

Representação de sistemas não-lineares invariantes no tempo através
do modelo de Volterra de segunda ordem com base de funções
ortonormais

Gabriel Santos Viana

TRABALHO DE CONCLUSÃO DE CURSO DE ENGENHARIA
ELÉTRICA

FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA

Universidade de Brasília
Faculdade de Tecnologia
Departamento de Engenharia Elétrica

Representação de sistemas não-lineares invariantes no tempo através
do modelo de Volterra de segunda ordem com base de funções
ortonormais

GABRIEL SANTOS VIANA

Trabalho final de graduação submetido ao Departamento de Engenharia Elétrica
da Faculdade de Tecnologia da Universidade de Brasília, como parte dos requisitos
necessários para a obtenção do grau de Engenheiro Eletricista.

APROVADA POR:

Prof. Dr. Alex da Rosa (ENE-UnB)
(Orientador)

Prof. Dr. José Alfredo Ruiz Vargas (ENE-UnB)
(Examinador Interno)

Prof. Dr. Renato Alves Borges (ENE-UnB)
(Examinador Interno)

Brasília/DF, Setembro de 2021.

FICHA CATALOGRÁFICA

VIANA, GABRIEL SANTOS

Representação de sistemas não-lineares invariantes no tempo através do modelo de Volterra de segunda ordem com base de funções ortonormais. [Distrito Federal] 2021.

xiii, 78p., 210 x 297 mm (ENE/FT/UnB, Engenheiro Eletricista, Engenharia Elétrica, 2021).

Trabalho de Graduação – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Volterra

3. NLIT

I. ENE/FT/UnB

2. GOBF

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

VIANA, G. S. (2021). Representação de sistemas não-lineares invariantes no tempo através do modelo de Volterra de segunda ordem com base de funções ortonormais, Trabalho de Graduação em Engenharia Elétrica, Publicação 2021, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, xp.

CESSÃO DE DIREITOS

AUTOR: Gabriel Santos Viana

TÍTULO: Representação de sistemas não-lineares invariantes no tempo através do modelo de Volterra de segunda ordem com base de funções ortonormais.

GRAU: Engenheiro Eletricista ANO: 2021

É concedida à Universidade de Brasília permissão para reproduzir cópias deste trabalho de graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse trabalho de graduação pode ser reproduzida sem autorização por escrito do autor.

Gabriel Santos Viana

Departamento de Eng. Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

AGRADECIMENTOS

Agradeço a Deus por ter me permitido essa grande oportunidade que foi ter acesso a um bom curso, em uma ótima universidade e também por ter me dado forças durante essa caminhada.

À minha família, minha mãe Odésia, meu pai Gleivam e meu irmão Matheus, que são meus pilares e principais apoiadores não só durante o curso, mas em todos os momentos vividos até aqui.

Às novas amizades feitas na UnB, que sabem quem são, e foram fundamentais em todos os momentos dessa jornada, seja em festas, que felizmente foram várias, seja em vésperas de provas, seja na elaboração desse trabalho.

Aos outros amigos, desde os amigos de infância, do Colégio Vovó Olívia, do Leonardo da Vinci, do Olimpo e também os conhecidos em tantas outras ocasiões que também me ajudaram em tantos momentos.

A meus professores, que buscaram transmitir da melhor forma o conhecimento necessário para o desenvolvimento durante o curso e, em especial, ao professor Alex que sempre se mostrou muito colaborativo e paciente durante o desenvolvimento desse trabalho final.

Aos grupos extracurriculares dos quais fiz parte, Piratas do Cerrado, CAENE e, principalmente, Enactus UnB, que me fizeram evoluir e me ensinaram o significado de resiliência.

A todos demais que esteja esquecendo de mencionar, muito obrigado.

RESUMO

A identificação de sistemas dinâmicos não-lineares invariantes no tempo através do modelo de Volterra de segunda ordem com base de funções ortonormais na representação dos *kernels* é um tópico amplamente discutido na literatura para dois tipos de bases, a de Laguerre e a de Kautz. Assim, a proposta de utilização da base de funções ortonormais generalizadas com um pólo real e um par de pólos conjugados se mostrou um tópico a ser explorado, especialmente para sistemas cuja dinâmica é de ordem superior a dois.

A reação de polimerização em um reator ideal de tanque agitado, *Continuous Stirred Tank Reactor* (CSTR), é um exemplo de planta industrial que atende a esses requisitos e, já tendo sido utilizado em diversos trabalhos como fonte de dados de entrada e de saída (I/O), se mostrou um bom estudo de caso.

Adotou-se na construção do algoritmo uma metodologia de busca exaustiva, em que foram testados todos os elementos de um dado conjunto de pólos que melhor identificavam a saída do reator durante a primeira metade da reação de polimerização. Com os *kernels* de primeira e segunda ordem obtidos, fez-se uma validação do modelo utilizando um outro conjunto de dados, no caso, a segunda metade do processo de reação.

São comparados os resultados obtidos utilizando os três tipos de base de funções e discutidos, além da apresentação de propostas futuras e de melhorias da metodologia e do algoritmo.

Palavras-chave: GOBF, Volterra, identificação de sistemas NLIT, *kernel*, CSTR

ABSTRACT

The identification of non-linear time invariant dynamic systems through second order Volterra models with orthonormal basis functions to reproduce the kernels is a topic well discussed in the literature for two particular function basis: Laguerre and Kautz. So, the representation through generalised orthonormal basis functions is a issue worth of discussion, specially for systems with more complex dynamic.

The continuous stirred tank reactor (CSTR) polymerisation process is a common case of study and is used as source for input and output data.

During the algorithm project, an exhausted search approach was adopted, where all the elements were tested of a certain set of poles that better estimate the output during the first half of the polymerisation process. Then, using the first-order and second-order kernels obtained with the second half of the data, the model was validated

The results obtained using these three classes of orthonormal basis functions were compared and discussed. Also, suggestions for improvements and for future works were presented.

Keywords: GOBF, Volterra, identification of NLTI systems, kernel, CSTR

SUMÁRIO

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Símbolos	vi
Glossário	vii
Capítulo 1 – Introdução	1
1.1 Motivação	1
1.2 Organização do Texto	2
Capítulo 2 – Modelos de Volterra Baseados em Funções Ortonormais	4
2.1 Modelo de Volterra	4
2.2 Base de Funções Ortonormais	6
2.2.1 Uso de Funções Ortonormais	6
2.2.2 Funções de Laguerre	7
2.2.3 Funções de Kautz	8
2.2.4 Base de Funções Ortonormais Generalizadas	10
2.3 Modelo OBF-Volterra	13
Capítulo 3 – Seleção dos pólos das bases de funções ortonormais	16
3.1 Geração de funções ortonormais	16
3.2 Estimação dos coeficientes via mínimos quadrados e cálculo dos <i>kernels</i>	17
3.3 Seleção do melhor conjunto de pólos e validação	19
Capítulo 4 – Modelagem de uma reação de polimerização em um CSTR	21
4.1 Reação de polimerização em um CSTR	21
4.2 Apresentação e discussão de resultados: Laguerre	24

4.3	Apresentação e discussão de resultados: Kautz	28
4.4	Apresentação e discussão de resultados: GOBF	32
4.5	Comparação de resultados entre as três bases propostas	35
Capítulo 5 – Conclusões e Propostas Futuras		38
Referências Bibliográficas		40
Apêndice A – Códigos		43
A.1	Geração da base de funções ortonormais	43
A.2	Análise de convergência	45
A.3	Filtragem da função de entrada pela base de funções ortonormais	46
A.4	Obtenção dos coeficientes de expansão via método de mínimos quadrados	47
A.5	Geração de todos os kernels de primeira e de segunda ordem para o domínio de pólos adotado	47
A.6	Seleção dos pólos	48

LISTA DE FIGURAS

2.1	Comportamento temporal das 6 primeiras funções da base de Laguerre com pólo $p = 0,8$	8
2.2	Comportamento temporal das 6 primeiras funções da base de Kautz com par de pólos $\beta = 0,8 \pm 0,4i$	10
2.3	Comportamento temporal das 6 primeiras funções da GOBF com 1 pólo real $p = 0,8$ e 1 par de conjugados $\beta = 0,8 \pm 0,4i$	13
4.1	Função de saída do CSTR para sequência de funções degrau de pequena amplitude	23
4.2	Kernel de 1ª ordem obtido para base Laguerre com pólo $p = 0,9113$	24
4.3	Kernel de 2ª ordem obtido para base Laguerre com pólo $p = 0,9113$	25
4.4	Função de saída do CSTR durante as primeiras 16 horas (linha contínua) e saída obtida a partir do modelo de Volterra com as funções de Laguerre (linha com símbolo)	26
4.5	Função de saída do CSTR durante as 16 horas restantes (linha contínua) e saída obtida a partir do modelo de Volterra com as funções de Laguerre (linha com símbolo)	26
4.6	Erro entre saída do sistema $y(k)$ e saída do modelo $\hat{y}(k)$ utilizando funções de Laguerre	27
4.7	Kernel de 1ª ordem obtido para base Kautz com pólo $\beta = 0,9000$	28
4.8	Kernel de 2ª ordem obtido para base Kautz com par de conjugados $\beta = -0,7000 \pm 0,7000i$	29
4.9	Função de saída do CSTR durante as primeiras 16 horas (linha contínua) e saída obtida a partir do modelo de Volterra com as funções de Kautz (linha com símbolo)	30

4.10	Função de saída do CSTR durante as 16 horas restantes (linha contínua) e saída obtida a partir do modelo de Volterra com as funções de Kautz (linha com símbolo)	30
4.11	Erro entre saída do sistema $y(k)$ e saída do modelo $\hat{y}(k)$ utilizando funções de Kautz	31
4.12	<i>Kernel</i> de 1ª ordem obtido para GOBF com pólos $p_{r_1} = 0,8930$ e $p_{c_1} = 0,9680 \pm 0,0060i$	32
4.13	<i>Kernel</i> de 2ª ordem obtido para GOBF com pólos $p_{r_2} = 0,8900$ e $p_{c_2} = 0,9680 \pm 0,0020i$	33
4.14	Função de saída do CSTR durante as primeiras 16 horas (linha contínua) e saída obtida a partir do modelo de Volterra com a GOBF proposta (linha com símbolo)	34
4.15	Função de saída do CSTR durante as 16 horas restantes (linha contínua) e saída obtida a partir do modelo de Volterra com a GOBF proposta (linha com símbolo)	34
4.16	Erro entre saída do sistema $y(k)$ e saída do modelo $\hat{y}(k)$ utilizando a GOBF proposta	35

LISTA DE TABELAS

4.1	Performance da simulação do modelo de OBF-Volterra com funções de Laguerre	27
4.2	Performance da simulação do modelo de OBF-Volterra com funções de Kautz . .	31
4.3	Performance da simulação do modelo de OBF-Volterra com a GOBF proposta .	35
4.4	Pólos obtidos que melhor identificam a função de saída do CSTR através do modelo OBF-Volterra para Laguerre, Kautz e GOBF proposta	36
4.5	Comparação dos erros EQM e EQN durante a etapa de identificação através do modelo OBF-Volterra para Laguerre, Kautz e GOBF proposta	36
4.6	Comparação dos erros EQM e EQN durante a etapa de validação do modelo OBF-Volterra para Laguerre, Kautz e GOBF proposta	36

LISTA DE SÍMBOLOS

h_1 Kernel de primeira ordem

h_2 Kernel de segunda ordem

Ψ função ortonormal

GLOSSÁRIO

AIBN	Azobisisobutironitrila
CSTR	<i>Continuous Stirred Tank Reactor</i>
EQM	Erro Quadrático Médio
EQN	Erro Quadrático Normalizado
GOBF	<i>Generalised Orthonormal basis of functions</i>
MMA	Metacrilato de Metila
NFIR	<i>Non-linear finite impulse response</i>
NLIT	Não linear invariante no tempo
OBF	<i>Orthonormal basis of functions</i>
SISO	<i>Single-Input Single-Output</i>

1.1 MOTIVAÇÃO

Desde a Primeira Revolução Industrial, iniciada com o advento da máquina a vapor, a humanidade registrou acelerado avanço tecnológico, chegando até a iniciar a exploração turística do espaço. E sistemas de controle foram e são parte fundamental desse processo, permitindo a representação e modelagem matemática de processos cada vez mais complexos que permitiram esse grande progresso (NISE, 2008).

Dentre as várias etapas necessárias para a modelagem matemática de sistemas de controle, uma delas é a adoção de uma metodologia que permita essa representação. As duas principais são caixa branca e caixa preta. Enquanto na primeira a construção é feita tomando-se os parâmetros físicos de um dado sistema, na segunda essa construção é feita através dos dados de entrada e dos dados de saída (LJUNG, 1999).

E para a modelagem matemática de um dado sistema, adota-se, de forma geral, algumas premissas que levam a uma simplificação do processo. E uma das simplificações mais adotadas consiste na adoção de premissas que permitam um sistema cuja dinâmica é não-linear, ser analisado e estudado como um sistema linear. Análise em torno de um ponto de equilíbrio, pequenas variações angulares, são exemplos de tais premissas.

Apesar dessa simplificação ter diversas vantagens, como utilização de ferramentas matemáticas mais simples, desvantagens como restrições na representação, perda de fidelidade em relação ao caso estudado, demandam o emprego de técnicas não-lineares (DRONGELEN, 2010).

Assim, a proposta deste trabalho é justamente dado um conjunto de dados de entrada e de saída (I/O) de sistema não-linear, ou seja, através da abordagem de caixa preta, fazer a construção de um modelo não-linear para representá-lo.

Dentre o vasto estudo dentro desse escopo, existem diferentes modelos matemáticos a serem

utilizados, cada qual com suas vantagens e desvantagens. Adotou-se para este trabalho a utilização do modelo de OBF-Volterra, que são modelos NFIR (*Non-linear finite impulse response*) com não-linearidade polinomial. Esse modelo tem por grandes vantagens que a sua representação se dá somente em termos de amostras passadas e é um modelo linear nos seus parâmetros. (HEUBERGER *et al.*, 2005).

O emprego de funções de Laguerre e de Kautz para identificação de sistemas não-lineares através do modelo OBF-Volterra já é bem discutido e consolidado, como em (HEUBERGER *et al.*, 2005; NINNESS; GUSTAFSSON, 1997); DOYLE *et al.*, 2002); DUMONT; FU, 1993); DA ROSA *et al.*, 2009); OLIVEIRA *et al.*, 2012)), de forma que a proposta do trabalho é justamente contribuir em uma nova linha de discussão, através da utilização de funções GOBF com três pólos, um real e um par de conjugados. Assim, se busca fazer uma comparação da precisão na identificação de um dado sistema não-linear utilizando essa GOBF proposta, com os resultados obtidos utilizando as outras bases, Laguerre e Kautz.

Para tal, foi desenvolvido um algoritmo descrito ao longo do trabalho e buscou-se empregá-lo de forma padronizada para essas três classes de bases de funções durante a identificação de uma reação de polimerização em um reator ideal de tanque agitado. Os dados de entrada e de saída utilizados e obtidos durante simulação dessa reação, foram utilizados como fonte para verificarmos a capacidade de identificação do modelo de OBF-Volterra utilizando cada uma das três bases, Laguerre, Kautz e a GOBF proposta utilizando o algoritmo desenvolvido.

1.2 ORGANIZAÇÃO DO TEXTO

Este trabalho encontra-se organizado da seguinte maneira:

- No capítulo 2, apresenta-se a fundamentação teórica do modelo de Volterra, das bases de funções ortonormais utilizadas e do modelo OBF-Volterra;
- No capítulo 3, é descrito o algoritmo adotado e a metodologia aplicada para melhor seleção dos pólos das bases de funções ortonormais;
- No capítulo 4, é descrito o estudo de caso e os resultados obtidos para a modelagem são apresentados e discutidos;
- Por fim, no capítulo 5, apresentam-se as conclusões deste trabalho, além de propostas

para trabalhos futuros.

MODELOS DE VOLTERRA BASEADOS EM FUNÇÕES ORTONORMAIS

2.1 MODELO DE VOLTERRA

O modelo de Volterra, desenvolvido pelo matemático italiano Vito Volterra no ano de 1930, é essencialmente uma forma de representação de sistemas não-lineares e invariantes no tempo (NLIT) através de uma sequência de operações de convolução ou similar a essas entre o sinal de entrada, $x(t)$, e as funções h_n , chamadas de *kernels* de Volterra, no domínio contínuo do tempo (DRONGELEN, 2010):

$$\begin{aligned}
 y(t) = & \int_{-\infty}^{+\infty} h_1(\tau_1)x(t - \tau_1)d\tau_1 \\
 & + \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2(\tau_1, \tau_2)x(t - \tau_1)x(t - \tau_2)d\tau_1d\tau_2 + \dots \\
 & + \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_n(\tau_1, \tau_2, \dots, \tau_n)x(t - \tau_1)x(t - \tau_2) \dots x(t - \tau_n)d\tau_1d\tau_2 \dots d\tau_n,
 \end{aligned} \tag{2.1}$$

sendo $y(t)$ a saída e o índice n dos *kernels* indicam a ordem desses.

Alguns autores, (BILLINGS, 1980); DUMONT; FU, 1993)) adotam a expressão com um h_0 que representa um *offset* do sistema.

Esse modelo também pode ser representado no domínio discreto do tempo (k) (SCHETZEN, 2006; RUGH, 1981)):

$$\begin{aligned}
 y(k) = & \sum_{\tau_1=-\infty}^{+\infty} h_1(\tau_1)x(k - \tau_1) \\
 & + \sum_{\tau_1=-\infty}^{+\infty} \sum_{\tau_2=-\infty}^{+\infty} h_2(\tau_1, \tau_2)x(k - \tau_1)x(k - \tau_2) + \dots \\
 & + \sum_{\tau_1=-\infty}^{+\infty} \sum_{\tau_2=-\infty}^{+\infty} \dots \sum_{\tau_n=-\infty}^{+\infty} h_n(\tau_1, \tau_2, \dots, \tau_n)x(k - \tau_1)x(k - \tau_2) \dots x(k - \tau_n).
 \end{aligned} \tag{2.2}$$

Por fins práticos, é adotado um limite ϵ_n de truncamento na representação, além do qual os *kernels* de ordem n são considerados nulos. Além disso, o limite inferior do somatório é $k_n = 0$,

uma vez que para uma representação suficientemente precisa, assume-se que o sistema é causal (BOYD; CHUA, 1985)) logo, com essas considerações a representação fica da seguinte forma:

$$\begin{aligned}
 y(k) &= \sum_{\tau_1=0}^k h_1(\tau_1)x(k - \tau_1) \\
 &+ \sum_{\tau_1=0}^k \sum_{\tau_2=0}^k h_2(\tau_1, \tau_2)x(k - \tau_1)x(k - \tau_2) + \dots \\
 &+ \sum_{\tau_1=0}^k \sum_{\tau_2=0}^k \dots \sum_{\tau_n=0}^k h_n(\tau_1, \tau_2, \dots, \tau_n)x(k - \tau_1)x(k - \tau_2) \dots x(k - \tau_n),
 \end{aligned} \tag{2.3}$$

com $k > \epsilon_n$.

Já foi estudado e comprovado que o truncamento do modelo de Volterra representa com precisão suficiente qualquer sistema não linear que satisfaça as seguintes condições (BOYD; CHUA, 1985)):

1. Seja causal, estável, invariante no tempo ;
2. A entrada seja limitada.

Assim, o sistema da equação (2.3) atende a essas condições.

Geralmente na literatura a expansão do modelo de Volterra é adotada até a segunda ordem ($n = 2$), tanto por razões práticas, como limitar número de variáveis a serem trabalhadas, e por razões teóricas, como aproximação suficientemente precisa (BILLINGS, 1980); DUMONT; FU, 1993)). Assim, a representação do modelo de Volterra adotada neste trabalho será:

$$y(k) = \sum_{\tau_1=0}^k h_1(\tau_1)x(k - \tau_1) + \sum_{\tau_1=0}^k \sum_{\tau_2=0}^k h_2(\tau_1, \tau_2)x(k - \tau_1)x(k - \tau_2). \tag{2.4}$$

Uma vez que a proposta de trabalho está limitada a sistemas do tipo SISO (*Single-Input Single-Output*), a função $x(k - \tau_n)$ representa diferentes amostras de uma mesma entrada, de tal forma que os *kernels* de segunda ordem, h_2 podem ser considerados simétricos, já que estão associados a um mesmo fator, ou seja, $h_2(\tau_1, \tau_2) = h_2(\tau_2, \tau_1)$ (SCHETZEN, 2006). Conforme será discutido de forma mais ampla no capítulo 3 uma das grandes vantagens desse modelo é a capacidade de representação de sistemas não lineares em um modelo cujos coeficientes são lineares, ou seja, mesmo com a relação não linear entre entrada, $x(k)$, e saída, $y(k)$, a relação nos coeficientes do modelo de Volterra, como os *kernels*, h_n é linear. O que facilita na utilização de métodos

numéricos mais comumente empregados, como mínimos quadrados, para sua estimação. Uma desvantagem desse modelo é a grande quantidade de termos para representação dos *kernels*, o que acarreta em um aumento da variância dos termos estimados (HEUBERGER *et al.*, 2005). Uma forma de reduzir essa desvantagem é a representação desse modelo usando bases de funções ortonormais (OBF) (SCHETZEN, 2006; DOYLE *et al.*, 2002)).

2.2 BASE DE FUNÇÕES ORTONORMAIS

2.2.1 Uso de Funções Ortonormais

A redução do número de termos necessário para estimação dos *kernels* com o emprego da base de funções ortonormais é resultado direto da sua própria definição:

$$\sum_{k=0}^{\infty} \Psi_i(k) \Psi_j(k) = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases}, \quad (2.5)$$

com $\Psi_i(k)$ e $\Psi_j(k)$ pertencentes a mesma base de funções ortonormais. Isso é embasado na propriedade de completude desse conjunto de funções, (OLIVEIRA *et al.*, 2011)), a qual garante que qualquer função dentro do espaço de Lebesgue $l^2[0, \infty)$ pode ser aproximada por uma combinação linear das funções que compõem a base de funções ortonormais com precisão definida pelo modelo, ou seja, para qualquer função $y(k) : \mathbb{N} \rightarrow \mathbb{R}$ em $l^2[0, \infty)$ existirão inteiros $n_1 > 0$ e $n_2 > 0$, $\forall \varepsilon > 0$ satisfazendo:

$$\sum_{k=0}^{\infty} \left(y(k) - \sum_{i=1}^{n_1} c_i \Psi_i(k) - \sum_{i_1=1}^{n_2} \sum_{i_2=1}^{n_2} c_{i_1} c_{i_2} \Psi_{i_1}(k) \Psi_{i_2}(k) \right) < \varepsilon, \quad (2.6)$$

em que ε é a precisão desejada, o conjunto de funções da base ortonormal é o utilizado para a aproximação da função $y(k)$ e os coeficientes c são escalares. Assim, esse conjunto de funções é uma alternativa viável para a representação ou identificação de um dado sistema. Essas funções podem ser entendidas como encadeamento de filtros passa-todas (NINNESS; GUSTAFSSON, 1997); HEUBERGER *et al.*, 2005):

$$\Psi_i(z) = z \frac{\sqrt{1 - |\beta_i|^2}}{z - \beta_i} \prod_{j=1}^{i-1} \left(\frac{1 - \bar{\beta}_j z}{z - \beta_j} \right), \quad i = 1, 2, \dots, \quad (2.7)$$

em que β_i são os pólos da base ortonormal ($\beta_i \in \mathbb{C} : |\beta_i| < 1$) e $\bar{\beta}_i$ é o conjugado complexo de β_i . As funções representadas acima são conhecidas como funções de Takenaka-Malmquist

(HEUBERGER *et al.*, 2005; MACHADO *et al.*, 2007)). Todo o desenvolvimento até aqui se dá no domínio da frequência, caso se deseje trabalhar no domínio do tempo, deve-se usar transformada z inversa das expressões utilizadas nessa seção (OLIVEIRA *et al.*, 2011)). Os pólos β_i são tais que $|\beta_i| < 1$, logo estáveis. As funções $\Psi_i(k)$ que compõem a base ortonormal são completas no espaço de Lebesgue se, e somente se, $\sum_{i=1}^{\infty} (1 - |\beta_i|) = \infty$, (NINNESS; GUSTAFSSON, 1997)), assim, os pólos dessas funções devem atender a esse requisito para que essas possam ser utilizadas na representação ou identificação de modelos que se desejar representar. Muitas vezes essas funções, devido à própria estrutura delas, possuem coeficientes complexos, o que é fisicamente irrealizável, mas esse problema pode ser contornado através de uma parametrização, montando uma base de funções modificadas em que todos os coeficientes são reais (NINNESS; GUSTAFSSON, 1997)). Existem algumas bases de funções ortonormais mais utilizadas que se diferenciam de acordo com a natureza dos pólos das funções que compõem a base e da quantidade de pólos distintos.

2.2.2 Funções de Laguerre

A base de Laguerre é um caso específico de base de funções ortonormais em que todos os pólos β_i são reais e iguais, ou seja, $\beta_i = \bar{\beta}_i = p$, tal que $p \in \mathbb{R}$ e, obviamente, $|p| < 1$. O parâmetro p é chamado de pólo de Laguerre e a expressão dos filtros ortonormais que constituem a base de Laguerre é descrita (FU; DUMONT, 1993); BELT; DEN BRINKER, 1995); OLIVEIRA E SILVA, 1994); TANGUY *et al.*, 1995)) assim:

$$\Psi_i(z) = z \frac{\sqrt{1-p^2}}{z-p} \left(\frac{1-pz}{z-p} \right)^{(i-1)}, \quad (2.8)$$

expressão que deixa evidente que se trata de um caso particular da expressão (2.7), em que p é o pólo único e se repete para a i -ésima função da base. As funções de Laguerre são mais utilizadas para a representação de sistemas com dinâmica amortecida de primeira ordem, pela própria natureza de sua construção, com apenas um pólo real (HEUBERGER *et al.*, 2005).

É interessante analisar o comportamento temporal dessas funções, uma vez que isso fornece insumos para analisarmos o quão oscilatória é a base de funções e o instante de convergência dessas, dois parâmetros de grande importância para a decisão de qual base adotar na representação de um sistema NLTI (TIELS; SCHOUKENS, 2014)). A figura 2.1 mostra o comportamento

temporal das seis primeiras funções de Laguerre geradas com um pólo $p = 0,8$:

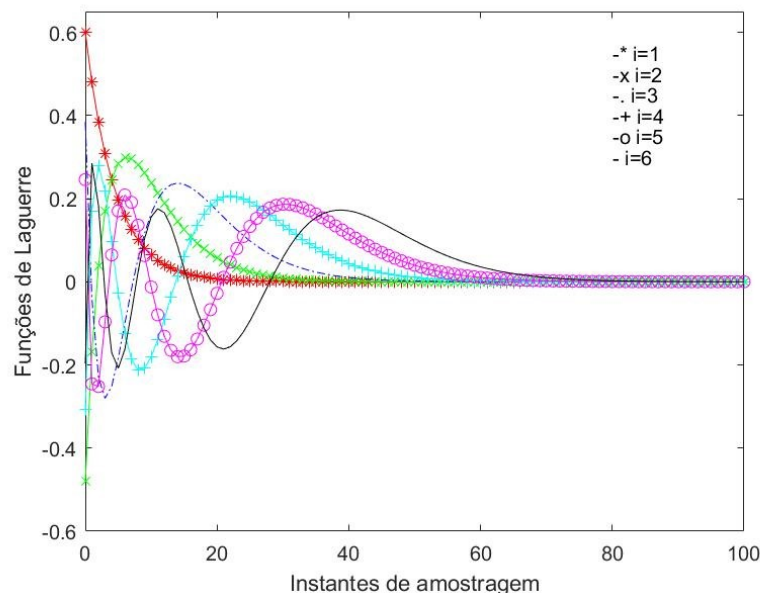


Figura 2.1. Comportamento temporal das 6 primeiras funções da base de Laguerre com pólo $p = 0,8$

2.2.3 Funções de Kautz

Um outro caso particular, de muito interesse, da base de funções ortonormais é o das funções de Kautz. Nesse, o encadeamento de filtros ortonormais da equação (2.7) é feito com funções cujos pólos são um único par de conjugados complexos, ou seja,, o par de pólos β_i será igual a β e $\bar{\beta}$. Assim, as funções de Takenaka-Malmquist ficariam da seguinte forma (NINNESS; GUSTAFSSON, 1997); HEUBERGER *et al.*, 2005):

$$\Psi_i(z) = z \frac{\sqrt{1 - |\beta|^2}}{z - \beta} \left(\frac{1 - \bar{\beta}z}{z - \beta} \right)^{i-1}. \quad (2.9)$$

Um problema da equação acima são os coeficientes complexos, devido ao par de conjugados β e $\bar{\beta}$. Como já mencionado, isso não é adequado para a representação de sistemas físicos reais e uma forma de se contornar isso é através de uma parametrização (NINNESS; GUSTAFSSON, 1997)). No caso específico de Kautz, essa parametrização é feita da seguinte forma (HEUBERGER *et*

al., 2005; TANGUY N.; CALVEZ, 2002); WAHLBERG, 1994)):

$$\Psi_{2i}(z) = z \frac{\sqrt{(1-\alpha_2^2)(1-\alpha_1^2)}}{z^2 + \alpha_1(\alpha_2 - 1)z - \alpha_2} \left[\frac{\alpha_2 z^2 + \alpha_1(\alpha_2 - 1)z + 1}{z^2 + \alpha_1(\alpha_2 - 1)z - \alpha_2} \right]^{i-1},$$

$$\Psi_{2i-1}(z) = z \frac{(z - \alpha_1)\sqrt{1-\alpha_2^2}}{z^2 + \alpha_1(\alpha_2 - 1)z - \alpha_2} \left[\frac{\alpha_2 z^2 + \alpha_1(\alpha_2 - 1)z + 1}{z^2 + \alpha_1(\alpha_2 - 1)z - \alpha_2} \right]^{i-1},$$
(2.10)

em que o par de funções Ψ_{2i} e Ψ_{2i-1} são resultado da parametrização dos pólos conjugados β e $\bar{\beta}$ em α_1 e α_2 :

$$\alpha_1 = \frac{\beta + \bar{\beta}}{1 + \beta\bar{\beta}},$$
(2.11)

$$\alpha_2 = \beta\bar{\beta},$$
(2.12)

assim, o par de funções Ψ_{2i} e Ψ_{2i-1} é construído com os parâmetros reais α_1 e α_2 , $|\alpha_1| < 1$ e $|\alpha_2| < 1$.

Uma das vantagens da base de Kautz em relação a de Laguerre é que, pela natureza dos pólos que possui, a primeira representa melhor sistemas oscilatórios (HEUBERGER *et al.*, 2005). Inclusive, pelo número de pólos das funções da base de Laguerre e de Kautz elas são mais utilizadas, respectivamente, na modelagem de sistemas com dinâmica de 1ª e de 2ª ordem. A figura 2.2 mostra o comportamento temporal das seis primeiras funções da Kautz. Esse exemplo é para o par de pólos $\beta = 0,8 \pm 0,4i$.

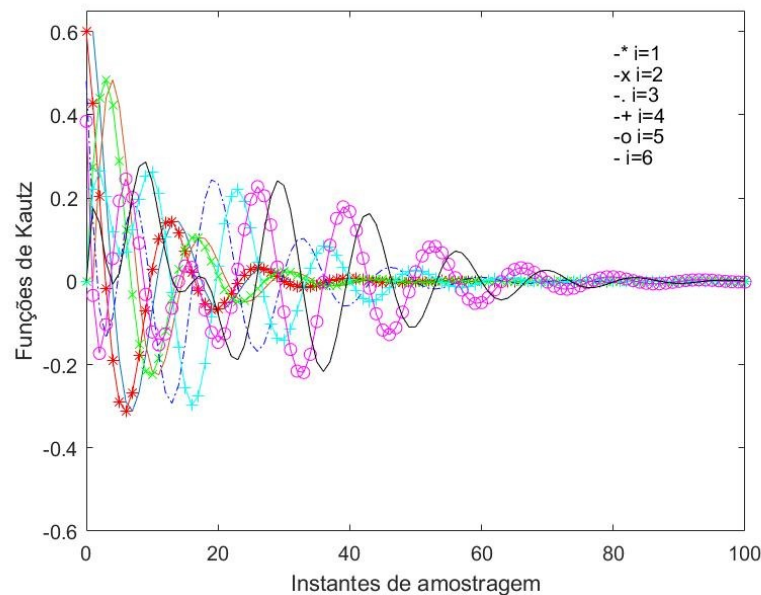


Figura 2.2. Comportamento temporal das 6 primeiras funções da base de Kautz com par de pólos $\beta = 0,8 \pm 0,4i$

Ao compararmos a figura 2.2 com a 2.1 percebemos algumas diferenças no comportamento das funções da base de Laguerre com as da base de Kautz, como o maior comportamento oscilatório das de Kautz, que as tornam mais adequadas para representação de sistemas de comportamento também mais oscilatório.

2.2.4 Base de Funções Ortonormais Generalizadas

Como já discutido nas seções anteriores, as bases de Laguerre e de Kautz são mais utilizadas na modelagem de sistemas de 1ª e de 2ª ordem, respectivamente (HEUBERGER *et al.*, 2005). Para sistemas com dinâmicas mais complexas, outras bases de funções ortonormais são preferíveis. As bases de funções ortonormais generalizadas, em inglês *Generalised Orthonormal basis of functions* (GOBF), (Van Den Hof *et al.*, 1995); HEUBERGER *et al.*, 2005; MACHADO *et al.*, 2007)) são uma alternativa para a modelagem de sistemas de ordem maior que o de segunda.

Considerando inicialmente um encadeamento de filtros ortonormais com pólos reais, representa-

se a equação (2.7) assim:

$$\Psi_i(z) = z \frac{\sqrt{1-p_i^2}}{z-p_i} \prod_{j=1}^{i-1} \left(\frac{1-p_j z}{z-p_j} \right), \quad i = 1, 2, \dots, \quad (2.13)$$

em que os pólos $p_1, p_2, \dots, p_i \in \mathbb{R}$. Dessa forma, as funções que compõem a base ortonormal que obteríamos a partir desses pólos seriam $\Psi_1, \Psi_2, \dots, \Psi_i$.

Considerando $i = n - 1$, ou seja, $n - 1$ pólos reais, teríamos $n - 1$ funções nessa base.

Para acrescentarmos um par de funções com pólos complexos nessa base ortonormal (NINNESS; GUSTAFSSON, 1997)), de forma que essas funções possuam coeficientes reais, a seguinte parametrização é proposta para as duas funções da base seguintes às $n - 1$ funções já obtidas a partir dos pólos reais:

$$\Psi'_n(z) = z \frac{\sqrt{(1-|\beta_n|^2)(\lambda' z + \gamma')}}{z^2 - (\beta_n + \bar{\beta}_n)z + |\beta_n|^2} \prod_{j=1}^{n-1} \left(\frac{1-p_j z}{z-p_j} \right), \quad (2.14)$$

$$\Psi''_n(z) = z \frac{\sqrt{(1-|\beta_n|^2)(\lambda'' z + \gamma'')}}{z^2 - (\beta_n + \bar{\beta}_n)z + |\beta_n|^2} \prod_{j=1}^{n-1} \left(\frac{1-p_j z}{z-p_j} \right),$$

em que as funções Ψ'_n e Ψ''_n são as últimas duas funções dessa nova base, gerada a partir da base anterior que continham as $n - 1$ funções $\Psi_1, \Psi_2, \dots, \Psi_i$, agora acrescidas das funções Ψ'_n e Ψ''_n parametrizadas pelo par de pólos complexos β_n e $\bar{\beta}_n$.

Os coeficientes $\lambda', \gamma', \lambda''$ e γ'' são reais e, de forma similar ao visto com a base de Kautz, resultam da parametrização do par de pólos complexos β_n e seu conjugado. Nesse caso a parametrização ocorre assim (NINNESS; GUSTAFSSON, 1997)):

$$\begin{bmatrix} \lambda' & \gamma' \end{bmatrix} \begin{bmatrix} 1 + |\beta_n|^2 & \beta_n + \bar{\beta}_n \\ \beta_n + \bar{\beta}_n & 1 + |\beta_n|^2 \end{bmatrix} \begin{bmatrix} \lambda' \\ \gamma' \end{bmatrix} = |1 - \beta_n^2|^2, \quad (2.15)$$

$$\begin{bmatrix} \lambda'' \\ \gamma'' \end{bmatrix} = \frac{1}{\sqrt{1 - \xi^2}} \begin{bmatrix} \xi & 1 \\ -1 & -\xi \end{bmatrix} \begin{bmatrix} \lambda' \\ \gamma' \end{bmatrix}, \quad (2.16)$$

em que a constante ξ é relacionada ao par de pólos β_n e $\bar{\beta}_n$ da seguinte forma:

$$\xi = \frac{\beta_n + \bar{\beta}_n}{1 + |\beta_n|^2}. \quad (2.17)$$

Das equações (2.16) e (2.17), nota-se que há uma infinidade de valores que para os parâmetros $\lambda', \gamma', \lambda''$ e γ'' . Quaisquer escolhas de valores para esses parâmetros que satisfaçam as condições das equações (2.15), (2.16) e (2.17) são válidas (NINNESS; GUSTAFSSON, 1997); ZIAEI; WANG, 2006)). Uma variação desse modelo é a inserção de mais de um par pólos conjugados.

Até então o modelo foi desenvolvido para um número $n - 1$ de pólos reais e a inserção de um par de pólos complexos conjugados . Para inserção de outros pólos complexos, sempre conjugados uma vez que trabalhamos com coeficientes reais para a representação de modelos físicos, o seguinte modelo é proposto (NINNESS; GUSTAFSSON, 1997); ZIAEI; WANG, 2006)):

$$\Psi'_{n+1}(z) = z \frac{\sqrt{(1 - |\beta_{n+1}|^2)(\rho' z + \mu')}}{z^2 - (\beta_n + \overline{\beta_{n+1}})z + |\beta_{n+1}|^2} \prod_{j=1}^{n-1} \left(\frac{1 - p_j z}{z - p_j} \right) F(z), \quad (2.18)$$

$$\Psi''_{n+1}(z) = z \frac{\sqrt{(1 - |\beta_{n+1}|^2)(\rho'' z + \mu'')}}{z^2 - (\beta_{n+1} + \overline{\beta_{n+1}})z + |\beta_{n+1}|^2} \prod_{j=1}^{n-1} \left(\frac{1 - p_j z}{z - p_j} \right) F(z),$$

sendo que

$$F(z) = \left(\frac{1 - \overline{\beta_n}}{z - \beta_n} \right) \cdot \left(\frac{1 - \beta_n}{z - \overline{\beta_n}} \right) \prod_{j=1}^{n-1} \left(\frac{1 - p_j z}{z - p_j} \right) \quad (2.19)$$

De forma análoga ao discutido na construção do modelo representado pela equação (2.14), as funções Ψ'_{n+1} e Ψ''_{n+1} correspondem às funções parametrizadas pela inserção do novo par de pólos conjugados β_{n+1} e $\overline{\beta_{n+1}}$, ou seja, é um par de funções acrescidos à base de funções geradas pelos $n - 1$ pólos reais e pelas duas funções Ψ'_n e Ψ''_n .

Também de forma similar ao que já foi proposto, os parâmetros ρ', μ', ρ'' e μ'' possuem a mesma relação com os pólos β_{n+1} e seu conjugado que os parâmetros $\lambda', \gamma', \lambda''$ e γ'' possuem com os pólos β_n e seu conjugado expressas nas equações (2.15), (2.16) e (2.17).

Um exemplo dessa construção é o caso da modelagem de um sistema com três pólos, um pólo real p e um par de conjugados β e $\overline{\beta}$. A figura 2.3 mostra o comportamento no tempo das seis primeiras funções do caso particular, em que os pólos são $p = 0, 8$ e $\beta = 0, 8 \pm 0, 4i$:

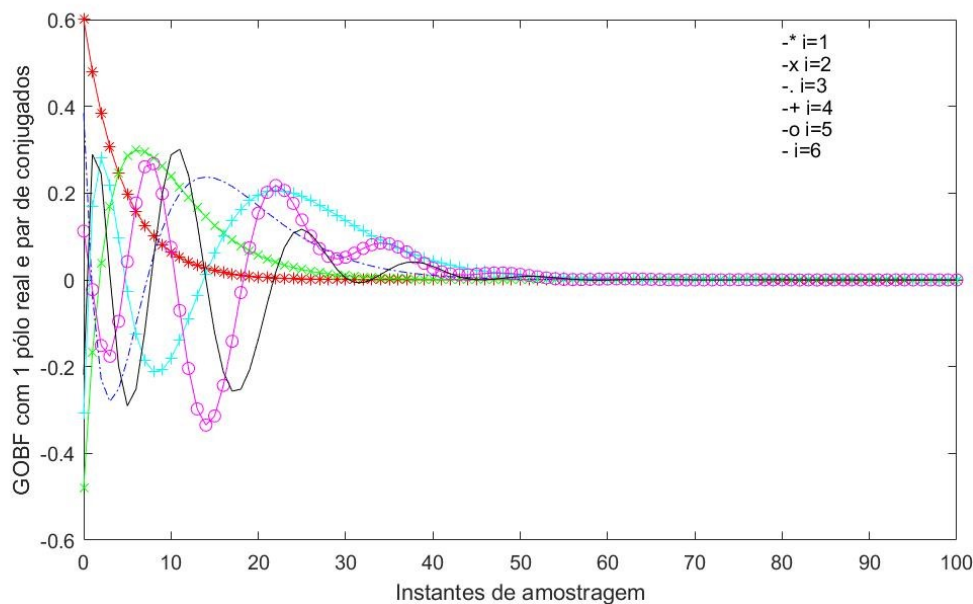


Figura 2.3. Comportamento temporal das 6 primeiras funções da GOBF com 1 pólo real $p = 0,8$ e 1 par de conjugados $\beta = 0,8 \pm 0,4i$

Como discutido nas seções de Laguerre (2.2.2) e de Kautz 2.2.3, a análise do comportamento da função ilustrado na figura acima é de grande utilidade para a análise da convergência das funções e do comportamento oscilatório delas (TIELS; SCHOUKENS, 2014)).

Ao compararmos o conjunto de equações (2.14), (2.15) e (2.16) com o conjunto das equações (2.10), (2.11) e (2.12) as semelhanças são várias, especialmente os coeficientes da variável complexa z . Já as duas últimas funções da GOBF de 3ª ordem são, de certa forma, uma combinação da construção da base de funções de Laguerre com a construção da base de funções de Kautz.

2.3 MODELO OBF-VOLTERRA

Como já discutido na seção 2.1, algumas das grandes vantagens da representação e modelagem de sistemas não-lineares através do modelo de Volterra é que este é linear nos parâmetros e a capacidade de representar sistemas com memória. .

Uma desvantagem dessa abordagem é o número alto de variáveis desconhecidas que devem ser estimadas via métodos numéricos, o que leva, muitas vezes, a uma alta variância desses parâmetros (OLIVEIRA *et al.*, 2012)). Uma forma de contornar isso é através da expansão dos *kernels* de Volterra usando uma base de funções ortonormais, reduzindo o número de parâmetros a serem estimados e assim, diminuindo a variância dessa estimativa (SCHETZEN, 2006; DOYLE *et al.*, 2002)).

Assume-se que os kernels h_n sejam truncados até um limite ϵ_n , ou seja, $h_n(k_1, \dots, k_n) = 0$ para $k_j > \epsilon_n (\forall j \in \{1, \dots, n\})$, assim eles podem ser representados no espaço de Lebesgue, permitindo que possam ser expandidos usando uma base de funções ortonormais 2.2. Assumindo que tanto o *kernel* de 1ª ordem quanto o de 2ª ordem sejam representados por funções ortonormais Ψ que compõem uma mesma base ortonormal tem-se que (SCHETZEN, 2006; RUGH, 1981)):

$$\begin{aligned} \hat{h}_1(k) &= \sum_{i=1}^{n_1} c_i \Psi_i(k), \\ \hat{h}_2(k_1, k_2) &= \sum_{i_1=1}^{n_2} \sum_{i_2=1}^{i_1} c_{i_1, i_2} \Psi_{i_1}(k_1) \Psi_{i_2}(k_2), \end{aligned} \tag{2.20}$$

em que Ψ_i é a i -ésima função da base ortonormal, n_1 e n_2 são o número de funções ortonormais Ψ que serão utilizadas na representação dos *kernels* de primeira e de segunda ordem e os escalares c_{i_1, i_2} são resultado da expansão dos *kernels* utilizando a base de funções ortonormais. Esses coeficientes, para o modelo de Volterra de 2ª ordem, com n_1 e n_2 funções ortonormais Ψ na expansão dos *kernels* são definidos como (SCHETZEN, 2006; RUGH, 1981)):

$$\begin{aligned} c_i &= \sum_{k=0}^{\epsilon_n} h_1(k) \Psi_i(k), \\ c_{i_1, i_2} &= \sum_{k_1=0}^{\epsilon_n} \sum_{k_2=0}^{\epsilon_n} h_2(k_1, k_2) \Psi_{i_1}(k_1) \Psi_{i_2}(k_2), \end{aligned} \tag{2.21}$$

em que $h_1(k)$ e $h_2(k_1, k_2)$ são os *kernels* de primeira e de segunda ordem, respectivamente, e ϵ_n é o limite de truncamento, dado pelo número de amostras da função $y(k)$ que se deseja representar, de tal forma que os *kernels* $h_1(k)$ e $h_2(k_1, k_2) = 0, k > \epsilon_n$. Quando se deseja representar um eventual *offset* do sistema através de um *kernel* de ordem 0 h_0 como em (BILLINGS, 1980); DUMONT; FU, 1993)), também se faz necessário estimar o coeficiente c_0 .

Assim, combinando as expressões (2.3),(2.20) e (2.21) , obtém-se a representação $\hat{y}(k)$ da função $y(k)$ segundo o modelo de Volterra de segunda ordem (SCHETZEN, 2006; RUGH, 1981)):

$$\hat{y}(k) = \sum_{i_1=1}^{n_1} c_{i_1} l_{i_1}(k) + \sum_{i_1=1}^{n_2} \sum_{i_2=1}^{i_1} c_{i_1, i_2} l_{i_1}(k) l_{i_2}(k), \quad (2.22)$$

sendo que $l_i(k) = \sum_{\tau=0}^k \Psi_{i_j}(\tau)x(k - \tau)$, ou seja, l_i é o resultado da entrada $x(k)$ filtrada pelas funções ortonormais Ψ .

Assim como os *kernels* de segunda ordem descritos na representação original do modelo de Volterra na seção 2.1, os coeficientes de segunda ordem c_{i_1, i_2} a serem estimados segundo equação (2.21) são também simétricos, uma vez que se trate de um sistema SISO, de tal forma que $c_{i,j} = c_{j,i}$.

SELEÇÃO DOS PÓLOS DAS BASES DE FUNÇÕES ORTONORMAIS

Uma das grandes vantagens do modelo de Volterra é a capacidade de representação de sistemas não-lineares em parâmetros lineares, como introduzido na seção (2.1) , facilitando o uso de algoritmos mais usuais. A metodologia aqui adotada explora isso, descrevendo o algoritmo montado para a representação de sistemas NLIT através do modelo OBF-Volterra utilizando três bases de funções ortonormais distintas, Laguerre, Kautz e a GOBF proposta com um pólo real e um par de conjugados. Todas as etapas do processo foram implementadas na plataforma do MATLAB[®]. Os códigos relativos a cada uma das etapas está anexado e foi programado de forma similar às três bases. Assim, será anexado relativo a uma delas após cada seção o código utilizado para a GOBF proposta.

Conforme será apresentado no decorrer desse capítulo, e também nos capítulos 4 e 5, o custo computacional foi uma variável importante e assim, informações do computador utilizado no trabalho são importantes. O trabalho foi realizado em um computador modelo Dell, Inspiron 14 com as seguintes características:

- Processador: Intel[®] Core[™]i5-8250U CPU @ 1.60GHz 1.80 GHz; ;
- RAM: 8,00 GB (utilizável: 7,66 GB) ;
- Sistema operacional: 64 bits, processador baseado em x64, Windows 10 Pro;
- HD: WDC WD10SOZX-75Z10T1, espaço de 927 GB.

3.1 GERAÇÃO DE FUNÇÕES ORTONORMAIS

O primeiro passo é a própria geração das funções ortonormais. Nesse trabalho, para a seleção do melhor conjunto de pólos que permitem uma identificação mais precisa do sistema dinâmico

foi projetado um algoritmo de busca exaustiva, de tal forma que se fez necessário a geração das seis primeiras funções de Laguerre, de Kautz e da GOBF proposta para todo um domínio de pólos, em que o número de elementos desse domínio foi limitado por razões computacionais. Assim para alcançarmos a precisão desejada de quatro casas decimais, foi necessário repetir o processo refinando os limites do conjunto de pólos a cada repetição. O algoritmo adotado para essa etapa pode ser visto na seção 3.1 do apêndice A.

A quantidade máxima de amostras k da equação (2.4) foi limitada segundo dois critérios. Primeiro, o instante de convergência das funções $\Psi(k)$, a partir do qual o valor delas permanece nulo. Para obter tal quantidade foi desenvolvido o algoritmo descrito na seção A.2 do anexo A. Porém, há algumas funções que demoram muito para convergir, e para essas, foi limitado ao número de amostras do próprio sistema que se deseja identificar, uma vez que além desse, os valores não são utilizados.

3.2 ESTIMAÇÃO DOS COEFICIENTES VIA MÍNIMOS QUADRADOS E CÁLCULO DOS *KERNELS*

Um dos pontos mais discutidos nesse trabalho foi a capacidade de se representar sistemas dinâmicos não-lineares através de parâmetros lineares usando o modelo de Volterra.

A equação (2.22) evidencia essa relação e também sintetiza o que foi implementado nessa etapa. Tendo já o conjunto de dados de entrada e de saída (I/O) do sistema NLIT estudado, $y(k)$ e $u(k)$, e já tendo sido geradas as bases de funções ortonormais para todo o conjunto de pólos propostos, o algoritmo implementado nessa etapa foi o de obtenção das funções $l_i(k)$, ou seja, o processo de filtragem das funções de entrada $u(k)$ pelas funções que compõem cada uma das bases. Esse processo foi feito através de *loops* para englobar as funções resultantes de todos os pólos utilizado. No caso deste trabalho, por se realizar uma comparação entre as três bases, buscou-se usar o mesmo número de funções para as três, assim, n_1 e n_2 da equação (2.22) são iguais a 6. O processo está descrito no código A.3.

Obtidas as funções $l_i(k)$, e com os dados de saída do sistema em questão, $y(k)$, estima-se os coeficientes c_i e c_{i_1, i_2} via mínimos quadrados. Nesse trabalho, utilizou-se a função do MATLAB®, *lsqr* (PAIGE; SAUNDERS, 1982)). Em suma, essa função resolve o sistema linear

sobredeterminado $A \cdot x = b$ que minimiza o módulo de $(b - A \cdot x)$.

No caso, a matriz de coeficientes A , o vetor x a se estimar e a função a qual se deseja representar b são $\lambda(k)$, ζ e $y(k)$, respectivamente:

$$\lambda(k) = \begin{bmatrix} l_1(0) & \cdots & l_6(0) & l_1(0)^2 & l_2(0)l_1(0) & l_2(0)^2 & \cdots & l_6(0)l_1(0) & l_6(0)l_2(0) & \cdots & l_6(0)^2 \\ l_1(1) & \cdots & l_6(1) & l_1(1)^2 & l_2(1)l_1(1) & l_2(1)^2 & \cdots & l_6(1)l_1(1) & l_6(1)l_2(1) & \cdots & l_6(1)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_1(k) & \cdots & l_6(k) & l_1(k)^2 & l_2(k)l_1(k) & l_2(k)^2 & \cdots & l_6(k)l_1(k) & l_6(k)l_2(k) & \cdots & l_6(k)^2 \end{bmatrix} \quad (3.1)$$

$$\zeta = [c_1 \quad \cdots \quad c_6 \quad c_{1,1} \quad c_{2,1} \quad c_{2,2} \quad \cdots \quad c_{6,1} \quad c_{6,2} \quad \cdots \quad c_{6,6}]^T \quad (3.2)$$

$$y(k) = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(k) \end{bmatrix} \quad (3.3)$$

A implementação foi feita conforme algoritmo A.4. No estudo de caso que será discutido no capítulo seguinte, fez-se a retirada do *offset* sistema em estudo para que não houvesse a necessidade da estimação do parâmetro c_0 , que seria usado para a representação do *kernel* h_0 , de forma a reduzir o custo computacional, de forma a reduzir o tamanho dos arquivos intermediários gerados no algoritmo na plataforma do MATLAB e também o tempo de simulação, permitindo uma varredura de um maior número de pólos a cada iteração.

Com os coeficientes c_i e c_{i_1, i_2} estimados, a obtenção dos *kernels* \hat{h}_1 e \hat{h}_2 é feita de acordo

com a expressão (2.20) e código A.5

Importante ressaltar que todo esse processo foi adaptado para que fosse realizado para cobrir todo um domínio de pólos e as expressões descritas nessa seção foram devidamente adaptadas para isso.

3.3 SELEÇÃO DO MELHOR CONJUNTO DE PÓLOS E VALIDAÇÃO

Fazendo uma breve recapitulação, dado um conjunto de pólos, foram calculadas as bases de Laguerre, Kautz e GOBF para esse conjunto, em que foram representadas as seis primeiras funções para cada base de cada pólo. Feito isso, estimou-se os coeficientes por expansão via mínimos quadrados, permitindo o cálculo dos *kernels* de primeira e de segunda ordem para toda essa variedade de pólos. Agora descreveremos a filtragem de todos esses dados para obtermos os pólos que permitem uma melhor identificação da saída do sistema usando cada uma das três bases de funções propostas.

Para tal, foi calculado a função (2.22) para todo o conjunto acima descrito e calculou-se o erro quadrático médio (EQM) da seguinte forma:

$$EQM = \frac{1}{N_d} \sum_{k=1}^{N_d} (y(k) - \hat{y}(k))^2, \quad (3.4)$$

em que $y(k)$ é a saída original, no caso, a primeira metade do conjunto de dados do estudo de caso e $\hat{y}(k)$ é a função obtida. Após esses cálculos, comparou-se os erros quadráticos médios gerados pelas diferentes elementos do conjunto de pólos e foi escolhida a combinação que resultava no menor EQM. Importante mencionar que o limite de elementos do conjunto de pólos mencionados no início da seção 3.1 foi estabelecido devido a essa etapa, uma vez que se chegou a um limite de tamanho de arquivos permitidos no software. Todo esse processo foi repetido quatro vezes para as funções de Laguerre, de Kautz e da GOBF em questão com o intuito de chegar à precisão desejada de quatro casas decimais, padrão desse trabalho. Também calculou-se o erro quadrático normalizado (EQN), utilizado nos trabalhos (OLIVEIRA *et al.*, 2011) e (OLIVEIRA *et al.*, 2012)) definido como:

$$EQN \triangleq 10 \log \frac{\sum_{k=1}^{N_d} [y(k) - \hat{y}(k)]^2}{\sum_{k=1}^{N_d} y(k)^2}. \quad (3.5)$$

Assim foi obtida a combinação de pólos que levam aos *kernels*, tanto de primeira ordem quanto de segunda ordem, que permitem uma melhor identificação, dentro do algoritmo descrito, do sistema em estudo via modelo de Volterra de segunda ordem. Para essa etapa se implementou o último código descrito nesse trabalho A.6.

Por fim, todo o processo foi validado, utilizando os *kernels* de primeira e de segunda ordem obtidos em um outro conjunto de dados. No caso desse trabalho, fez-se um estudo de caso em que se utilizou a primeira metade do período do sistema em estudo para a etapa de estimação e a segunda metade para a validação, usando o mesmo código A.6.

MODELAGEM DE UMA REAÇÃO DE POLIMERIZAÇÃO EM UM CSTR

Inicialmente será feita uma descrição do sistema NLIT estudado, uma reação de polimerização em um reator ideal de tanque agitado, em inglês *Continuous Stirred Tank Reactor* (CSTR). A escolha deste estudo de caso se deve a esse ser um sistema cuja dinâmica é não-linear, de ordem superior à segunda e é um caso bem discutido em importantes trabalhos (DOYLE *et al.*, 1995); DOYLE *et al.*, 2002); OLIVEIRA *et al.*, 2011); OLIVEIRA *et al.*, 2012)). São enumeradas as premissas simplificadoras, descreve-se a representação no espaço de estados dessa reação e são apresentados os gráficos da saída em função do tempo.

Na seção seguinte, são apresentados e discutidos os resultados obtidos para a identificação da primeira metade da reação no CSTR, utilizando o modelo OBF-Volterra de segunda ordem com cada base de funções discutida, Laguerre, Kautz e a GOBF proposta. Também serão analisados a queda na precisão entre o processo de seleção e o de validação, ou seja, na representação pelo modelo de Volterra da primeira metade do conjunto de dados de saída do CSTR com da segunda metade, de forma a se discutir o viés do trabalho. Ao final desse capítulo, comparam-se os resultados entre cada uma das bases e se discute as vantagens e desvantagens comparativas entre elas, além de se as expectativas do trabalho foram atendidas e os eventuais motivos de não terem sido.

4.1 REAÇÃO DE POLIMERIZAÇÃO EM UM CSTR

Para a discussão da representação do modelo de Volterra para sistemas NLIT, fazemos o estudo de caso de uma reação de polimerização de radical livre de metacrilato de metila (MMA) com Azobisisobutironitrila (AIBN) como iniciador e tolueno como solvente em um CSTR. A representação dessa reação foi a mesma adotada em (DOYLE *et al.*, 1995)), na qual foram

adotadas as seguintes premissas afim de simplificação:

- Operação isotérmica;
- Mistura perfeita;
- Capacidade de calor constante;
- Sem polímeros no fluxo de entrada;
- Sem reação de autoaceleração (efeito gel);
- Volume constante do reator;
- Hipóteses de estado quase estacionário e de cadeia longa adotadas.

Com essas premissas, foi obtida a representação no modelo de espaço de estados tendo como saída o peso molecular médio em número (M_n), $y(t)$ (kg/mol), e como entrada o quociente de vazão do iniciador, $u(t)$ [m^3/h] (DOYLE *et al.*, 1995) :

$$\begin{cases} \dot{x}_1(t) = 10(6 - x_1(t)) - 2.4568x_1(t)\sqrt{x_2(t)}, \\ \dot{x}_2(t) = 80u - 10.1022x_2(t), \\ \dot{x}_3(t) = 0.0024121x_1(t)\sqrt{x_2(t)} + 0.112191x_2(t) - 10x_3(t), \\ \dot{x}_4(t) = 245.978x_1(t)\sqrt{x_2(t)} - 10x_4(t), \\ y = \frac{x_4(t)}{x_3(t)}, \end{cases} \quad (4.1)$$

em que as variáveis de estado são: x_1 , concentração do monômero, x_2 , concentração do iniciador (AIBN), x_3 e x_4 são, respectivamente, o peso total de todas as moléculas poliméricas da amostra e o número total de moléculas da amostra, utilizados para a obtenção da saída $y(t)$. As condições iniciais de estudo adotadas foram $x_1(0) = 5,50677$, $x_2(0) = 0,132906$, $x_3(0) = 0,0019752$, $x_4(0) = 49,3818$, $u(0) = 0,016783$ e $y(0) = 25000,5$. Importante reforçar que todo o processo de obtenção desse modelo está descrito com detalhes em (DOYLE *et al.*, 1995); MANER *et al.*, 1996).

Também nessa mesma referência, foi feita uma normalização do modelo de estados da equação (4.1), para fins numéricos durante os processos de simulação, em que as variáveis de entrada, de saída e de estado normalizadas foram definidas como $z_i = \frac{x_i - x_i(0)}{x_i(0)}$, de tal forma que as condições iniciais para todas elas são nulas. Feito isso, é possível, também para fins numéricos

durante o processo de simulação, fazer uma aproximação por séries de Taylor truncadas na segunda ordem e obter o seguinte modelo de espaço de estados (DOYLE *et al.*, 1995):

$$\begin{cases} \dot{z}_1(t) = -10.8957z_1(t) - 0.447837z_2(t) - 0.447837z_1(t)z_2(t) + 0.111959z_2^2(t), \\ \dot{z}_2(t) = -10.1022z_2(t) + 10.1022\tilde{u}(t), \\ \dot{z}_3(t) = 2.4162z_1(t) + 8.7744z_2(t) - 10z_3(t) + 1.22581z_1(t)z_2(t) - 0.306453z_2^2(t), \\ \dot{z}_4(t) = 10z_1(t) + 5.00001z_2(t) - 10z_4(t) + 5.00001z_1(t)z_2(t) - 1.25z_2^2(t), \\ \tilde{y}(t) = -z_3(t) + z_4(t) - z_3(t)z_4(t) + z_3^2(t). \end{cases} \quad (4.2)$$

Para esse sistema, foi considerada que a entrada $u(t)$ fosse uma sequências de funções degrau de amplitude uniformemente distribuídos no intervalo de $[-0, 6, 0, 8]$ ao longo de trinta e duas horas, com período de amostragem de 0,03 horas. O comportamento temporal simulado é representado no gráfico a seguir.

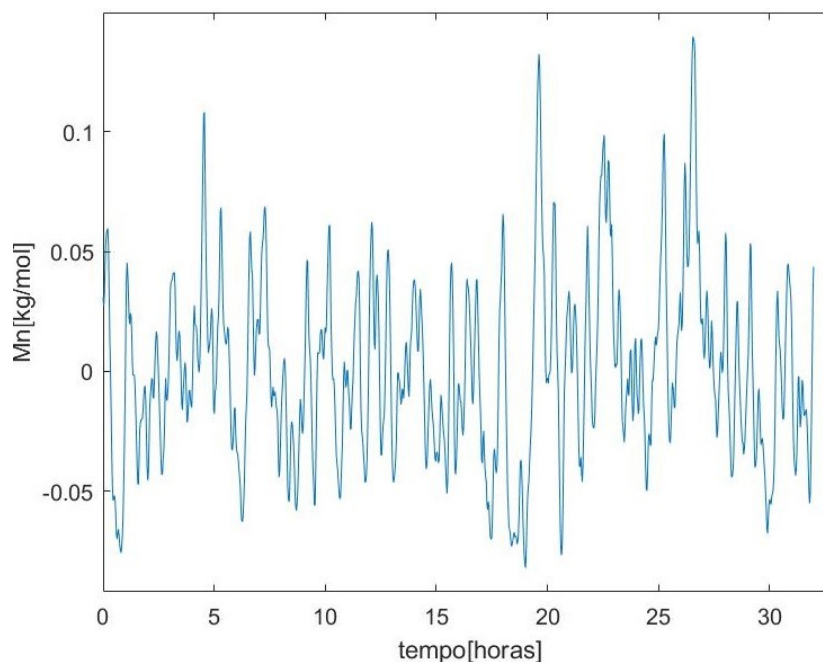


Figura 4.1. Função de saída do CSTR para sequência de funções degrau de pequena amplitude

A primeira metade do período (16 horas) será usada para a seleção do conjunto de pólos que permitem a melhor identificação do sistema através do modelo de Volterra para cada uma das três bases de funções ortonormais propostas e a segunda metade será utilizada para a validação do modelo proposto. Assim, na primeira etapa utilizou-se as primeiras 533 amostras, enquanto

que na segunda, as demais 534. Na representação acima já foi retirado o *offset* da função para fins numéricos.

4.2 APRESENTAÇÃO E DISCUSSÃO DE RESULTADOS: LAGUERRE

Como já descrito no capítulo anterior, foram realizadas quatro iterações para atingir a precisão desejada. Assim, a combinação de pólos que melhor identifica o sistema em questão para a base de funções de Laguerre foi o pólo $p = 0,9113$ para o *kernel* de primeira ordem e também para o *kernel* de segunda ordem.

O comportamento do *kernel* de primeira ordem h_1 é apresentado a seguir:

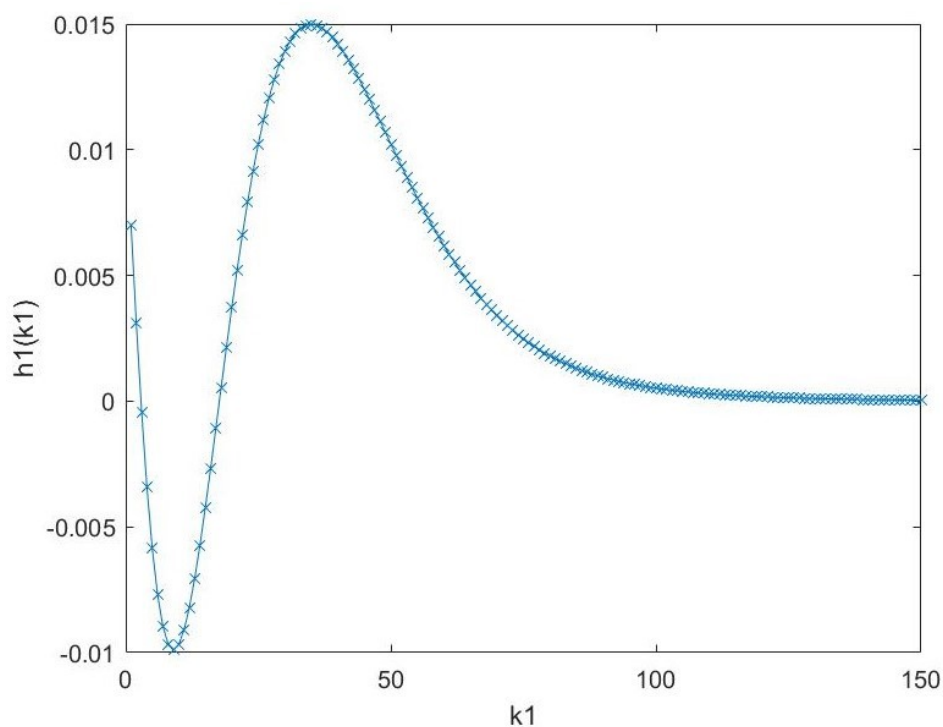


Figura 4.2. Kernel de 1ª ordem obtido para base Laguerre com pólo $p = 0,9113$

E o comportamento do *kernel* de segunda ordem h_2 :

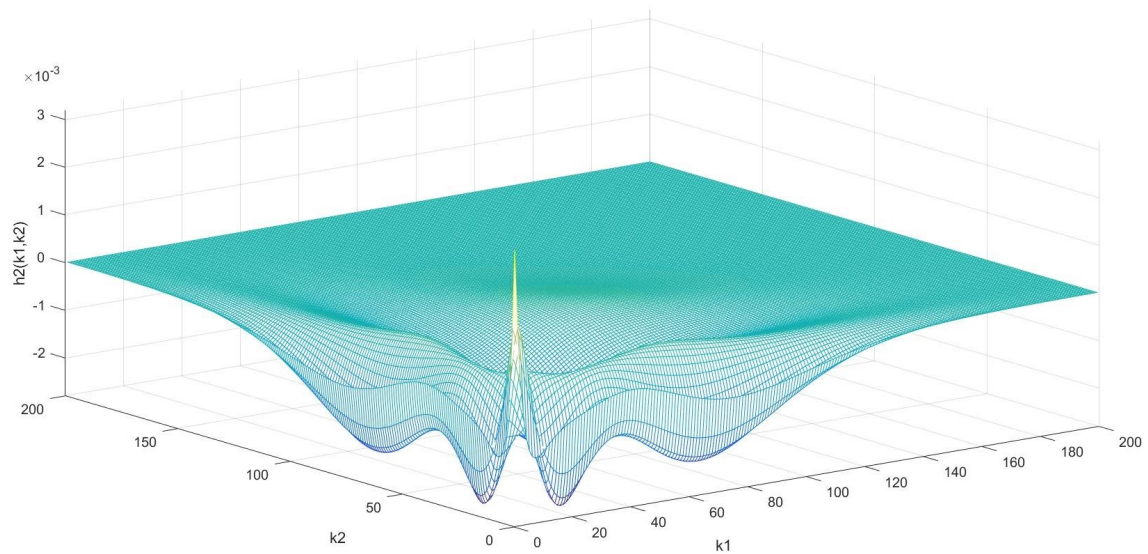


Figura 4.3. Kernel de 2^a ordem obtido para base Laguerre com pólo $p = 0,9113$

Alguns pontos interessantes desses resultados foram os módulos desses pólos, que foram próximos a um . Por um lado, isso evidencia o que foi discutido no capítulo 2, que as funções de Laguerre seriam mais adequadas para a representação de sistemas de primeira ordem, diferente do em estudo, que é quarta ordem. Por outro lado, também indica uma deficiência do algoritmo e da metodologia adotada, uma vez que o número de pólos analisados é menor do que poderia ser e o próprio processo de refinação para alcançar a precisão desejada, em que restringe os limites do domínio em prol de uma maior precisão enviesa a seleção para que os pólos dos *kernel* de primeira e de segunda ordem estejam próximos. Essa limitação é mais evidente ainda quando se analisa os resultados obtidos para Kautz e para a GOBF proposta, que serão apresentados mais adiante. Outro ponto a se evidenciar é a representação dos *kernels* de 2^a ordem no gráfico 4.3 em que fica claro a simetria, conforme posto no capítulo 2 .

Todavia tanto a identificação, quanto a validação apresentaram uma boa precisão, como os gráficos a seguir mostram:

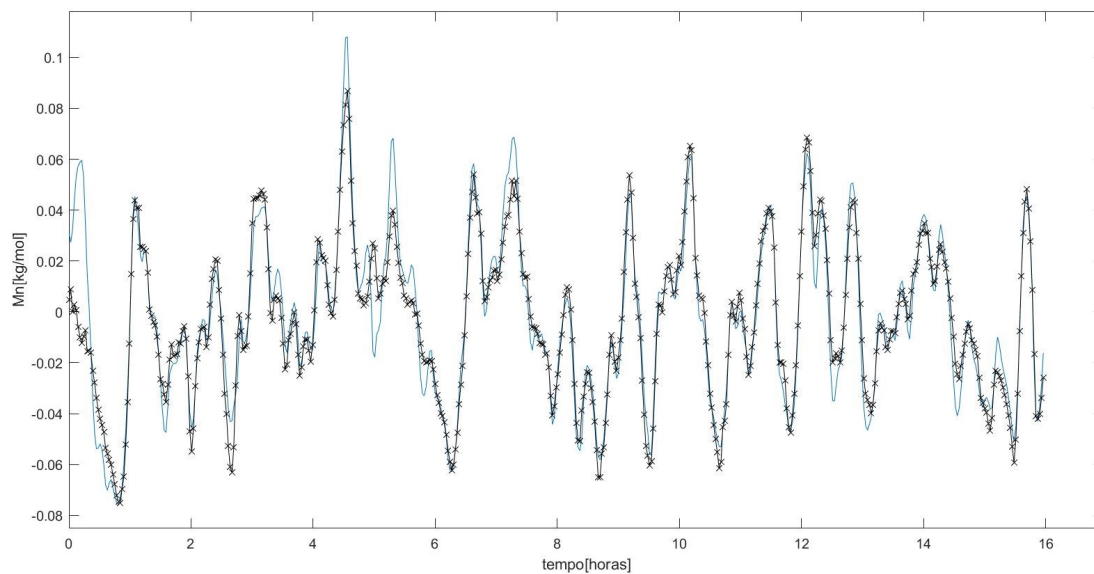


Figura 4.4. Função de saída do CSTR durante as primeiras 16 horas (linha contínua) e saída obtida a partir do modelo de Volterra com as funções de Laguerre (linha com símbolo)

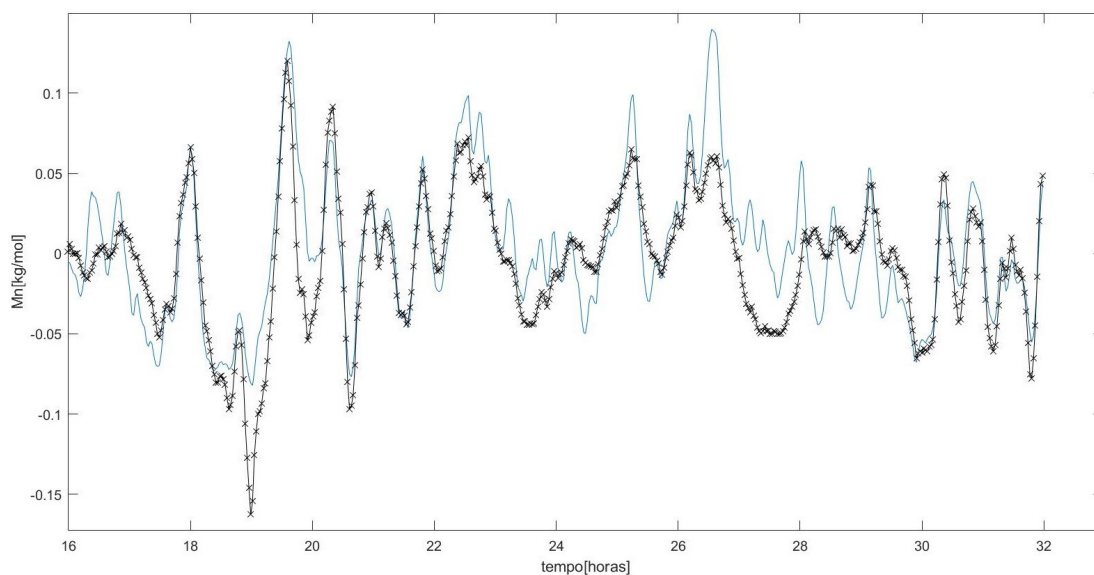


Figura 4.5. Função de saída do CSTR durante as 16 horas restantes (linha contínua) e saída obtida a partir do modelo de Volterra com as funções de Laguerre (linha com símbolo)

Comparando-se o gráfico 4.5 com o 4.4 percebe-se um prejuízo na precisão, mas que era

esperado dado o viés estatístico.

Uma observação na representação gráfica foi evidenciar que ao realizarmos a identificação via modelo de Volterra e, também, na representação dos *kernels*, trabalha-se no domínio discreto do tempo em que as funções só podem ser avaliadas nos instantes de amostragem. Para representar esses instantes, o símbolo "x" foi utilizado.

Os resultados ilustrados nos gráficos acima são agora expostos numericamente, apresentando os erros EQM e EQN tanto para o processo de recuperação dos estados de Laguerre (seleção) quanto para a validação do modelo:

	EQM	EQN (dB)
Estimação	$1,1254 \times 10^{-4}$	-9,7301
Validação	$7,0830 \times 10^{-4}$	-4,2506

Tabela 4.1. Performance da simulação do modelo de OBF-Volterra com funções de Laguerre

Por fim, para reforçar os resultados da tabela acima, também foi gerado o gráfico do erro da representação durante todo o período da reação:

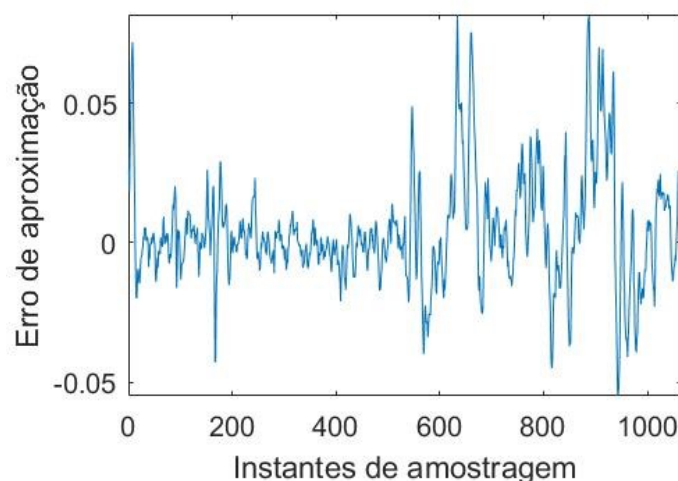


Figura 4.6. Erro entre saída do sistema $y(k)$ e saída do modelo $\hat{y}(k)$ utilizando funções de Laguerre

O gráfico mostrado na figura 4.6 deixa bem evidente a perda de precisão entre as etapas de

recuperação dos estados de Laguerre e da validação do modelo, o que era esperado dado o viés estatístico da abordagem adotada.

4.3 APRESENTAÇÃO E DISCUSSÃO DE RESULTADOS: KAUTZ

Nas simulações para Kautz, foi o primeiro momento em que foi identificado um prejuízo maior pela limitação do algoritmo proposto. Dados os limites de elementos do conjunto de pólos em que são feitas a busca por exaustão, observou-se que essa limitação levou a um maior erro no caso de Kautz em comparação com Laguerre, o que foge do esperado, uma vez que a base de Kautz contempla as funções da base de Laguerre.

Por mais que tenha sido identificada essa falha, manteve-se o padrão de procedimento adotado para Laguerre e para a GOBF proposta, de tal forma que soluções e melhorias para esse problema são discutidas no capítulo 5.

Dessa forma, utilizando a base de funções de Kautz, a melhor combinação de pólos foi de $\beta = 0,9000$ para o *kernel* de 1ª ordem h_1 e o par conjugados $\beta = -0,7000 \pm 0,7000i$ para o de 2ª ordem h_2 . O comportamento do *kernel* de primeira ordem obtido utilizando a base de funções de Kautz é apresentado a seguir:

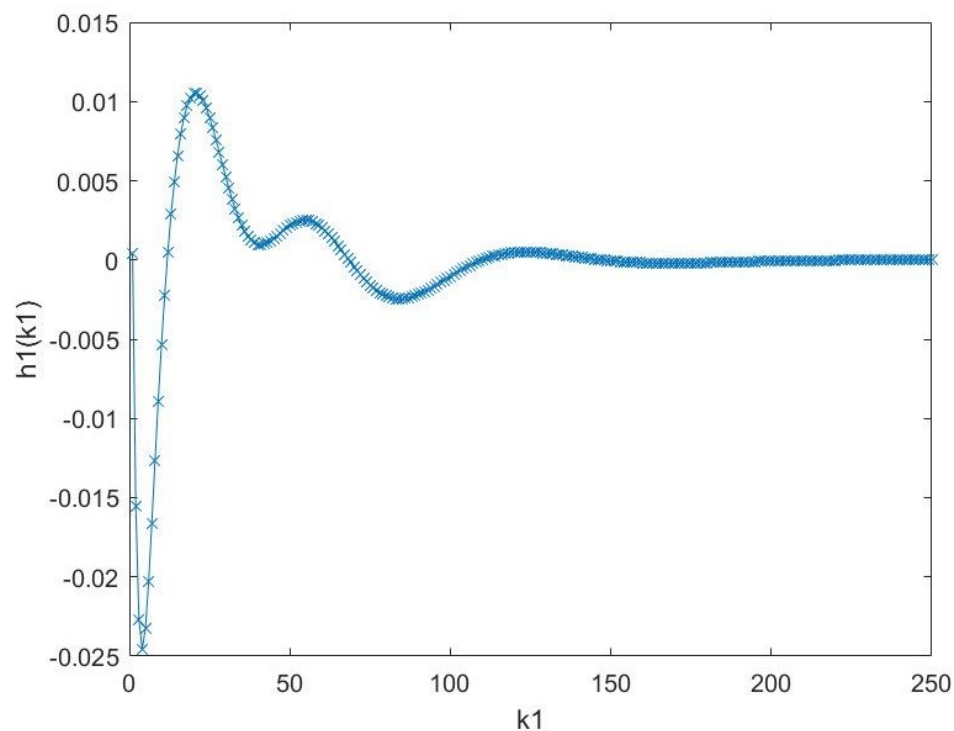


Figura 4.7. *Kernel* de 1ª ordem obtido para base Kautz com pólo $\beta = 0,9000$

Alguns pontos a se destacar são o comportamento mais oscilatório dos *kernel* de 1ª ordem quando comparado ao obtido para Laguerre, mesmo com pólos de magnitude semelhantes, o que se deve a um comportamento mais oscilatório das funções de Kautz e, por consequência, do *kernel*. Outra consequência desse comportamento, é o maior número de amostras necessário para que o *kernel* convirja.

Para o de 2ª ordem tem-se:

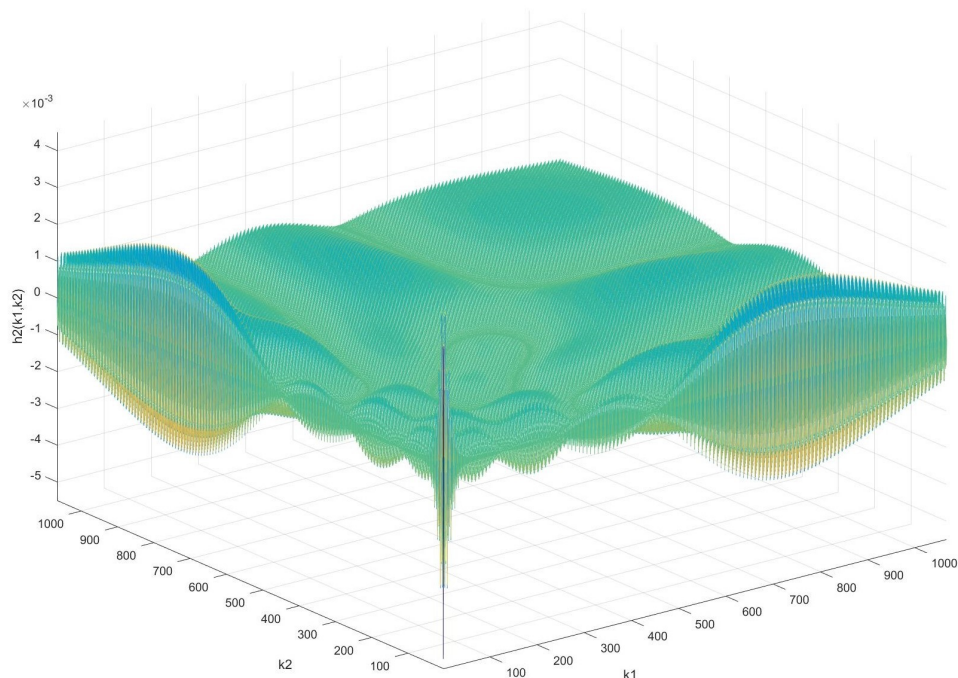


Figura 4.8. Kernel de 2ª ordem obtido para base Kautz com par de conjugados $\beta = -0,7000 \pm 0,7000i$

Nesse caso, percebe-se pelo gráfico 4.8 uma alternância mais frequente entre valores positivos e negativos no *kernel* de 2ª ordem, muito por decorrência do sinal negativo da parte real do pólo obtido. E, como já discutido no início dessa seção 4.3, houve uma queda na precisão da identificação no caso de Kautz quando comparado à Laguerre, como o gráfico a seguir ilustra:

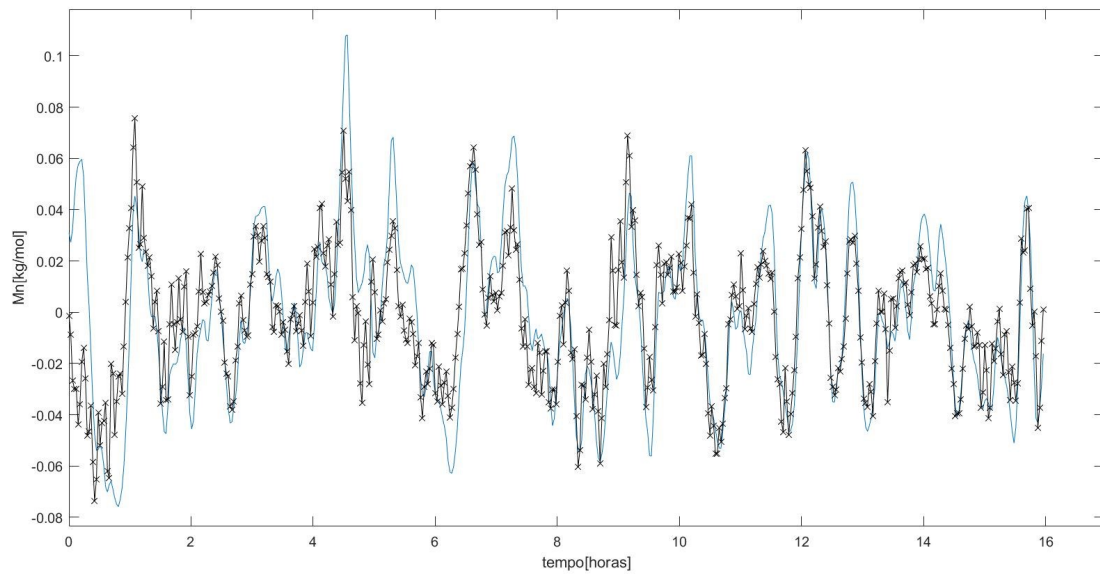


Figura 4.9. Função de saída do CSTR durante as primeiras 16 horas (linha contínua) e saída obtida a partir do modelo de Volterra com as funções de Kautz (linha com símbolo)

E, por consequência, também observamos essa perda de acurácia na etapa de validação:

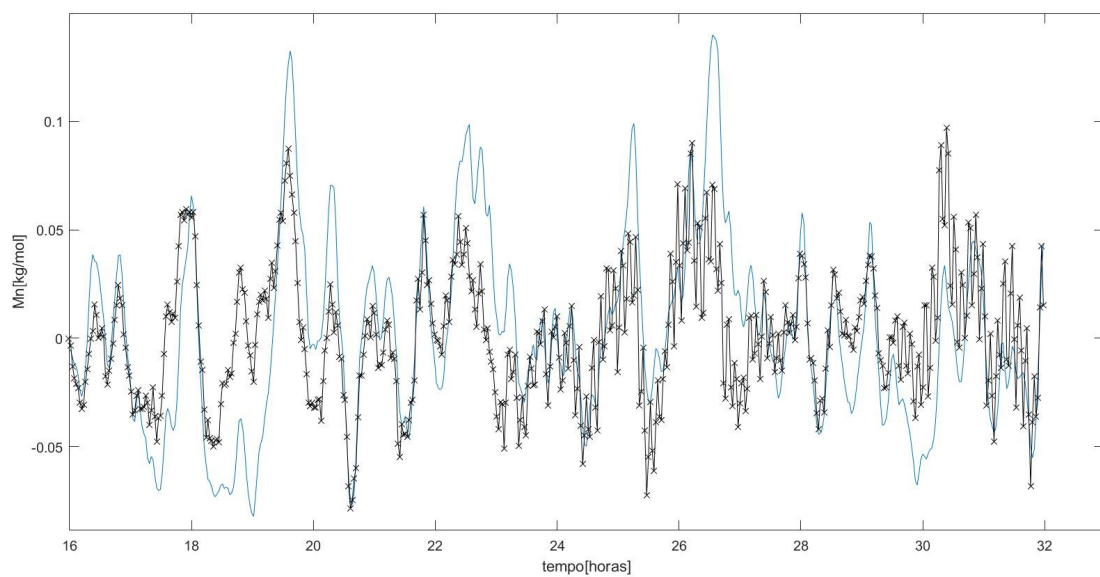


Figura 4.10. Função de saída do CSTR durante as 16 horas restantes (linha contínua) e saída obtida a partir do modelo de Volterra com as funções de Kautz (linha com símbolo)

Contudo, os gráficos e os valores numéricos apresentados a seguir ainda demonstram uma boa capacidade de identificação do modelo e, mais que isso, indicam para um bom potencial de melhoria uma vez resolvidos os problemas da metodologia proposta.

Assim, utilizando o modelo de Volterra-OBF com base de funções de Kautz obtivemos os seguintes resultados para erros, representados abaixo tanto numericamente quanto na forma gráfica, assim como feito para o caso de Laguerre:

	EQM	EQN (dB)
Estimação	$4,1370 \times 10^{-4}$	-4,0764
Validação	0,0011	-2,3582

Tabela 4.2. Performance da simulação do modelo de OBF-Volterra com funções de Kautz

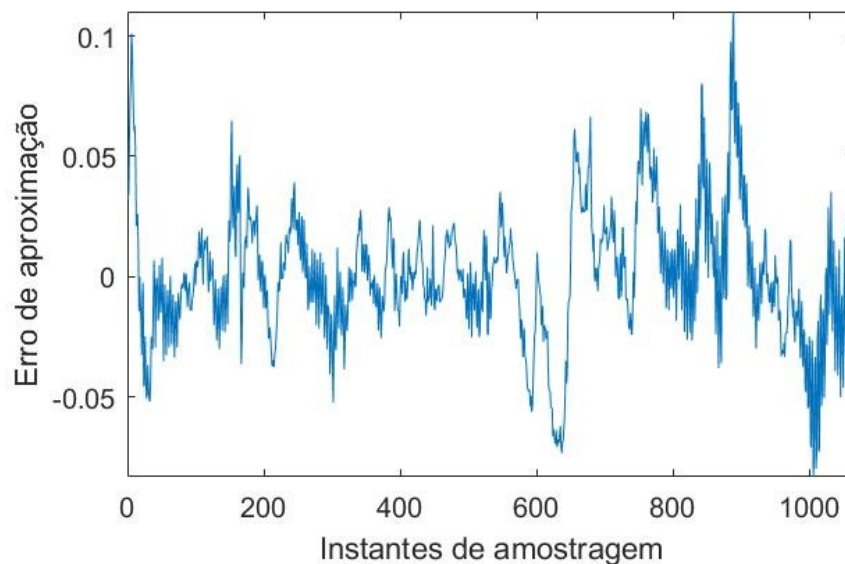


Figura 4.11. Erro entre saída do sistema $y(k)$ e saída do modelo $\hat{y}(k)$ utilizando funções de Kautz

As mesmas observações feitas para o caso de Laguerre na análise do gráfico 4.6 e da tabela 4.1 são válidas para esse caso também.

4.4 APRESENTAÇÃO E DISCUSSÃO DE RESULTADOS: GOBF

Por fim, discutiremos os resultados obtidos para o modelo OBF-Volterra com a GOBF de um pólo e um par de conjugados.

Durante a geração das funções, foi identificada uma falha devido ao método do algoritmo proposto, uma vez que no processo de refinamento da seleção de pólos, na terceira iteração observamos um erro menor do que na quarta. Como já discutido na seção de Kautz 4.3, melhorias serão propostas e discutidas no último capítulo desse trabalho.

Seguindo o procedimento adotado nos dois casos anteriores, os pólos obtidos que melhor representam o sistema em estudo foram $p_{r_1} = 0,8930$ e $p_{c_1} = 0,9680 \pm 0,0060i$ para o *kernel* de 1ª ordem e o conjunto de pólos $p_{r_2} = 0,8900$ e $p_{c_2} = 0,9680 \pm 0,0020i$ para o de 2ª ordem.

O comportamento tanto de $h_1(k_1)$ quanto de $h_2(k_1, k_2)$ é mostrado a seguir:

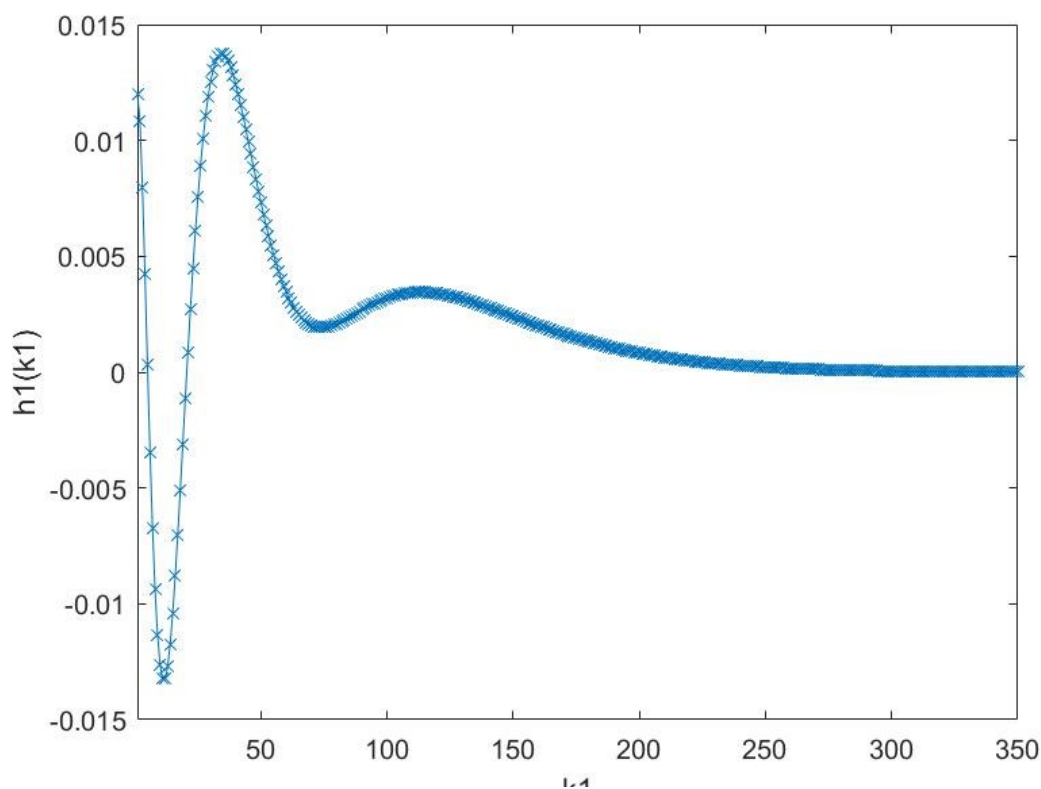


Figura 4.12. *Kernel* de 1ª ordem obtido para GOBF com pólos $p_{r_1} = 0,8930$ e $p_{c_1} = 0,9680 \pm 0,0060i$

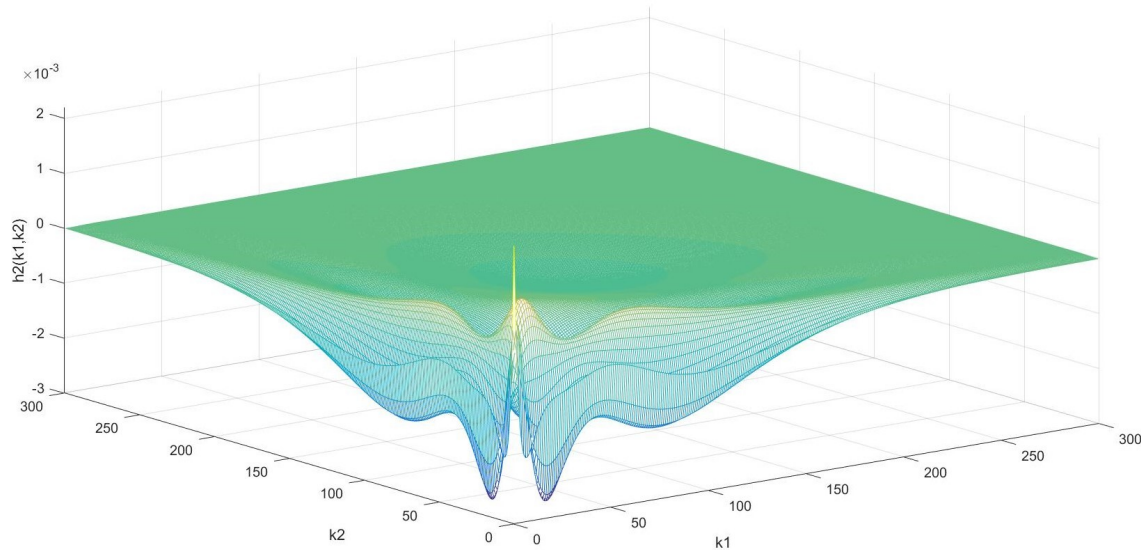


Figura 4.13. *Kernel* de 2^a ordem obtido para GOBF com pólos $p_{r_2} = 0,8900$ e $p_{c_2} = 0,9680 \pm 0,0020i$

A semelhança entre os *kernels* obtidos para a GOBF em questão e os obtidos para Laguerre, figuras 4.2 e 4.3, era esperada, especialmente após os resultados obtidos na etapa de seleção dos pólos. Uma vez que para a GOBF proposta, as quatro primeiras funções da base são iguais às quatro primeiras de Laguerre, dada a expressão (2.13). Essa semelhança será vista também mais à frente na apresentação dos resultados para os erros.

Com os *kernels* obtidos acima, obteve-se o seguinte resultado para o processo de identificação da reação de polimerização no CSTR, representado no gráfico 4.14, e para validação do modelo utilizando o período restante da reação, no 4.15:

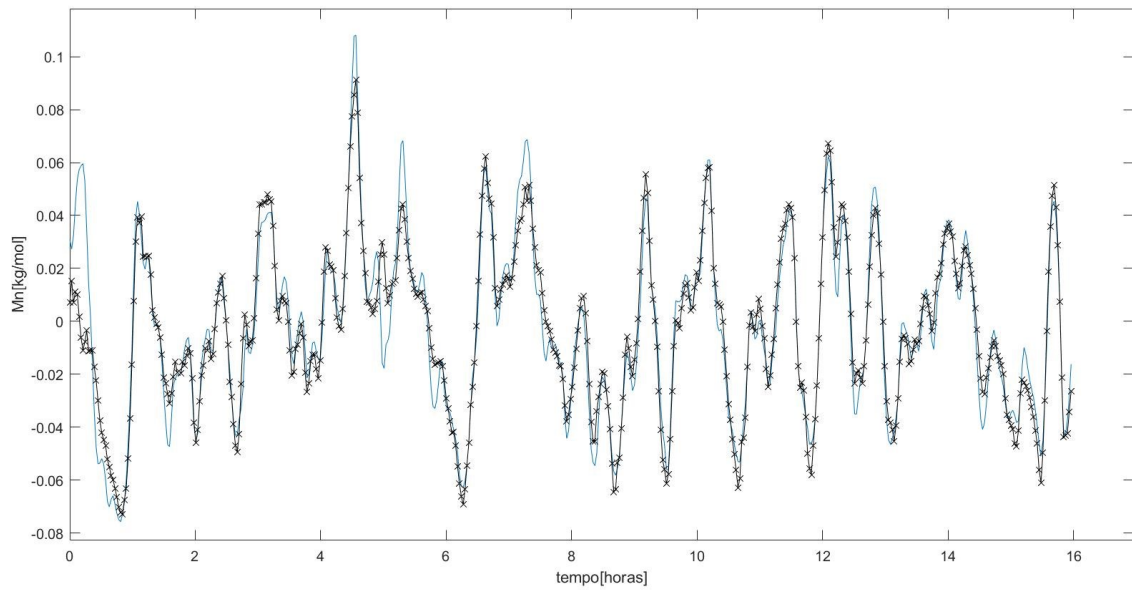


Figura 4.14. Função de saída do CSTR durante as primeiras 16 horas (linha contínua) e saída obtida a partir do modelo de Volterra com a GOBF proposta (linha com símbolo)

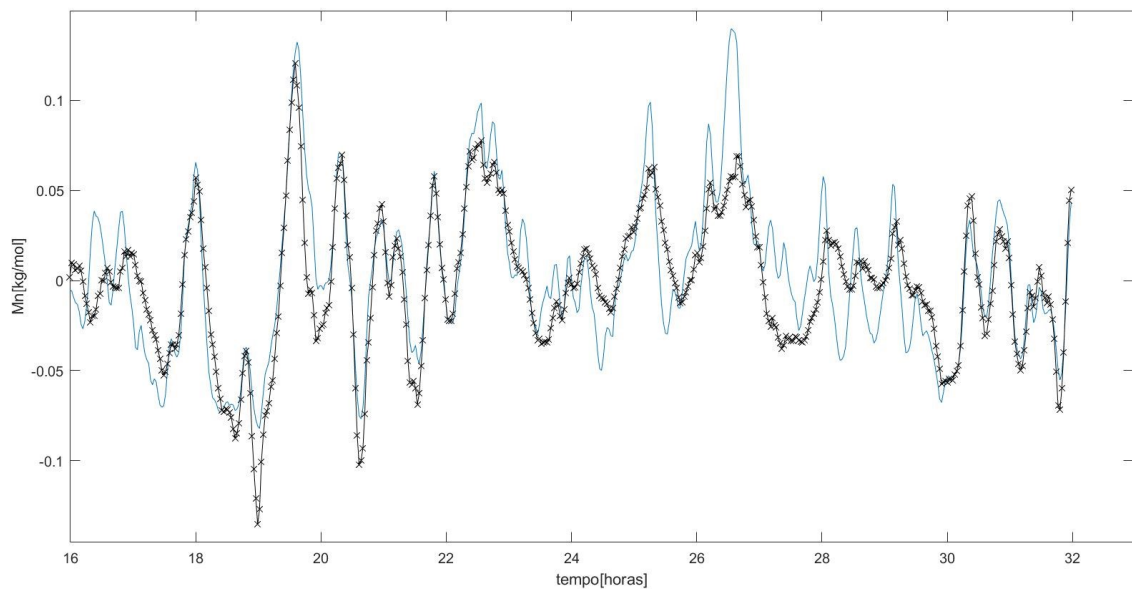


Figura 4.15. Função de saída do CSTR durante as 16 horas restantes (linha contínua) e saída obtida a partir do modelo de Volterra com a GOBF proposta (linha com símbolo)

E, como antecipado, os erros obtidos para o modelo OBF-Volterra utilizando essa base de funções, foram próximos daqueles obtidos com as de Laguerre, mas com uma pequena melhora. Deve-se ressaltar também que dadas as limitações do algoritmo proposto e essa ligeira melhora na precisão, avalia-se que há um bom potencial de melhoria na representação de sistemas não lineares de comportamento dinâmico com ordem maior que dois utilizando tanto essa GOBF quanto variações dela, conforme expressões (2.13), (2.14) e (2.18).

	EQM	EQN (dB)
Estimação	$1,0161 \times 10^{-4}$	-10,1739
Validação	$5,2368 \times 10^{-4}$	-5,5621

Tabela 4.3. Performance da simulação do modelo de OBF-Volterra com a GOBF proposta

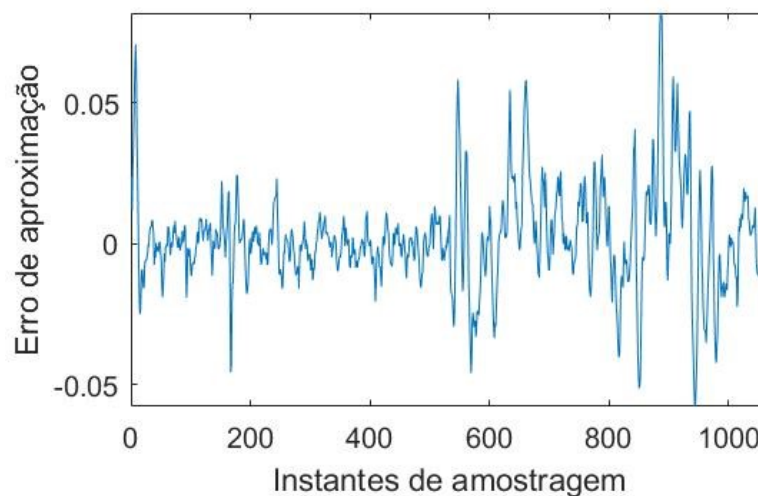


Figura 4.16. Erro entre saída do sistema $y(k)$ e saída do modelo $\hat{y}(k)$ utilizando a GOBF proposta

4.5 COMPARAÇÃO DE RESULTADOS ENTRE AS TRÊS BASES PROPOSTAS

Durante as seções anteriores já foram feitas algumas ponderações e discussões entre os resultados obtidos para cada uma das bases de funções ortonormais utilizadas, Laguerre, Kautz e a

GOBF proposta, e inclusive algumas comparações entre elas já foram introduzidas. O objetivo dessa seção será apresentar os resultados compilados dessas três bases e apontar vantagens e desvantagens comparativas entre elas. Para facilitar a discussão, iremos expor os resultados tanto da etapa de recuperação de estados quanto da validação do modelo na forma de tabela.

	Kernel de 1 ^a ordem	Kernel de 2 ^a ordem
Laguerre	$p = 0,9113$	$p = 0,9113$
Kautz	$\beta = 0,9000$	$\beta = -0,7000 \pm 0,7000i$
GOBF	$p_r = 0,8930$ e $p_c = 0,9680 \pm 0,0060i$	$p_r = 0,8900$ e $p_c = 0,9680 \pm 0,0020i$

Tabela 4.4. Pólos obtidos que melhor identificam a função de saída do CSTR através do modelo OBF-Volterra para Laguerre, Kautz e GOBF proposta

	EQM	EQN(dB)
Laguerre	$1,1254 \times 10^{-4}$	-9,7301
Kautz	$4,1370 \times 10^{-4}$	-4,0764
GOBF	$1,0161 \times 10^{-4}$	-10,1739

Tabela 4.5. Comparação dos erros EQM e EQN durante a etapa de identificação através do modelo OBF-Volterra para Laguerre, Kautz e GOBF proposta

	EQM	EQN(dB)
Laguerre	$7,0830 \times 10^{-4}$	-4,2506
Kautz	0,0011	-2,3582
GOBF	$5,2368 \times 10^{-4}$	-5,5621

Tabela 4.6. Comparação dos erros EQM e EQN durante a etapa de validação do modelo OBF-Volterra para Laguerre, Kautz e GOBF proposta

Observa-se que a localização dos pólos obtidos para os *kernels* de 1^a ordem no plano z são bem próximos entre si nos três casos. Além disso, suas localizações são próximas da circunferência unitária, ou seja, próximas da região de instabilidade no plano z , indicando a necessidade de um maior número de termos. Isso mostra que dentro do processo de seleção de cada base, as funções que melhor aproximaram o sistema foram funções que necessitaram de um maior número de amostras para convergir. Isso é resultado, principalmente, da função que se deseja identificar, no caso, da saída do CSTR, uma vez que se trata de um função NLIT de 4^a ordem, como a representação no espaço de estados mostrada expressões (4.1) e (4.2) mostram.

Já para o kernel de 2^a ordem, o pólo de Kautz se distancia dos demais, dada a limitação e falha observada no algoritmo, já discutida na seção 4.3.

Outro ponto a se observar é a proximidade entre a localização do pólo obtido para Laguerre e do pólo real obtido para a GOBF em questão, reforçando a contribuição das quatro primeiras funções de Laguerre na base GOBF proposta. Contudo, a inserção do par de conjugados nessa última, tanto para o kernel de 1^a ordem quanto para o de 2^a resultou em uma melhora no processo de identificação e, naturalmente, na etapa de validação também, como as tabelas 4.5 e 4.6 mostram.

Portanto a inserção de um par de conjugados com parte imaginária de pequena contribuição, já resultou em uma melhora no processo de identificação. E também foi propiciada uma maior melhoria na representação da reação na etapa de validação, quando comparamos a diferença entre os erros das três bases de funções, o que pode indicar uma melhor robustez da GOBF quando comparada às demais.

CONCLUSÕES E PROPOSTAS FUTURAS

Um dos principais objetivos desse trabalho foi acrescentar um novo tópico de discussão para a identificação de sistemas NLIT através do modelo OBF-Volterra de 2ª ordem, no caso, a utilização da base GOBF proposta, com um pólo real e um par de conjugados. Para tal, tomou-se um caso real já utilizado na literatura que foi o da reação de polimerização de MMA com AINBD utilizando tolueno como solvente em um CSTR. Para essa reação, foram simulados a representação dela através do modelo OBF-Volterra utilizando os três tipos de bases e comparados os resultados entre elas.

Avalia-se que foi satisfatório o resultado do processo, especialmente quando se olha esse objetivo, instigar um novo tópico de discussão. Houve uma melhora ao utilizarmos a GOBF proposta como base em comparação com as de Laguerre e de Kautz dentro da metodologia descrita no capítulo 3, tanto na identificação da reação quanto na validação do modelo, o que atendeu, em certa parte, as expectativas. E, tendo em perspectiva as limitações da metodologia discutidas no capítulo 4, a discussão do uso da GOBF para sistemas cuja dinâmica seja mais complexa se mostra ainda mais relevante, uma vez que foram obtidos melhores resultados, especialmente na validação do modelo, mesmo com essas limitações.

Dentre as quais, a principal limitação da metodologia adotada foi o grande custo computacional para a implementação das funções ortonormais. Essa questão foi, além de um desafio, um fator limitador do trabalho, uma vez que trouxe um prejuízo na própria identificação, como discutido principalmente na seção 4.3, em que ficou bem evidente que os resultados deveriam ter sido melhores para as funções de Kautz e não o foram em decorrência dessa deficiência. Dessa forma, para propostas futuras, a mudança de uma busca exaustiva para algoritmos mais sofisticados, como de gradiente descendente adotada em trabalhos como (DA ROSA *et al.*, 2009) , se mostram necessárias, não só pela redução de custo computacional, mas pela própria melhoria do processo de representação e de identificação de sistemas NLIT. Também a inser-

ção de de incertezas já na construção do modelo também é um meio de se melhorar o custo computacional (MAYER, 2021)).

Outro ponto de melhora é a utilização de modelos matemáticos mais complexos para a representação de plantas industriais. Nesse trabalho, foram adotadas algumas premissas simplificadoras, enumeradas na seção 4.1, além de ferramentas matemáticas que também simplificaram o processo. E naturalmente essas simplificações trazem um prejuízo na fidelidade do modelo. Utilização de modelos estocásticos como em (VILLANI *et al.*, 2019)) trazem uma maior fidelidade e também uma melhora de resultados, por exemplo.

Além disso, o emprego de GOBFs cuja construção seja feita com um maior número de pólos, seja com a inserção de mais pólos reais, seja com outros pares de conjugados, também são pontos a serem explorados. O próprio estudo de caso desse trabalho é um modelo de ordem quatro, assim, a utilização de GOBFs com quatro pólos em sua construção poderia propiciar uma melhora na identificação da reação.

Além dessas, outras propostas pertinentes são a de utilização de modelos OBF-Volterra de ordem maior que a segunda, algo pouco explorado na academia (BILLINGS, 1980); DUMONT; FU, 1993)). E também a utilização de ferramentas matemáticas mais avançadas que permitam a obtenção de expressões analíticas para o erro, como feito no trabalho (DA ROSA *et al.*, 2007)) para as funções de Kautz, para a representação do modelo de Volterra com a GOBF, tanto a proposta com um pólo real e um par de conjugados, quanto para outras variações.

REFERÊNCIAS BIBLIOGRÁFICAS

- BELT, H.; DEN BRINKER, A. *Optimality condition for truncated generalized Laguerre networks*. [S.l.], 1995). 227–235 f. Citado na página 7.
- BILLINGS, S. *Identification of nonlinear systems - a survey*. [S.l.], 1980). 272–285 f. Citado 4 vezes nas páginas 4, 5, 14, and 39.
- BOYD, S.; CHUA, L. *Fading memory and the problem of approximating nonlinear operators with Volterra series*. [S.l.], 1985). 1150–1161 f. Citado na página 5.
- DA ROSA, A.; CAMPELLO, R. J.; AMARAL, W. *Choice of free parameters in expansions of discrete-time Volterra models using Kautz functions*. [S.l.], 2007). 1084–1091 f. Citado na página 39.
- DA ROSA, A.; CAMPELLO, R. J.; AMARAL, W. *Exact Search Directions for Optimization of Linear and Nonlinear Models Based on Generalized Orthonormal Functions*. [S.l.], 2009). 2757–2772 f. Citado 2 vezes nas páginas 2 and 38.
- DOYLE, F. J.; OGUNNAIKE, B. A.; PEARSON, R. K. *Nonlinear Model-based Control Using Second-order Volterra Models*. [S.l.], 1995). 697–714 f. Citado 3 vezes nas páginas 21, 22, and 23.
- DOYLE, F. J.; OGUNNAIKE, B. A.; PEARSON, R. K. *Identification and Control Using Volterra Models*. [S.l.], 2002). Citado 4 vezes nas páginas 2, 6, 14, and 21.
- DRONGELEN, v. W. *Signal Processing for Neuroscientists, A Companion Volume: Advanced Topics, Nonlinear Techniques and Multi-Channel Analysis*. Chicago, USA: Elsevier, 2010. Citado 2 vezes nas páginas 1 and 4.
- DUMONT, G.; FU, Y. *Non-linear adaptive control via Laguerre expansion of Volterra kernels*. [S.l.], 1993). 367–382 f. Citado 5 vezes nas páginas 2, 4, 5, 14, and 39.
- FU, Y.; DUMONT, G. *An optimum time scale for discrete Laguerre network*. [S.l.], 1993). 934–938 f. Citado na página 7.
- HEUBERGER, P.; VAN DEN HOF, P.; BOKOR, J. *Modelling and Identification with Rational Orthogonal Basis Functions*. London, UK: Springer-Verlag, 2005. Citado 6 vezes nas páginas 2, 6, 7, 8, 9, and 10.
- LJUNG, L. *System Identification: Theory for the user Second Edition*. California, USA: Prentice Hall, 1999. Citado na página 1.
- MACHADO, J. B.; CAMPELLO, R. J.; AMARAL, W. *Asymmetric Volterra Models Based on Ladder-Structured Generalized Orthonormal Basis Functions*. [S.l.], 2007). 2879 – 2891 f. Citado 2 vezes nas páginas 7 and 10.

- MANER, B.; DOYLE III, F.; OGUNNAIKE, B.; PEARSON, R. *Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models*. [S.l.], 1996). 1285–1301 f. Citado na página 22.
- MAYER, D. *Orthonormal Filters for Identification in Active Control Systems*. [S.l.], 2021). Citado na página 39.
- NINNESS, B.; GUSTAFSSON, F. *A unifying construction of orthonormal bases for system identification*. [S.l.], 1997). 515–521 f. Citado 6 vezes nas páginas 2, 6, 7, 8, 11, and 12.
- NISE, N. S. *Control Systems of Engineering Fifth Edition*. California, USA: John Wiley & Sons, Inc., 2008. Citado na página 1.
- OLIVEIRA E SILVA, T. *Optimality conditions for truncated Laguerre networks*. [S.l.], 1994). 2528–2530 f. Citado na página 7.
- OLIVEIRA, G. H.; DA ROSA, A.; CAMPELLO, R. J.; MACHADO, J. B.; AMARAL, W. *An introduction to models based on Laguerre, Kautz and other related orthonormal functions – part I: linear and uncertain models*. [S.l.], 2011). 121–132 f. Citado 4 vezes nas páginas 6, 7, 19, and 21.
- OLIVEIRA, G. H.; DA ROSA, A.; CAMPELLO, R. J.; MACHADO, J. B.; AMARAL, W. C. *An introduction to models based on Laguerre, Kautz and other related orthonormal functions – part II: non-linear models*. [S.l.], 2012). 1–14 f. Citado 4 vezes nas páginas 2, 14, 19, and 21.
- PAIGE, C. C.; SAUNDERS, M. A. *LSQR: An Algorithm for Sparse Linear Equations And Sparse Least Squares*. [S.l.], 1982). 43–71 f. Citado na página 17.
- RUGH, W. *Nonlinear System Theory: The Volterra/Wiener Approach*. [S.l.], 1981). Citado 3 vezes nas páginas 4, 14, and 15.
- SCHETZEN, M. *The Volterra and Wiener Theories of Nonlinear Systems, Reprint Edition with Additional Material*. Malabar, Florida, USA: Krieger Publishing Company, 2006. Citado 5 vezes nas páginas 4, 5, 6, 14, and 15.
- TANGUY, N.; VILBÉ, P.; CALVEZ, L. *Optimum choice of free parameter in orthonormal approximations*. [S.l.], 1995). 1811–1813 f. Citado na página 7.
- TANGUY N., M. R. V. P.; CALVEZ, L. *Pertinent choice of parameters for discrete Kautz approximation*. [S.l.], 2002). 783–787 f. Citado na página 9.
- TIELS, K.; SCHOUKENS, J. *Wiener system identification with generalized orthonormal basis functions*. [S.l.], 2014). 3147–3154 f. Citado 2 vezes nas páginas 7 and 13.
- Van Den Hof, P.; HEUBERGER, P.; BOKOR, J. *System identification with generalized orthonormal basis functions*. [S.l.], 1995). 1821–1834 f. Citado na página 10.
- VILLANI, L.; CUNHA, A.; DA SILVA, S. *On the detection of a nonlinear damage in an uncertain nonlinear beam using stochastic Volterra series*. [S.l.], 2019). 1137–1150 f. Citado na página 39.
- WAHLBERG, B. *System identification using Kautz models*. [S.l.], 1994). 1276–1282 f. Citado na página 9.

ZIAEI, K.; WANG, D. *Application of orthonormal basis functions for identification of flexible-link manipulators*. [S.l.], 2006). 99-106 f. Citado na página 12.

APÊNDICE A

CÓDIGOS

Todos os códigos aqui anexados são de autoria do autor desse trabalho e foram desenvolvidos na plataforma do MATLAB®.

A.1 GERAÇÃO DA BASE DE FUNÇÕES ORTONORMAIS

```
function [C] = Geracao
%Geracao dos coeficientes , cada linha da matriz representa a funcao de
%laguerre gerada para um polo%

eqmgf = 1000000;
q=4;
linf1 = -1;
incl = .1;
lsup1=.9;
maxi = 1;

linf2 = -1;
inc2 = .1;
lsup2=.9;
maxi2 = 1;

linf3 = -1;
inc3 = .1;
lsup3=0;
maxi3 = .1;

o =10;

while((lsup1 < maxi ) || (lsup2 < maxi2) || (lsup3 < maxi3 + inc3 ) ||
      (lsup1 ==0 ) || (lsup2 == 0 ) || (lsup3 == 0 ))

    lsup1 = linf1 + 8*incl;
    lsup1 = round(lsup1*o)/o;
    if lsup1 >= maxi
        lsup1 =maxi +incl;
    end
```



```

lsup2 = linf2 + 8*inc2;
lsup2 = round(lsup2*o)/o;
if lsup2 >= maxi2
    lsup2 =maxi2 +inc2;
end

lsup3 = linf3 + 8*inc3;
lsup3 = round(lsup3*o)/o;
if lsup3 >= maxi3
    lsup3 =maxi3 + inc3;
end

syms z; %em que i eh o numero de funcoes
k = [1 zeros(1,2000)]; %geradas
a = 0;
t = 0;
C = cell(52,1);
PG = zeros(52,2);
for p = linf1:inc1:lsup1
    for m = linf2:inc2:lsup2
        for n = linf3:inc3:lsup3
            bn = m + n*1i;
            if (abs(bn) < 1 && abs(p)<1)
                a = a+1;
                PG(a,:) = [p,bn];

                lamb1 = abs(1-bn^2)/sqrt(1+abs(bn)^2);
                gama1= 0;
                lamb2 = -(bn + conj(bn))/sqrt(1+ abs(bn)^2);
                gama2 = sqrt(1 + abs(bn)^2) ;

            for g = 1:q+2
                P(g,:) = k;
            end

            x = (z^2 - (bn + conj(bn))* z + abs(bn)^2)* (z-p)^q;
            y1 = (lamb1*z^2 + gama1*z)*sqrt(1-abs(bn)^2) * (-p*z+1)^q;
            y2 = (lamb2*z^2 + gama2*z)*sqrt(1-abs(bn)^2) * (-p*z+1)^q;

            den = expand(x);
            d = sym2poly(den);
            num1 = expand(y1);
            n1=sym2poly(num1);
            num2 = expand(y2);
            n2= sym2poly(num2);

```

```

%Laguerre

for cont=1:q;
    num0 = sqrt(1-p^2)* z *(1 - p*z)^(cont-1);
    n0 = expand(num0);
    n0 = sym2poly(n0);

    den0 = (z-p)^cont;
    d0 = expand(den0);
    d0 = sym2poly(d0);
    f0 = filter(n0,d0,k);

    P(cont,:) = f0;

end

f1 = filter(n1,d,k);
f2 = filter(n2,d,k);
P(q+1,:) = f1;
P(q+2,:) = f2;
C(a) = {P};
else
    t = t+1;
end
end
end
end
end
end

```

A.2 ANÁLISE DE CONVERGÊNCIA

```

function [kc] = checonv (F)
%verificar se é uma matriz e se for, checar convergencia

h = size(F); %para todas as linhas
lin = h(1);
col = h(2);
kc = zeros(1,lin); %vetor de zeros para montagem de
                    %vetor de convergencia
for cv = 1:1:lin %transposta e verifica o comprimento
    count_max = fix(.1*col); %contagem maxima necessaria para
                            %confirmar convergencia
    c = 0; %variavel auxiliar para marcar instante
           %em que a convergencia confirma
           %ver instante de convergencia das funcoes
    for count = 1:col %contador para ir checando valores
                    %das funcoes ate a convergencia

```

```

H = F(cv, :);
if abs(H(count)) < .0001           %tolerancia
    c = c + 1;
    if c > count_max
        kc(cv) = count - count_max;
        break
    end
else
    c = 0;
end
end
end
end

```

A.3 FILTRAGEM DA FUNÇÃO DE ENTRADA PELA BASE DE FUNÇÕES ORTONORMAIS

```

function [CL] = matlagv3 (FL1, FL2, FL3, FL4, FL5, FL6)

t = size(FL1);
n = t(2);                                     %pegando as 6
                                              %primeiras

p = t(1);
CL = cell(p,1);
for l=1:1:p                                   %polo p
    a1 = FL1(l, :);
    a2 = FL2(l, :);
    a3 = FL3(l, :);
    a4 = FL4(l, :);
    a5 = FL5(l, :);
    a6 = FL6(l, :);
    A = [a1; a2; a3; a4; a5; a6];           %concatenação para montar
                                              %vetor do polo p em questao

    MFL = zeros(n,42);

    for k=1:1:n                               %ira pegar todos
                                              %as amostras k

        for q=1:1:6
            MFL(k,q) = A(q,k);
        end
        q = 7;
        for i1=1:1:6                          %pegar cada funcao de laguerre
            for i2=1:1:6
                MFL(k,q) = A(i1,k)*A(i2,k); %fazer produto dentro de
                %uma mesma amostra
            end
            q = q + 1;
        end
    end
end

```

```

end

CL(1) = {MFL};
end
%cada coluna representa um parametro que sera usado
%para calcular os 28 c

```

A.4 OBTENÇÃO DOS COEFICIENTES DE EXPANSÃO VIA MÉTODO DE MÍNIMOS QUADRADOS

```

function [cl, fll, rrl, itl, rvl, lsl] = coeflag (CL,y)
%funcao pra obter os coeficientes c
n = size(CL,1);
cl = zeros(42,n);           %pegar numero de polos e gerar
                           %vetor iniciais para serem preenchidos
fll = zeros(1,n);         %pelos retornos das funcoes de minimos
rrl = zeros(1,n);
itl = zeros(1,n);
rvl = zeros(100,n);
lsl = zeros(100,n);
for p=1:1:n
    a = CL(p);             %ir gerando os coeficientes
    A = cell2mat(a);
    [x, f, rr, it, rv, lv] = lsqr(A,y,1e-4,100);
    cl(:,p) = x;
    fll(1,p) = f;
    rrl(1,p) = rr;
    itl(1,p) = it;
    rvl(1:it+1,p) = rv;
end

```

A.5 GERAÇÃO DE TODOS OS KERNELS DE PRIMEIRA E DE SEGUNDA ORDEM PARA O DOMÍNIO DE PÓLOS ADOTADO

```

function [H1, H2] = kernels2n (c, L1, L2, L3, L4, L5, L6, kl)
%insiro todas as amostras das seis funcoes de laguerre, os coeficientes c
%relativos ao polo que gera menor residuo e o indice desse polo
q=1;
np = size(L1,1);
na = size(L1,2);

psi1 = zeros(np,na);
psi2 = zeros(np,na);
psi3 = zeros(np,na);
psi4 = zeros(np,na);
psi5 = zeros(np,na);

```

```

psi6 = zeros(np,na);

H0 = zeros(1,np); %%coluna diz qual eh o polo
H1 = zeros(na,np);
H2 = cell(1,np);

for p=1:1:np
    psi1(p,:) = L1(p, :);           %funcoes q geraram menor residuo
    psi2(p,:) = L2(p, :);
    psi3(p,:) = L3(p, :);
    psi4(p,:) = L4(p, :);
    psi5(p,:) = L5(p, :);
    psi6(p,:) = L6(p, :);
    psii = [psi1(p,:) ; psi2(p,:); psi3(p,:); psi4(p,:)
            psi5(p,:); psi6(p,:)];
    cli = c(1:6, p);                %cs de 1 ordem
    clij = c(7:42, p);              %cs de 2 ordem
    e = kl(:, p);
    e = max(e);
    HL2 = zeros(e, e);
    for k=1:1:e                       %geracao dos kernels de 1
        for i=1:1:6
            soma1 = cli(i)*psii(i,k); %kernel gerado p/ cada amostra
            H1(k,p) = soma1 + H1(k,p);
        end
    end
end

for k1=1:1:e                           %analogo para 2 ordem
    for k2=1:1:e
        for i1=1:1:6
            for i2=1:1:6
                soma2= clij(q)*psii(i1,k1)*psii(i2,k2);
                HL2(k1,k2) = soma2 + HL2(k1,k2);
                q = q+1;
            end
        end
        q = 1;
    end
end
H2(p) = {HL2};
end
H1 = H1';

```

A.6 SELEÇÃO DOS PÓLOS

```

function [yhat, pr1, pc1, pr2, pc2,Kernel1, Kernel2, eqmmin, ip1,ip2] =
selgobf(H1,Hp2,u1,ytil1 ,P)

```

```

np = size(H1,1); %numero de polos
t = length(u1); %numero de amostras da funcao exemplo
e = length(H1); %numero maximo de amostras do kernel
comb = np*np;
kern1 = zeros(t,np);
kern2 = zeros(t,np);
yapro = zeros(t,comb);
a=1;
polos = zeros(comb,2);
eqm = zeros(1,comb);

for p=1:1:np
    b = Hp2(p);
    H2 = cell2mat(b);
    lim = length(H2); %variavel auxiliar para
                    %pegar o maior numero

    for c = 1:1:t %"c" e "k" indica qual
                    %instante de amostragem

        k = c;
        for tau=1:1:e
            vez = H1(p, tau)* u1(k);
            kern1(c,p) = vez + kern1(c,p);
            k = k - 1;
            if k < 1
                break;
            end
        end
    end

    for tau1=1:1:lim
        k=c+1;
        for tau2=1:1:lim
            if (k-tau1 < 1) || (k-tau2 < 1)
                break;
            else
                vez = H2(tau1,tau2)* u1(k-tau1)*u1(k-tau2);
                kern2(c,p) = vez + kern2(c,p);
            end
        end
    end
end
end
for p2=1:1:np
    for p1=1:1:np
        for c = 1:1:t
            yapro(c,a) = kern1(c,p1) + kern2(c,p2);
        end
        polos(a,1) = p1;
        polos(a,2) = p2;
    end
end

```

```
        a = a+1;

    end

end

for n=1:1:comb
    eqm(n) = immse(ytil1 , yapro (: ,n));
end
[eqmmin , indmin] = min(eqm);
ip1 = polos (indmin ,1);
ip2 = polos (indmin ,2);
pr1 = P(ip1 ,1);
pc1 = P(ip1 ,2);
pr2 = P(ip2 ,1);
pc2 = P(ip2 ,2);
yhat = yapro (: , indmin);
Kernel1 = H1(ip1 ,:);
k2 = Hp2(ip2);
Kernel2 = cell2mat(k2);
```