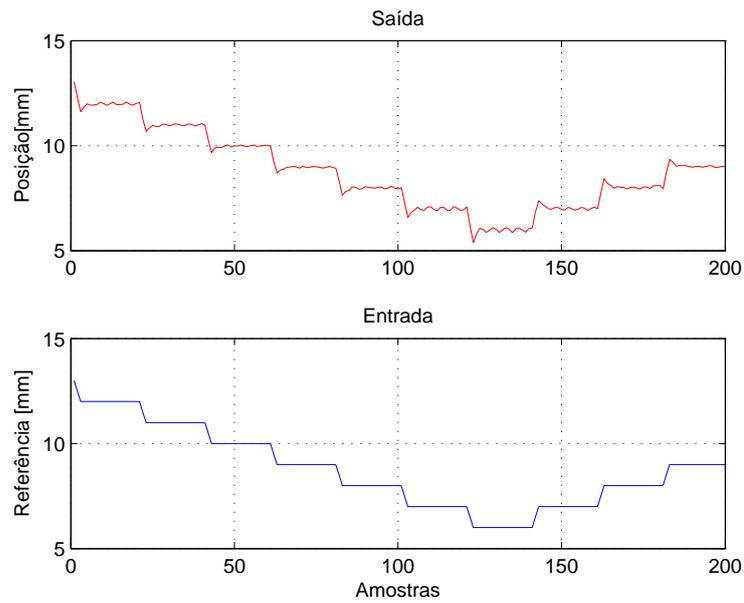


Figura 31 – Dados de identificação do sistema de levitação magnética.

Os dados utilizados no processo de validação podem ser observados na Figura 32

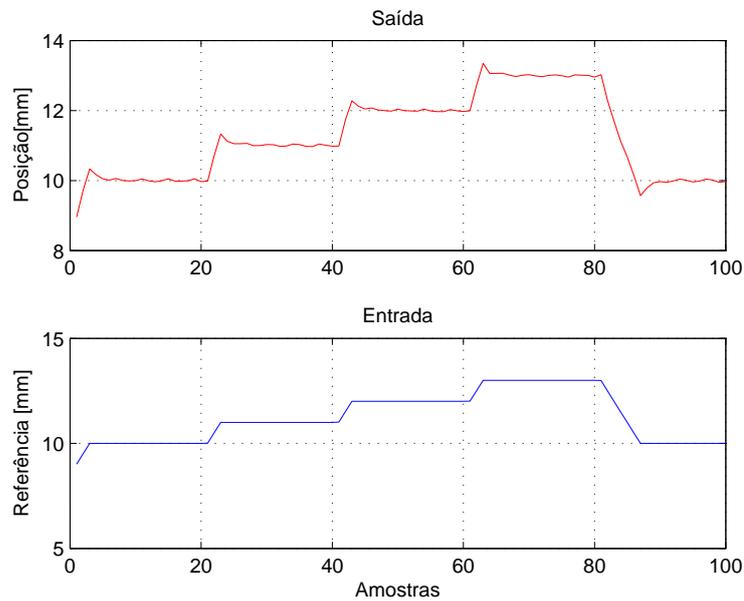


Figura 32 – Dados de identificação do sistema de levitação magnética.

4.2.3.1 Otimização não linear

O mesmo procedimento apresentado no exemplo anterior foi aplicado para o sistema de levitação magnética. A Tabela 12 apresenta o número de funções ortonormais (NF), o número de polos (NP), o número de centros (*clusters*) da rede neural aplicada e o número de parâmetros totais estimados e otimizados do modelo (TP), o EQM, AIC e BIC para os melhores modelos obtidos (EQM's obtidos a partir dos dados de validação).

Tabela 12 – Modelos GOBF-RBF para o sistema de levitação magnética.

NF	NP	NC	TP	EQM	AIC	BIC
3	2	2	29	0,2197	-93,5492	13,8605
4	2	2	37	0,2055	-84,2309	52,8090
2	3	2	30	0,2432	-81,3871	29,7264
3	3	6	118	0,1758	62,1592	499,2055
1	4	3	32	0,2980	-57,0662	61,4549

A Tabela 13 apresenta os valores iniciais e finais de Υ_i para o modelo com melhor EQM.

Tabela 13 – Modelo com melhor EQM para o sistema de levitação magnética.

Υ inicial	Υ final	EQM
$\Upsilon = [0.5000 \ 0.5000 \ 0.5000]$	$\Upsilon = [-0,1843 \ 0,5579 \ 0,9375]$	0,0105

O valor dos polos iniciais associados aos valores do vetor Υ inicial são $p_1 = 0,9495$ e $p_2 = -0,4748 \pm 0,5488i$, o valor do polos finais, referentes ao vetor Υ apresentado na Tabela 13, após o processo de otimização são $p_1 = 0,9908$ e $p_2 = 0,7977 \pm 0,5568i$. A Figura 33 apresenta a saída do modelo GOBF-RBF com melhor EQM para os dados de validação do sistema de levitação magnética.

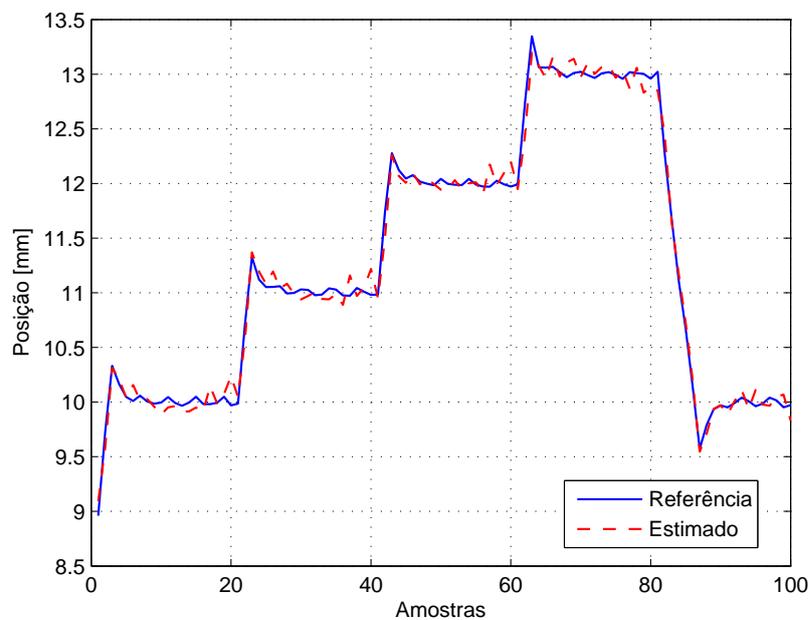


Figura 33 – Resposta do modelo selecionado pelo EQM para os dados de validação do sistema de levitação magnética, utilizando a otimização não linear.

Como pode ser observado pela Figura 33 e pelo EQM do modelo obtido para

os dados de validação, foi possível representar o sistema de levitação magnética com a metodologia proposta com boa exatidão.

Novamente o modelo selecionado pelo BIC difere do selecionado considerando apenas o EQM. A Figura 34 apresenta a resposta do modelo com menor BIC para os dados de validação

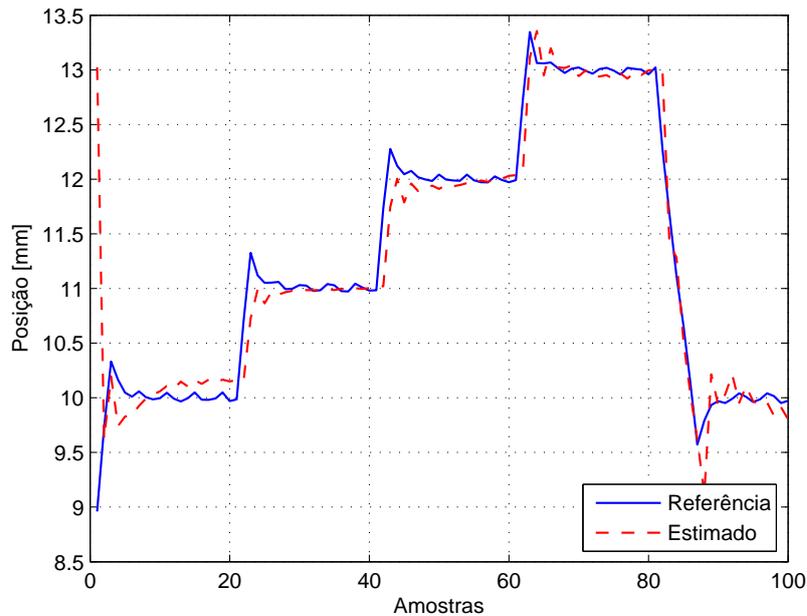


Figura 34 – Resposta do modelo selecionado pelo BIC para os dados de validação do sistema de levitação magnética, utilizando a otimização não linear.

Para o conjunto de soluções obtido para o sistema de levitação magnética, a análise da parcimoniedade da estrutura se torna crítica, uma vez que o modelo com menor BIC é capaz de representar a com apenas um quarto dos parâmetros do modelo com menor EQM.

4.2.3.2 Otimização pelo QPSO

O procedimento de otimização pelo QPSO foi realizado nas duas estruturas selecionadas na seção anterior (a de melhor EQM e de melhor BIC). Os parâmetros utilizados no algoritmo encontram-se na Tabela 14.

Tabela 14 – Parâmetros do QPSO para o sistema de levitação magnética.

nº de partículas	nº máx. de passos	α	precisão
50	2000	1,2 \rightarrow 0,8	10^{-3}

A Tabela 15 apresenta o EQM, AIC e BIC para esses modelos.

Tabela 15 – Modelos GOBF-RBF para o sistema de levitação magnética.

NF	NP	NC	TP	EQM	AIC	BIC
3	2	2	29	0,2750	-71,0984	36,3113
3	3	6	118	0,3827	139,9496	576,9959

A Figura 35 apresenta o comportamento do modelo de melhor EQM obtido pela otimização não linear para os dados de validação.

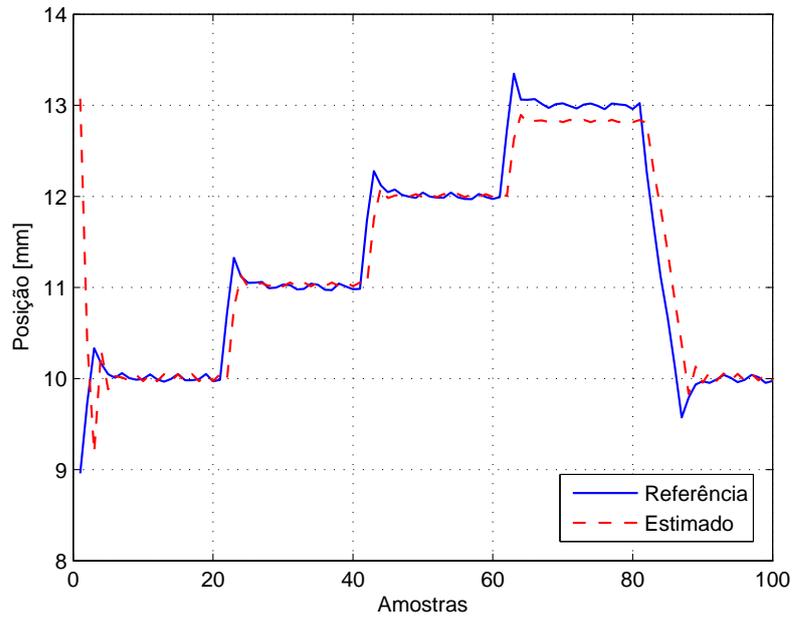


Figura 35 – Resposta do modelo com 3 polos e 2 funções na base para os dados de validação do sistema de levitação magnética.

O comportamento do algoritmo ao longo das iterações pode ser observado na Figura 36.

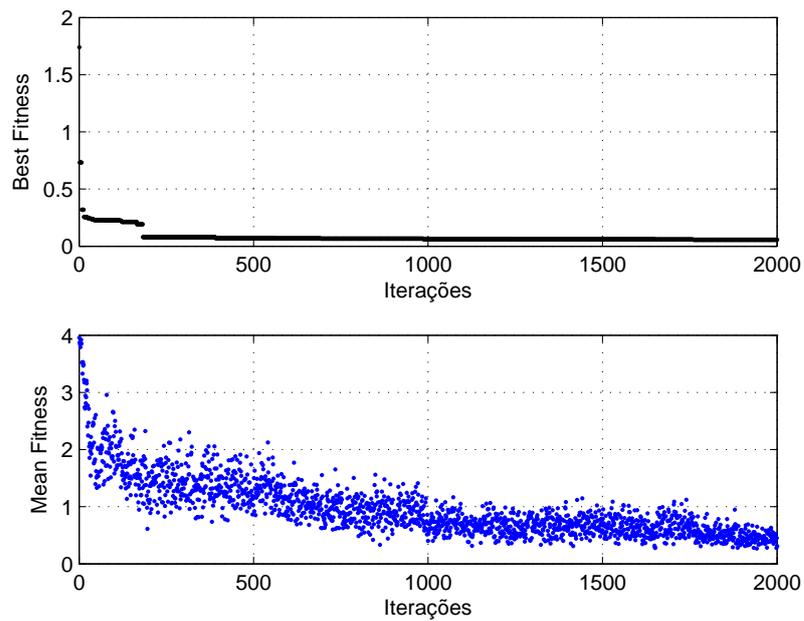


Figura 36 – Comportamento do algoritmo QPSO para o modelo com 3 polos e 2 funções na base para o sistema de levitação magnética.

A Figura 37 apresenta o comportamento do modelo de melhor BIC para os dados de validação.

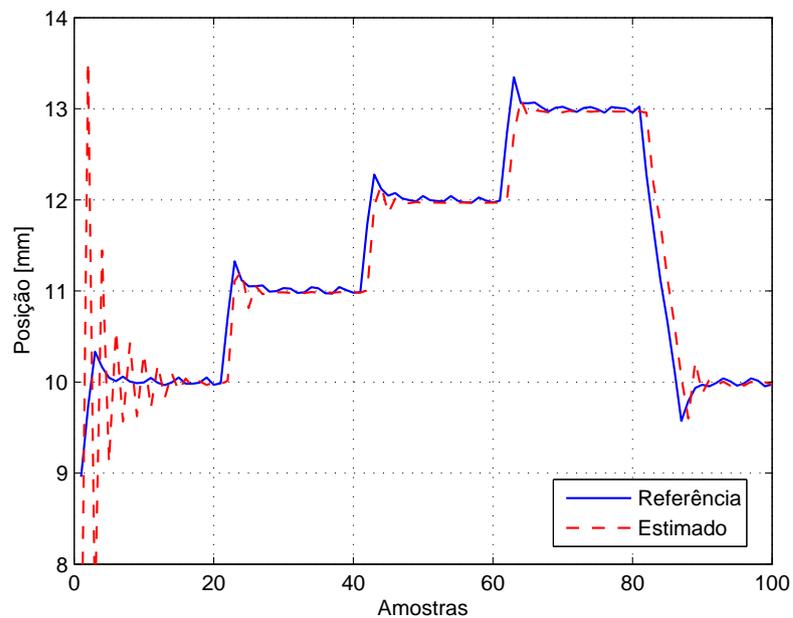


Figura 37 – Resposta do modelo com 3 polos e 3 funções na base otimizado pelo QPSO para os dados de validação do sistema de levitação magnética.

O comportamento do algoritmo ao longo das iterações pode ser observado na Figura 38.

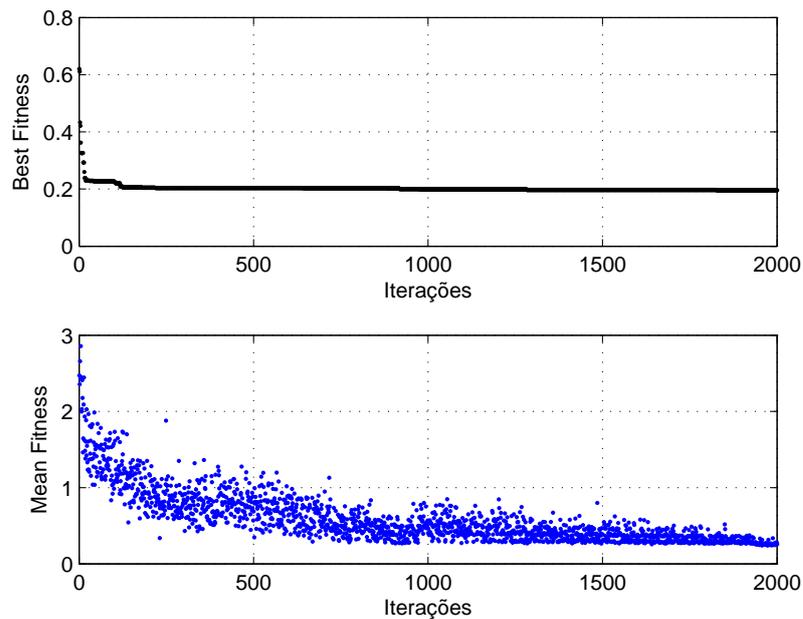


Figura 38 – Comportamento do algoritmo QPSO para o modelo com 3 polos e 3 funções na base para o sistema de levitação magnética.

Como pode-se observar pelos resultados, a metodologia baseada no QPSO obteve uma resposta satisfatória, porém inferior à obtida pela otimização não linear. Nota-se também, que o algoritmo comporta-se melhor quando o número de parâmetros a serem otimizados é menor, ou seja, a resposta se aproxima mais do método de otimização não linear.

4.3 Simulação de Monte Carlo

A fim de analisar a robustez da metodologia proposta e sua capacidade de convergência sob a hipótese de inicializações aleatórias dos polos, desenvolveu-se algumas simulações de Monte Carlo.

Duas estruturas de modelo do sistema de nível e duas do sistema de levitação magnética foram selecionadas (uma com o melhor EQM e outra com o melhor BIC) e 100 inicializações aleatórias uniformemente distribuídas dos polos foram realizadas.

Os resultados das simulações para o sistema de nível para o procedimento envolvendo a otimização não linear encontram-se na Tabela 16. Os valores de EQM apresentados nessa foram obtidos a partir da resposta do modelo para os dados de validação.

Tabela 16 – Resultados da simulação de Monte Carlo para o sistema de nível usando a otimização não linear.

Modelo	EQM (Médio)	EQM (desvio padrão)	Melhor EQM	Pior EQM
NF=8 NP=1	0,3318	0,0382	0,2998	0,4307
NF=3 NP=1	0,3656	0,0518	0,3104	0,4611

As Figuras 39a e 39b apresentam, na forma de histograma, as distribuições para as simulações realizadas.

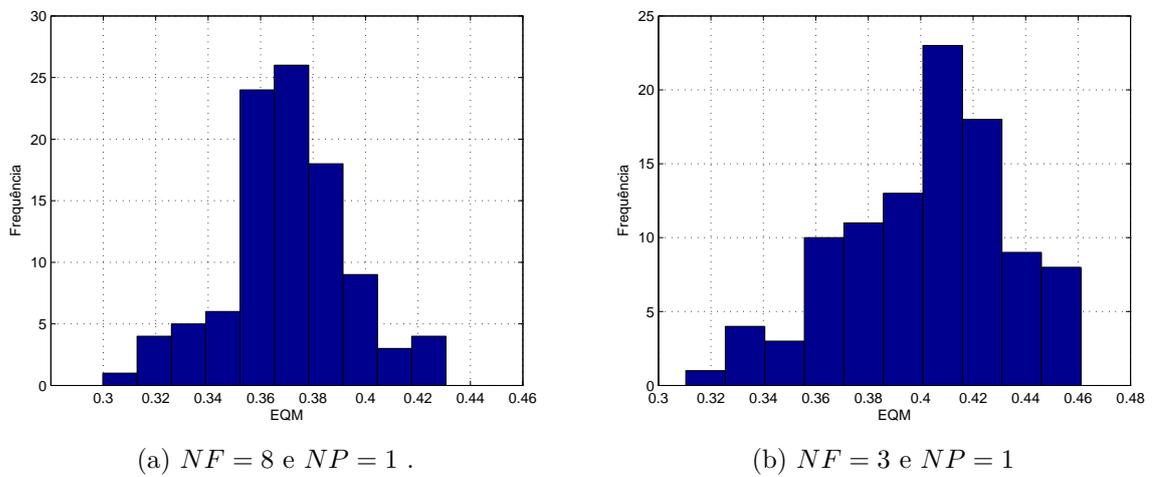


Figura 39 – Histograma das simulações de Monte Carlo para o sistema de nível.

Os resultados das simulações para o sistema de levitação magnética encontram-se na Tabela 17. Da mesma forma que anteriormente, os valores de EQM apresentados nessa foram obtidos a partir da resposta do modelo para os dados de validação.

Tabela 17 – Resultados da simulação de Monte Carlo para o sistema de levitação magnética usando a otimização não linear.

Modelo	EQM (Médio)	EQM (desvio padrão)	Melhor EQM	Pior EQM
NF=3 NP=3	0,2655	0,0512	0,1536	0,3725
NF=3 NP=2	0,2944	0,0556	0,1771	0,4032

As Figuras 40a e 40b apresentam, na forma de histograma, as distribuições para as simulações realizadas para o sistema de levitação magnética.

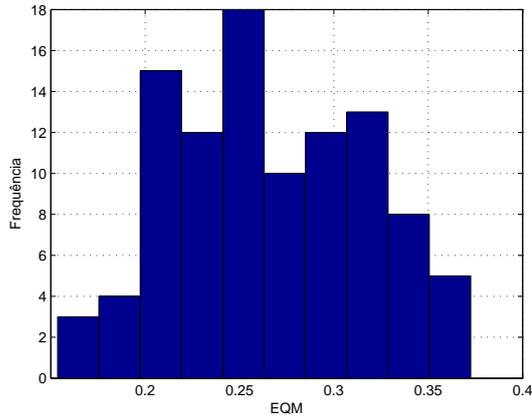
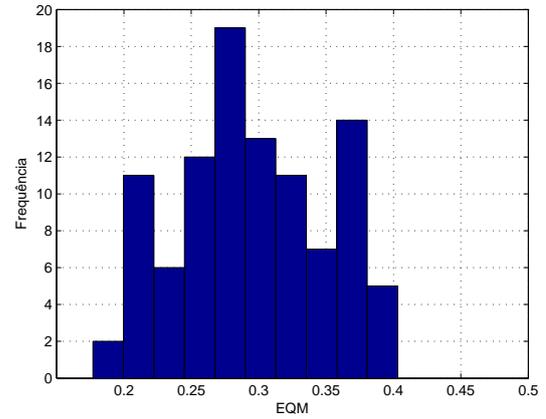
(a) $NF = 3$ e $NP = 3$.(b) $NF = 3$ e $NP = 2$.

Figura 40 – Histograma das simulações de Monte Carlo para o sistema de levitação magnética.

Os resultados apresentados demonstram que a metodologia proposta para a otimização não linear é eficaz mesmo sob a suposição de inicializações aleatórias dos polos. Possibilitando ainda que, uma vez obtido um conjunto de modelos com mesma estrutura, pode-se escolher entre eles o com melhor exatidão.

5 Comentários Finais, Conclusões e Contribuições

Desenvolveu-se neste trabalho a modelagem de sistemas dinâmicos não lineares através de base de funções ortonormais generalizadas (GOBF) com funções internas em uma estrutura modificada aliada a redes de funções de base radial (RBF). A abordagem foi capaz de representar sistemas dinâmicos não lineares a partir de sinais medidos ou simulados de entrada e saída desses.

A principal contribuição desta tese constituiu-se em apresentar procedimentos sistemáticos de otimização e ajuste fino dos parâmetros dos modelos não lineares baseados na estrutura GOBF-RBF em estudo, tendo em vista os resultados apresentados em outros trabalhos (OLIVEIRA *et al.*, 2011; CAMPELLO; AMARAL, 2002; MACHADO, 2011; MEDEIROS; AMARAL; CAMPELLO, 2006; CAMPELLO; OLIVEIRA; AMARAL, 2007). O modelo GOBF-RBF em estudo é capaz de manter as características vantajosas da modelagem de sistemas dinâmicos por BFO e agregar as capacidades de aproximação universal das redes com funções de base radial, com vantagens para aplicações de controle baseado em modelos quando comparada a técnicas semelhantes citadas na literatura.

O processo de identificação de sistemas dos exemplos estudados foi realizado seguindo a abordagem proposta. Inicialmente definem-se os valores iniciais dos parâmetros γ das GOBFs que se relacionam diretamente com os polos das mesmas. Com as saídas dos filtros dessas GOBFs e as saídas esperadas do sistema a ser modelado aplica-se a técnica *Fuzzy C-means* aliada ao *Subtractive Clustering*, garantindo-se um número ótimo de *clusters* e com densidades apropriadas. Com os dados dos *clusters*, determina-se os parâmetros das RBFs (centros, desvios padrão e pesos da camada de saída), obtendo-se um modelo inicial não refinado. Foram apresentadas duas sugestões para o ajuste fino dos parâmetros do modelo, um com base no cálculo dos gradientes e a técnica de otimização não linear de Levenberg-Marquardt, que constitui outra contribuição do trabalho, e outro utilizando o método metaheurístico QPSO.

Os resultados mostraram que, apesar da qualidade das respostas obtidas pelos refinamentos realizados via QPSO, os obtidos pela otimização não linear se mostraram superiores. A otimização não linear, porém, se vale do uso dos gradientes exatos obtidos, o que torna o método mais complexo, exigindo um melhor entendimento da estrutura em análise.

Entre as perspectivas para trabalhos futuros relacionados ao tema estudado apontam-

se:

- Análise de uma possível melhoria da performance dos resultados obtidos pelo QPSO pela divisão do processo em dois laços de otimização aninhados. O primeiro otimizaria apenas os polos da estrutura linear formada pelas GOBFs e o segundo, os parâmetros da rede RBF. Vale frisar, no entanto, o possível aumento do já elevado custo computacional que essa proposta traria;
- Estudar a capacidade de aproximação da arquitetura estudada, ou seja, quais tipos de moledos dinâmicos não lineares podem ser aproximados por esta;
- Estudo do uso de técnicas multiobjetivo para determinação da estrutura e seus parâmetros. Destaca-se entre essas, tendo em vista os estudos já realizados no trabalho, o chamado MOPSO (COELLO; LECHUNGA, 2002);
- Utilização dos modelos aqui obtidos aliados a técnicas de controle a fim de sintetizar controladores baseados em modelos.

APÊNDICE A – Cálculos Analíticos dos Gradientes para as Funções Generalizadas

Para o cálculo dos gradientes analíticos utiliza-se o modelo de estados descrito pela equação (2.24) e representado por

$$\Psi(k+1) = A_f \Psi(k) + B_f u(k), \quad (\text{A.1})$$

na qual $\Psi(k)$ é formado por $n * n_b$ saídas das funções de base.

Desenvolvendo-se o modelo (A.1) é possível verificar que a solução de $\Psi(k)$ é dada por

$$\Psi(k) = A_f^k \Psi(0) + \sum_{i=0}^{k-1} A_f^i B_f u(k-1-i). \quad (\text{A.2})$$

O gradiente de $\Psi(k)$ com relação aos parâmetros γ_l é dado por

$$\frac{\partial \Psi(k)}{\partial \gamma_l} = \frac{\partial A_f^k}{\partial \gamma_l} \Psi(0) + \sum_{i=0}^{k-1} \left(\frac{\partial A_f^i}{\partial \gamma_l} B_f + A_f^i \frac{\partial B_f}{\partial \gamma_l} \right) u(k-1-i), \quad (\text{A.3})$$

sendo $\frac{\partial A_f^i}{\partial \gamma_l}$ dada por

$$\frac{\partial A_f^i}{\partial \gamma_l} = \sum_{j=1}^i A_f^{j-1} \frac{\partial A_f^j}{\partial \gamma_l} A_f^{i-j}. \quad (\text{A.4})$$

Como as matrizes A_f e B_f dependem de A_b e B_b deve ser obtido também os gradientes dessas funções para o cálculo dos gradientes anteriores. Para a matriz A_b tem-se que sua derivada com relação a γ_l é dada por

$$\frac{\partial A_b}{\partial \gamma_l} = \begin{bmatrix} da_{1,1} & da_{1,2} & da_{1,3} & \cdots & da_{1,nb} \\ da_{2,1} & da_{2,2} & da_{2,3} & \cdots & da_{2,nb} \\ da_{3,1} & da_{3,2} & da_{3,3} & \cdots & da_{3,nb} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ da_{nb,1} & da_{nb,2} & da_{nb,3} & \cdots & da_{nb,nb} \end{bmatrix} \quad (\text{A.5})$$

na qual os termos $da_{i,j}$, com i sendo a linha que ao qual o elemento pertence e j a coluna, são definidos por:

- Para $i = 1$ os elementos são:

- Para $l = 1$:

$$da_{1,j} = 1, \text{ para } j = 1$$

$$da_{1,j} = -\gamma_1(1 - \gamma_1)^{-1/2} \gamma_j \prod_{k=2}^{j-1} \sqrt{1 - \gamma_k^2}, \text{ para } j = 2, \dots, nb.$$

- Para $l = 2, \dots, n_b$:

$$da_{1,j} = 0, \text{ para } j = 1$$

$$da_{1,j} = -\gamma_l(1 - \gamma_l)^{-1/2} \gamma_j \prod_{\substack{k=2 \\ k \neq l}}^{j-1} \sqrt{1 - \gamma_k^2}, \text{ para } l < j.$$

$$da_{1,j} = \prod_{k=1}^{j-1} \sqrt{1 - \gamma_k^2}, \text{ para } j = 1$$

$$da_{1,j} = 0, \text{ para } l > j;$$

- Para $i = 2, \dots, n_b$ os elementos são:

- Para $l < i - 1$:

$$da_{i,j} = 0;$$

- Para $l = i - 1$:

$$da_{i,j} = -\gamma_l(1 - \gamma_j^2)^{-1/2}, \text{ para } j = i - 1;$$

$$da_{i,j} = -\gamma_j, \text{ para } j = i;$$

$$da_{i,j} = -\gamma_j \prod_{k=i}^{j-1} \sqrt{1 - \gamma_k^2}, \text{ para } j = i + 1, \dots, n_b;$$

$$da_{i,j} = 0, \text{ para os demais elementos}$$

- Para $l = i$ e $l \neq n_b$:

$$da_{i,j} = 0, \text{ para } j = i - 1;$$

$$da_{i,j} = -\gamma_{j-1}, \text{ para } j = i;$$

$$da_{i,j} = -\gamma_{i-1}\gamma_j\gamma_l(1-\gamma_l^2)^{1/2} \prod_{\substack{k=i \\ k \neq l}}^{j-1} \sqrt{1-\gamma_k^2}, \text{ para } j = i+1, \dots, n_b;$$

$$da_{i,j} = 0, \text{ para os demais elementos}$$

– Para $l = n_b$:

$$da_{i,j} = 0, \text{ para } j = i-1;$$

$$da_{i,j} = -\gamma_{j-1}, \text{ para } j = i \text{ e } i = n_b;$$

$$da_{i,j} = -\gamma_{i-1} \prod_{k=i}^{j-1} \sqrt{1-\gamma_k^2}, \text{ para } j = n_b \text{ e } i \neq n_b;$$

$$da_{i,j} = 0, \text{ para os demais elementos.}$$

A matriz B_b do modelo (2.24) é definida por

$$\frac{\partial B_b}{\partial \gamma_l} = \begin{bmatrix} db_{1,1} \\ db_{2,1} \\ db_{3,1} \\ \vdots \\ db_{n_b,1} \end{bmatrix}, \quad (\text{A.6})$$

cujos parâmetros são dados por

• Para $i = 1$ os elementos são:

– Para $l = 1, \dots, n_b$:

$$db_{1,1} = \gamma_l(1-\gamma_l^2)^{-1/2} \prod_{\substack{k=1 \\ k \neq l}}^{n_k} \sqrt{1-\gamma_k^2}, \text{ para } j = 1$$

• Para $l = 2, \dots, n_b$ os elementos são:

– Para $l < i-1$:

$$db_{i,1} = 0;$$

– Para $l = i-1$:

$$db_{i,i} = -\prod_{k=i}^{n_b} \sqrt{1-\gamma_k^2};$$

– Para $l \geq i$:

$$db_{i,1} = -\gamma_{i-1}\gamma_l(1 - \gamma_l^2)^{-1/2} \prod_{\substack{k=i \\ k \neq l}}^{n_k} \sqrt{1 - \gamma_k^2}, \text{ com } n_k = n_b, \text{ se } l \neq n_b \text{ e } n_k = n_b - 1$$

se $l = n_b$.

A matriz A_I também compõe a matriz A_f , presente no modelo GOBF apresentado em (2.24), e assim deve ter suas derivadas calculadas com relação ao parâmetros γ_l para compor o cálculo de $\frac{\partial A_f}{\partial \gamma_l}$ que aparece na equação (A.3). A matriz $\frac{\partial A_I}{\partial \gamma_l}$ é dada por

$$\frac{\partial A_I}{\partial \gamma_l} = \begin{bmatrix} dai_{1,1} & dai_{1,2} & dai_{1,3} & \cdots & dai_{1,n_b} \\ dai_{2,1} & dai_{2,2} & dai_{2,3} & \cdots & dai_{2,n_b} \\ dai_{3,1} & dai_{3,2} & dai_{3,3} & \cdots & dai_{3,n_b} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ dai_{n_b,1} & dai_{n_b,2} & dai_{n_b,3} & \cdots & dai_{n_b,n_b} \end{bmatrix} \quad (\text{A.7})$$

com seus elementos $(dai_{i,j})$ dados de acordo com o γ a que se refere a derivada, no qual o índice i indica a linha da matriz e o índice j a coluna:

- Para $i = 1, \dots, n_b$ e $j \neq n_b$ os elementos serão:

$$dai_{i,j} = 0;$$

- Para $j = n_b$ os elementos serão:

Com $i = 1$:

– Para $l = 1, \dots, n_b - 1$:

$$dai_{i,j} = (-\gamma_l)(1 - \gamma_l^2)^{-1/2}(1 - \gamma_{n_b}^2) \prod_{\substack{k=i \\ k \neq l}}^{n_b-1} \sqrt{1 - \gamma_k^2};$$

– Para $l = n_b$:

$$dai_{i,j} = -2\gamma_l \prod_{k=i}^{n_b-1} \sqrt{1 - \gamma_k^2}.$$

Com $i = 2, \dots, n_b$:

– Para $l = i - 1$:

$$dai_{i,j} = -(1 - \gamma_{n_b}^2) \prod_{k=i}^{n_b-1} \sqrt{1 - \gamma_k^2};$$

– Para $l = i, \dots, n_b - 1$:

$$dai_{i,j} = (-\gamma_l)(1 - \gamma_l^2)^{-1/2}(1 - \gamma_{n_b}^2)(-\gamma_{i-1}) \prod_{\substack{k=i \\ k \neq l}}^{n_b-1} \sqrt{1 - \gamma_k^2};$$

– Para $l = n_b$:

$$dai_{i,j} = (-2\gamma_{n_b})(-\gamma_{i-1}) \prod_{l=i+1}^{n_b-1} \sqrt{1 - \gamma_l^2};$$

– Para os demais elementos

$$dai_{i,j} = 0.$$

Utilizando os valores de $\frac{\partial A_b}{\partial \gamma_l}$ e $\frac{\partial A_I}{\partial \gamma_l}$ é possível realizar o cálculo da matriz $\frac{\partial A}{\partial \gamma_l}$ e, em sequência, determinar $\frac{\partial \Psi(k)}{\partial \gamma_l}$. Com n sendo o número de blocos G_b utilizados no modelo, o cálculo da matriz $\frac{\partial A_f}{\partial \gamma_l}$ pode ser dado por

- Para $l \neq n_b$ os elementos serão:

$$\frac{\partial A_f}{\partial \gamma_l} = \begin{bmatrix} \frac{\partial A_b}{\partial \gamma_l} & 0 & 0 & \dots & 0 \\ \frac{\partial A_I}{\partial \gamma_l} & \frac{\partial A_b}{\partial \gamma_l} & 0 & \dots & 0 \\ (-\gamma_{n_b}) \frac{\partial A_I}{\partial \gamma_l} & \frac{\partial A_I}{\partial \gamma_l} & \frac{\partial A_b}{\partial \gamma_l} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (-\gamma_{n_b})^{n-2} \frac{\partial A_I}{\partial \gamma_l} & (-\gamma_{n_b})^{n-3} \frac{\partial A_I}{\partial \gamma_l} & (-\gamma_{n_b})^{n-4} \frac{\partial A_I}{\partial \gamma_l} & \dots & \frac{\partial A_b}{\partial \gamma_l} \end{bmatrix} \quad (\text{A.8})$$

- Se $l = n_b$:

$$\frac{\partial A_f}{\partial \gamma_l} = \begin{bmatrix} \frac{\partial A_b}{\partial \gamma_l} & 0 & 0 & \dots & 0 \\ \frac{\partial A_I}{\partial \gamma_l} & \frac{\partial A_b}{\partial \gamma_l} & 0 & \dots & 0 \\ ta_{3,1} & \frac{\partial A_I}{\partial \gamma_l} & \frac{\partial A_b}{\partial \gamma_l} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ ta_{n,1} & ta_{n,2} & ta_{n,3} & \dots & \frac{\partial A_b}{\partial \gamma_l} \end{bmatrix} \quad (\text{A.9})$$

na qual:

$$ta_{i,j} = \left((-1)(i - j - 1)(-\gamma_{n_b}^{(i-j-2)}) A_I + (-\gamma_{n_b})^{i-(j+1)} \frac{\partial A_I}{\partial \gamma_l} \right)$$

para valores de $i > 2$ e $j < i - 1$.

O cálculo de $\frac{\partial B_f}{\partial \gamma_l}$ é dado por

- Para $l \neq n_b$ os elementos serão:

$$\frac{\partial B_f}{\partial \gamma_l} = \begin{bmatrix} \frac{\partial B_b}{\partial \gamma_l} \\ (-1)B_b + (-\gamma_{n_b})\frac{\partial B_b}{\partial \gamma_l} \\ (-1)(2)(-\gamma_{n_b})B_b + (-\gamma_{n_b})^2\frac{\partial B_b}{\partial \gamma_l} \\ \vdots \\ (-1)(n_b - 1)(-\gamma_{n_b})^{n_b-2}B_b + (-\gamma_{n_b})^{n-1}\frac{\partial B_b}{\partial \gamma_l} \end{bmatrix}; \quad (\text{A.10})$$

- Se $l = n_b$:

$$\frac{\partial B_f}{\partial \gamma_l} = \begin{bmatrix} \frac{\partial B_b}{\partial \gamma_l} \\ (-\gamma_{n_b})\frac{\partial B_b}{\partial \gamma_l} \\ (-\gamma_{n_b})^2\frac{\partial B_b}{\partial \gamma_l} \\ \vdots \\ (-\gamma_{n_b})^{n-1}\frac{\partial B_b}{\partial \gamma_l} \end{bmatrix}. \quad (\text{A.11})$$

De posse de todos esses valores, é possível obter o gradiente de $\Psi(k)$ com relação aos parâmetros γ_l apresentado na equação (3.25) e utilizar o cálculo exato do gradiente no método de otimização não linear empregado ao longo do trabalho.

APÊNDICE B – Seleção de Modelos - Critérios de Informação

Dentre as várias técnicas propostas para a seleção de modelos e estruturas no processo de representação de sistemas dinâmicos, uma amplamente utilizada é chamada de Critério de Informação. O critério de informação (B.1) leva em consideração uma função de custo e a complexidade do modelo em análise.

$$\text{critério de informação} = IC(\text{função de custo, complexidade do modelo}) \quad (\text{B.1})$$

O melhor modelo seria o com menor valor de critério de informação. Em vias gerais, não existe uma única proposta para a função (B.1). São opções válidas funções em que a penalidade pela complexidade aumenta com o número de parâmetros e decaem com o aumento do número de amostras N (NELLES, 2001).

Várias propostas para o termo de complexidade foram feitas ao longo dos anos, destacam-se algumas:

- Critério de informação de Akaike (AIC):

$$AIC(\rho, n_\theta) = N \ln(I(\underline{\theta})) + \rho n_\theta. \quad (\text{B.2})$$

A escolha mais usual para ρ é 2.

- Critério de informação de Bayes (BIC):

$$BIC(n_\theta) = N \ln I(\underline{\theta}) + \ln(N)n_\theta. \quad (\text{B.3})$$

- Erro final de predição (FPE):

$$FPE(\rho, n_\theta) = N \ln(I(\underline{\theta})) + 2\rho \ln(\ln(N))n_\theta. \quad (\text{B.4})$$

N é o número de amostras, n_θ é o número de parâmetros e a função de custo $I(\underline{\theta})$ é definida por (B.5).

$$I(\underline{\theta}) = \frac{1}{N} \sum_{i=1}^N e^2(i) \quad \text{com } e(i) = y(i) - \hat{y}(i). \quad (\text{B.5})$$

É importante frisar que esses critérios são fundamentalmente estatísticos e não garantem que o modelo com o número “ótimo” de termos, entre os candidatos, seja um modelo válido (AGUIRRE, 2007b).

Referências

- ABBAS, H. A.; ZEGNINI, B.; BELKHEIRI, M.; ABHI, A. Radial basis function neural network-based adaptive control of uncertain nonlinear systems. In: *3rd International Conference on Control, Engineering and Information Technology (CEIT)*. [S.l.]: IEEE, 2015.
- AGUIRRE, L. A. (Ed.). *Enciclopédia de Automática: Controle e Automação, Volume III*. 1. ed. [S.l.]: Editora Blucher, 2007.
- AGUIRRE, L. A. *Introdução à Identificação de Sistemas*. 3. ed. [S.l.]: Editora UFMG, 2007.
- AHMAD, I.; JAVAID, M. A. Nonlinear model & controller design for magnetic levitation system. In: *Recent Advances in Signal Processing, Robotics and Automation*. [S.l.]: WSEAS Press, 2010. p. 324–328.
- AKAIKE, H. Stochastic theory of minimal realization. *IEEE Trans. Automatic Control*, n. 26, p. 667–673, 1974.
- BABUSKA, R.; VEEN, P. Van der; KAYMAK, U. Improved covariance estimation for gustafson-kessel clustering. In: *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems*. [S.l.]: IEEE, 2002. p. 1081 – 1085.
- BACK, T.; FOGEL, D. B.; MICHALEWICZ, Z. *Evolutionary Computation 1: Basic Algorithms and Operators*. 1. ed. [S.l.]: Bristol and Philadelphia: Institute of Physics Publishing, 2000.
- BEZDEK, J. C. A convergence theorem for the fuzzy isodata clustering algorithms. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, v. 2, n. 1, p. 1 – 8, 1980.
- BEZDEK, J. C. *Pattern Recognition with Fuzzy Objective Function Algorithms*. [S.l.]: Plenum Press, 1981.
- BLACKWELL, T. M.; BRANKE, J. Multi-swarm optimization in dynamic environments. In: *Applications of Evolutionary Computing - LNCS 3005*. [S.l.: s.n.], 2004. p. 489–500.
- BOKOR, J.; SCHIPP, F. Non-linear adaptive control via Laguerre expansion of Volterra kernels. *International Journal of Adaptive Control and Signal Processing*, v. 7, p. 463–468, 1998.
- BROOME, P. W. Discrete orthonormal sequences. *Journal of the Association for Computing Machinery*, v. 12, n. 2, p. 151–168, 1965.
- BROOMHEAD, D. S.; LOWE, D. Multivariable functional interpolation and adaptive networks. *Complex Systems Publications, Inc.*, n. 2, p. 321 – 355, 1988.
- CAMPELLO, R. J. G. B.; AMARAL, W. C. Takagi-Sugeno fuzzy models within orthonormal basis function framework and their application to process control. In: *IEEE Int. Conf. Fuzzy Systems*. [S.l.]: IEEE, 2002. p. 1399–1404.

- CAMPELLO, R. J. G. B.; FAVIER, G.; AMARAL, W. C. Optimal expansions of discrete-time Volterra models using Laguerre functions. In: *Proc. 13th IFAC Symposium on System Identification*. [S.l.: s.n.], 2003.
- CAMPELLO, R. J. G. B.; OLIVEIRA, G. H. C.; AMARAL, W. C. Identificação e controle de processos via desenvolvimento em séries ortonormais: Parte A (identificação) e B (controle). *Controle e Automação*, v. 18, n. 3, p. 298–332, 2007.
- CHEN, J. Y.; QIN, Z. Training RBF neural networks with PSO and improved subtractive clustering algorithms. In: *Lecture Notes in Computer Science*. [S.l.]: Springer-Verlag, 2006. v. 4233, p. 1148–1155.
- CHEN, S.; BILLINGS, S. A.; COWAN, C. F. N.; GRANT, P. M. Nonlinear systems identification using radial basis functions. *Int. J. Systems Science*, v. 21, p. 2513–2539, 1990.
- CHEN, S.; BILLINGS, S. A.; COWAN, C. F. N.; GRANT, P. M. Practical identification of NARMAX models using radial basis functions. *Int. J. of Control*, v. 52, p. 1327–1350, 1990.
- CHEUNG, N. C.; SUN, Z. C.; ZHAO, S. W.; GAN, W. C. Design and simulation of a testing fixture for planar magnetic levitation system control using switched reluctance actuator. In: *Proceedings in the 3rd International Conference on Power Electronics Systems and Applications*. [S.l.]: IEEE, 2009. p. 1–6.
- CHIU, S. L. A cluster extension method with extension to fuzzy model identification. In: *Proceedings of the Third IEEE Conference on Fuzzy Systems*. [S.l.]: IEEE, 1994. p. 1240–1245.
- CHIU, S. L. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, v. 2, n. 3, p. 267–278, 1994.
- CLERC, M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In: *Proceedings of the Congress on Evolutionary Computation*. [S.l.: s.n.], 1999. p. 1951–1957.
- CLERC, M.; KENNEDY, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 1, p. 58–73, 2002.
- COELLO, C. A. C.; LECHUNGA, M. S. MOPSO: A proposal for multiple objective particle swarm optimization. In: *Proceeding of Congress Evolutionary Computation, Vol 1, CEC 2002*. [S.l.: s.n.], 2002. p. 1051–1056.
- DUMONT, G. A.; FU, Y. Non-linear adaptive control via Laguerre expansion of Volterra kernels. *International Journal of Adaptive Control and Signal Processing*, v. 7, p. 367–382, 1993.
- EYKHOFF, P. *System identification: parameter and state estimation*. 1. ed. [S.l.]: John Wiley & Sons, 1974.
- FAUSTINO, C. P. *Previsão de Séries Temporais via Modelos Baseados em Regras*. Dissertação (Mestrado) — IESTI/UNIFEI, Itajubá/MG/Brazil, 2011.

- FREEMAN, J. A. S.; SAAD, D. Learning and generalization in radial basis function networks. *Neural Computation*, v. 5, n. 7, p. 1000–1020, 1995.
- GAN, M.; CHEN, C. L. P.; LI, H.-X.; CHEN, L. Gradient radial basis function based varying-coefficient autoregressive model for nonlinear and nonstationary time series. *Signal Processing Letters*, v. 22, n. 7, p. 809–812, 2015.
- GAO, F.; Z.-Q., L.; TONG, H.-Q. Parameters estimation on-line for lorenz system by a novel quantum-behaved particle swarm optimization. *Chinese Physics B*, v. 17, n. 4, p. 1196–1201, 2008.
- GATH, I.; GEVA, A. Unsupervised optimal fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 17, n. 11, p. 773–781, 1989.
- GRAY, J. A.; MARKEL, J. A normalized digital filter structure. *IEEE Transactions on Acoustics, Speech and Signal Processing*, v. 23, n. 3, p. 268–277, 1975.
- GUSTAFSON, D. E.; KESSEL, W. C. Fuzzy clustering with a fuzzy covariance matrix. In: *Proceedings on the 1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*. [S.l.]: IEEE, 1979. p. 761–766.
- HAYKIN, S. *Neural Networks, A Comprehensive Foundation*. 2. ed. [S.l.]: Prentice Hall, 1999.
- HELLENDOORN, H.; DRIANKOV, D. *Fuzzy Model Identification: Selected Approaches*. [S.l.]: Springer-Verlag, 1997.
- HEUBERGER, P. S. C.; HOF, P. M. J. Van den; BOSGRA, O. H. A generalized orthonormal basis for linear dynamical systems. *Transactions on Automatic Control*, v. 40, n. 3, p. 451–465, 1995.
- HEUBERGER, P. S. C.; HOF, P. M. J. Van den; WAHLBERG, B. *Modelling and Identification with Rational Orthogonal Basis Functions*. [S.l.]: Springer-Verlag, 2005.
- HOF, P. M. J. Van den; HEUBERGER, P. S. C.; BOKOR, J. System identification with generalized orthonormal basis function. *Automatica*, v. 31, n. 12, p. 1821–1834, 1995.
- HOLLAND, J. *Adaptation in Natural and Artificial Systems*. [S.l.]: University of Michigan Press, 1975.
- JANG, J. S. R.; SUN, C. T.; MIZUTANI, E. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. 1. ed. [S.l.]: Prentice Hall, 2007.
- KARAYIANNIS, N. Gradient descent learning of radial basis neural networks. In: *Proceedings of the IEEE International Conference on Neural Networks*. [S.l.]: IEEE, 1997. p. 1815–1820.
- KENNEDY, J.; EBERHART, R. C. Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*. [S.l.]: IEEE, 1995. p. 1942–1948.
- LAZAR, M. *Nonlinear controller based on the EP-SAC approach*. Dissertação (Mestrado) — University of Ghent, Departament of Control, Engineering and Automation, Flanders, Belgium, 2001.

- LEVENBERG, K. A method for the solution of certain nonlinear problems in least square. *Quarterly of Applied Mathematics*, v. 2, n. 2, p. 164 – 168, 1944.
- LI, S.; WANG, R. G.; HU, W. W.; SUN, J. Q. A new QPSO based BP neural network for face detection. In: *Proceedings of the Second International Conference of Fuzzy Information and Engineering (ICFIE)*. [S.l.: s.n.], 2007. p. 355–363.
- LIM, J.; KIM, C.; LEE, J.; HAN, H.; PARK, D. Design of magnetic levitation electro-magnet for high speed maglev train. In: *Proceedings in the 2013 International Conference on Electrical Machines and Systems*. [S.l.]: IEEE, 2013. p. 1975–1977.
- LIMA, I.; PINHEIRO, C. A. M.; SANTOS, F. A. O. *Inteligência Artificial*. 1. ed. [S.l.]: Elsevier, 2014.
- LINDSKOG, P. *Methods, Algorithms and Tools for System Identification Based on Prior Knowledge*. Tese (Doutorado) — Linköping University, Sweden, 1996.
- LIU, J.; SUN, J.; XU, W. B. Quantum-behaved particle swarm optimization with adaptive mutation operator. In: *Proceeding of Advances in Natural Computation: Second International Conference, ICNC 2006, Part 1*. [S.l.: s.n.], 2006. p. 959–967.
- LJUNG, L. *System identification: Theory for the user*. 2. ed. [S.l.]: Prentice Hall, 1999.
- LOWE, D. Adaptive radial basis function nonlinearities, and the problem of generalization. In: *Proceedings In the First IEE International Conference on Artificial Neural Networks*. [S.l.]: IEEE, 1989. p. 171–175.
- LUENBERGER, D. G. *Linear and Nonlinear Programming*. 2. ed. [S.l.]: Spring-Verlag, 2003.
- MACHADO, J. B. *Modelagem e Controle Preditivo utilizando Multimodelos*. Dissertação (Mestrado) — DCA/FEEC/UNICAMP, Campinas/SP/Brazil, 2007.
- MACHADO, J. B. *Modelagem de Sistemas Não-lineares por Base de Funções Ortonormais Generalizadas com Funções Internas*. Tese (Doutorado) — Universidade de Campinas, Campinas/SP/Brazil, 2011.
- MACHADO, J. B. GA optimization of ladder-structured GOBF models. In: *8th Annual IEEE Systems Conference (SysCon)*. [S.l.]: IEEE, 2014. p. 396–401.
- MACHADO, J. B.; CAMPELLO, R. J. G. B.; AMARAL, W. C. Takagi-sugeno fuzzy models in the framework of orthonormal basis functions. *IEEE Transactions on Cybernetics*, v. 43, n. 3, p. 858–870, 2013.
- MARQUARDT, D. An algorithm for the least-square estimation of nonlinear parameters. *Journal of Applied Mathematics*, v. 11, n. 2, p. 431 – 441, 1963.
- MBAREK, A.; MESSAOUD, H.; FAVIER, G. Robust predictive control using Kautz model. In: *ICECS - 10th IEEE International Conference on Electronics, Circuits and Systems*. [S.l.]: IEEE, 2003. p. 184–187.
- MEDEIROS, A. V.; AMARAL, W.; CAMPELLO, R. J. G. B. GA optimization of OBF-TS fuzzy models with linear and non linear local models. In: *Ninth Brazilian Symposium on Neural Networks*. [S.l.]: SBRN, 2006. p. 66–71.

- MOODY, J.; DARKEN, C. Fast learning in networks of locally-tuned processing units. *Neural Computation*, v. 1, n. 2, p. 281–294, 1989.
- NELLES, O. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Model*. [S.l.]: Springer-Verlag, 2001.
- NINNESS, B.; GUSTAFSSON, F. A unifying construction of orthonormal bases for system identification. *IEEE Transactions on Automatic Control*, v. 42, n. 4, p. 515–521, 1997.
- NOCEDAL, J.; WRIGHT, S. J. *Numerical Optimization*. [S.l.]: Springer Verlag, 1999.
- OLIVEIRA, G. H. C. *Controle Preditivo para Processos com incertezas estruturadas baseado em séries de funções ortonormais*. Tese (Doutorado) — DCA/FEEC/UNICAMP, Campinas/SP/Brazil, 1997.
- OLIVEIRA, G. H. C.; AMARAL, W. C.; FAVIER, G.; DUMONT, G. A. Constrained robust predictive controller for uncertain processes modeled by orthonormal series functions. *Automatica*, v. 36, n. 4, p. 563–571, 2000.
- OLIVEIRA, G. H. C.; ROSA, A. da; CAMPELLO, R. J. G. B.; MACHADO, J. B.; AMARAL, W. C. An introduction to models based on Laguerre, Kautz and other related orthonormal functions - part I: Linear and uncertain models. *International Journal of Modelling, Identification and Control*, v. 14, p. 121–132, 2011.
- OMKAR, S. N.; KHANDELWAL, R.; ANANTH, T.; NAIK, G. N.; GOPALAKRISHNAN, S. Quantum behaved particle swarm optimization (QPSO) for multi-objective design optimization of composite structures. *Expert Systems with Applications*, v. 36, n. 8, p. 11312–11322, 2009.
- POWELL, M. J. D. Radial basis function approximations to polynomials. In: *Proc. in the 12th Biennial Numerical Analysis Conf.* [S.l.: s.n.], 1987. p. 223–241.
- QUANSER. *User Manual: Magnetic Levitation Experiment*. [S.l.], 2012.
- ROSA, A. da. *Identificação de Sistemas Não-Lineares Usando Modelos de Volterra Baseados em Funções Ortonormais de Kautz e Generalizadas*. Tese (Doutorado) — DCA/FEEC/UNICAMP, Campinas/SP/Brazil, 2009.
- 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. *Automatica*, v. 54, n. 12, p. 2757–2772, 2009.
- SCHETZEN, M. *The Volterra and Wiener Theories of Nonlinear Systems*. [S.l.]: John Wiley and Sons, 1980.
- SHI, Y.; EBERHART, R. A modified particle swarm. In: *Proceedings of 1998 IEEE International Conference on Evolutionary Computation*. [S.l.]: IEEE, 1998. p. 1945–1950.
- SHU, G.; MEISINGER, R. State estimation and simulation of the magnetic levitation system of a high-speed maglev train. In: *Proceedings in the 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*. [S.l.]: IEEE, 2011. p. 944–947.

- SUN, J.; FANG, W.; WU, X.; PALADE, V.; XU, W. Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection. *Evolutionary Computation*, v. 20, n. 3, p. 349–393, 2012.
- SUN, J.; FENG, B.; XU, W. B. Particle swarm optimization with particle having quantum behavior. In: *Proceedings of 2004 Congress on Evolutionary Computation*. [S.l.: s.n.], 2004. p. 325–331.
- SUN, J.; LAI, C.; WU, X. *Particle Swarm Optimization: Classical and Quantum Perspectives*. 1. ed. [S.l.]: CRC Press, 2012.
- SUN, J.; XU, W. B.; FENG, B. A global search strategy of quantum-behaved particle swarm optimization. In: *Proceedings of 2004 IEEE Conference on Cybernetics and Intelligent Systems*. [S.l.: s.n.], 2004. p. 111–116.
- SUSTER, P.; JADLOVSKA, A. Modeling and control design of magnetic levitation system. In: *Proceedings in the 10th IEEE Jubilee International Symposium on Applied Machine Intelligence and Informatics*. [S.l.]: IEEE, 2012. p. 295–299.
- TAO, X.; RENFU, L.; MYONGCHOL, T.; JUNCHOL, R.; GYANGHYOK, H. Trajectory planning for autonomous mobile robot using a hybrid improved QPSO algorithm. *Soft Computing*, v. 21, n. 9, p. 2421–2437, 2017.
- TSEKOURAS, G. E.; TSIMIKAS, J. On training RBF neural networks using input-output fuzzy clustering and particle swarm optimization. *Fuzzy Sets and Systems*, v. 221, p. 65 – 89, 2013.
- WAHLBERG, B. System identification using Laguerre models. *IEEE Transactions on Automatic Control*, v. 36, n. 5, p. 551 – 562, 1991.
- WAHLBERG, B. System identification using Kautz models. *IEEE Transactions on Automatic Control*, v. 39, n. 6, p. 1276 – 1282, 1994.
- WAHLBERG, B.; MAKILA, P. M. Approximation of stable linear dynamical systems using Laguerre e Kautz functions. *Automatica*, v. 32, n. 5, p. 693–708, 1996.
- YAGER, R.; FILEV, D. *Approximate clustering via the mountain method*. [S.l.], 1992. MII-1305.
- ZHAO, Z.; WU, X.; LU, C.; GLOTIN, H.; GAO, J. Optimizing widths with PSO for center selection of gaussian radial basis function networks. *Science China - Information Sciences*, v. 57, n. 5, p. 1–17, 2014.