

UNIVERSIDADE DE BRASÍLIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ALEXANDRE DE OLIVEIRA SILVA

**REPRESENTAÇÃO DE SISTEMAS NÃO-LINEARES POR MEIO DO MODELO DE
VOLTERRA DE TERCEIRA ORDEM COM FUNÇÕES DE LAGUERRE**

Brasília, DF

2024

ALEXANDRE DE OLIVEIRA SILVA

**REPRESENTAÇÃO DE SISTEMAS NÃO-LINEARES POR MEIO DO MODELO DE
VOLTERRA DE TERCEIRA ORDEM COM FUNÇÕES DE LAGUERRE**

Trabalho final de Graduação, apresentado a
Universidade de Brasília, como parte das
exigências para a obtenção do título de
Engenheiro Eletricista.

Orientador: Prof. Dr. Alex da Rosa

Brasília, DF

2024

FICHA CATALOGRÁFICA

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer a Deus por ter me dado a oportunidade de estudar em uma das melhores Universidades do país e acima de tudo por ter tido saúde ao longo desses anos em que estive na graduação.

À minha família, em especial os meus pais, Fabrício e Márcia, que sempre me deram todo o apoio e incentivo necessário para dar continuidade no curso. Também ao amor da minha vida, Ana, que nunca me deixou abater ou desanimar pelas dificuldades que encontrei ao longo do caminho.

A meus professores, em especial meu orientador que é um professor exemplar e nunca mediu esforços para me auxiliar em qualquer tipo de obstáculo ou dúvida que tive ao longo desse trabalho e da disciplina de circuitos elétricos 2 em que fui seu aluno e posteriormente monitor.

À UnB que sem dúvida alguma é uma das melhores Universidades do país e me proporcionou diversas oportunidades de aprendizado e ótimas amizades ao longo da minha trajetória.

Por último, mas não menos importante, gostaria de agradecer aos meus amigos que me apoiaram e acompanharam durante essa fase tão importante da minha vida.

RESUMO

Esse trabalho tem como foco a aplicação do Modelo de Volterra para sistemas de controle não-linear. Esse modelo tem como principal parâmetro os kernels que serão representados matematicamente por funções ortonormais, mais especificamente as funções de Laguerre. O sistema não linear escolhido foi uma reação química de polimerização em um reator de tanque agitado. Será utilizada a abordagem de caixa preta, explorando os dados de entrada e saída a partir do seu Modelo no Espaço de Estados. Teoricamente, o Modelo de Volterra tem ordem infinita, mas a grande maioria dos estudos utiliza-o até a 2º ordem. O objetivo desse trabalho será explorar o modelo de 3º ordem para compará-lo com o Modelo de 2º ordem para um mesmo sistema não-linear. O custo computacional é parte importante do trabalho e ao aumentar a ordem do modelo observou-se um aumento nessa importante variável do processo. Em contrapartida, houve uma diminuição nos erros calculados entre a saída real do sistema e a saída do Modelo de Volterra.

Palavras-chave: Modelo de Volterra; kernel; sistema não-linear, funções de Laguerre.

ABSTRACT

This work is focused on the application of the Volterra Model for nonlinear control systems. The main parameter of this model is called kernel and will be mathematically represented by orthonormal functions, more specifically by Laguerre functions. The nonlinear system chosen was a chemical polymerization reaction in a stirred tank reactor. The black box approach will be used exploring the input and output data only from its state space model. Theoretically the Volterra Model has infinite order, however the great majority of the works use the 2° order model. The objective of this work will be to use the 3° order model to compare the results with those of the 2° order model for the same non-linear system. The computational cost is an important part of the work, as the order of the model was increased, an increase in the computational cost was observed. On the other hand, a considerable decrease was observed in the calculated errors between the system output data and the Volterra Model output.

Keywords: Volterra Model; kernel; non-linear system, Laguerre functions

LISTA DE ILUSTRAÇÕES

Figura 1 - Região de convergência da Transformada Z	15
Figura 2 – Comportamento das funções de Laguerre	16
Figura 3 – Entrada degrau e saída real do sistema	24
Figura 4 - Representação de h_2 para $p_{ótimo}$	24
Figura 5 - Representação de h_1 para $p_{ótimo}$	24
<i>Figura 6 - Figura 5.4 - y_{real} e y_{modelo} para os Modelos de 2° e 3° ordens</i>	<i>25</i>
Figura 7 - Representação de $E(k)$ para o modelo de 2° e 3° ordens	25
<i>Figura 8 - h_2 do Modelo de 3° ordem para $p_{ótimo}$</i>	<i>26</i>
Figura 9 – h_1 do Modelo de 3° ordem para $p_{ótimo}$	26
<i>Figura 10 – y_{real} e y_{modelo} para os Modelos de 2° e 3° ordens</i>	<i>27</i>
Figura 11 – Representação de $E(k)$ para o modelo de 2° e 3° ordens	27

LISTA DE ABREVIATURAS

AIBN	Azobisisobutironitrila
CSTR	Reator de Tanque com Agitação Contínua
EDO	Equação Diferencial Ordinária
EQM	Erro Quadrático Médio
EQN	Erro Quadrático Médio Normalizado
MMA	Metacrilato de Metila
OFB	Bases de Funções Ortonormais
ROC	Região de Convergência
SISO	Uma entrada e uma saída
NFIR	Resposta ao impulso finita não-linear

LISTA DE SÍMBOLOS

δ	Impulso
β	Vetor de coeficientes
u	Entrada do sistema
p	Polo de Laguerre
Y	Vetor da saída real
H	Matriz de regressores
ψ_i	i-ésima função de Laguerre
z_n	n-ésima variável de estado
y_{real}	Saída real
y_{modelo}	Saída do Modelo OBF-Volterra
y	Saída do Modelo de Volterra
$p_{ótimo}$	Polo ótimo de Laguerre
l_i	i-ésima função de Laguerre filtrada pela entrada
h_n	Kernel de ordem n
\dot{z}_n	Derivada da n-ésima variável de estado

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Motivação	11
1.2	Organização do texto	12
2	MODELO DE VOLTERRA	14
2.1	Representações	14
2.2	Bases de Funções Ortonormais	16
2.2.1	Funções de Laguerre	17
2.3	Modelo OBF-Volterra	18
2.4	Otimização dos kernels e aplicações	19
3	APRESENTAÇÃO DO SISTEMA E RESULTADOS	21
3.1	Reação de polimerização em um CSTR	21
3.2	Algoritmos	22
3.2.1	Modelo de Espaço de Estados e funções de Laguerre	22
3.2.2	Estimação dos coeficientes e cálculo dos kernels	23
3.2.3	Cálculo da saída do Modelo de Volterra	24
3.2.4	Seleção do valor ótimo para o polo de Laguerre	24
3.3	Apresentação e discussão de resultados.....	24
3.3.1	Resposta a sequência de degraus	25
3.3.2	Resposta ao impulso $\delta(k)$	28
3.3.3	Comparação dos resultados	30
4	CONCLUSÕES E PROPOSTAS FUTURAS.....	32
	REFERÊNCIAS	33
	APÊNDICE A – CÓDIGOS	35
	APÊNDICE B – CÁLCULO DAS FUNÇÕES DE LAGUERRE.....	40

1 INTRODUÇÃO

1.1 Motivação

Problemas que envolvem sistemas não-lineares têm chamado muita atenção e gerado interesse de muitos físicos, matemáticos e engenheiros. Isso se dá porque a maior parte dos sistemas do mundo real possuem naturalmente uma dinâmica não-linear. Para possibilitar a representação desses sistemas, foram propostos muitos modelos matemáticos, dentre eles o Modelo de Volterra (CHENG *et al.*, 2017).

Entende-se por sistema não-linear aquele que não satisfaz o princípio da superposição. Para que um sistema satisfaça esse princípio, a saída do sistema para duas ou mais entradas quaisquer deverá ser igual a soma das saídas que teriam sido obtidas para cada entrada separadamente (DA SILVA, 2006).

Sistemas com dinâmica não-linear geralmente são complexos do ponto de vista matemático. Usualmente algumas simplificações são consideradas para que um determinado sistema originalmente não-linear possa ser analisado como um sistema linear. Dentre elas podemos citar: linearização em torno do ponto de equilíbrio, aproximação de pequeno sinal e a análise em pequenos intervalos (VIANA, 2021).

Nesse cenário, pode-se dizer que o sistema simplificado passa a se distanciar cada vez mais do sistema original na medida em que mais simplificações matemáticas são utilizadas. Em contrapartida, o estudo do sistema torna-se muito mais simples. É recomendado que haja um equilíbrio dessas simplificações para que a fidedignidade seja mantida, porém nem sempre é possível encontrar esse equilíbrio e o uso de técnicas não-lineares torna-se indispensável.

Existem diversos modelos matemáticos não-lineares como opção, sendo o Modelo de Volterra um dos mais populares atualmente e que surgiu nos trabalhos do matemático italiano Vito Volterra. Durante grande parte do século XX sua aplicação era muito limitada por ausência de tecnologia disponível e apenas a partir da criação e popularização dos computadores que esse modelo passou a ter ampla aplicação em áreas como: aeroelasticidade, dinâmica de fluidos, engenharia biomédica e engenharia elétrica (CHENG *et al.*, 2016).

As bases de funções ortonormais são extremamente úteis nesse contexto pois possuem propriedades que permitem diminuir a complexidade e o número de termos que surgem quando é feita a expansão do sistema pelo Modelo de Volterra e geralmente essas são apresentadas no domínio da frequência.

Entre essas famílias de funções, destacam-se as Funções de Laguerre que possuem em sua construção apenas múltiplos polos reais. Optou-se por utilizar essas funções para representar os kernels, considerados o núcleo do Modelo de Volterra, porque têm uma construção mais simples e geralmente satisfazem uma grande variedade de equações diferenciais (DA ROSA; CAMPELLO; AMARAL, 2009).

Além disso essas funções geralmente são apropriadas para representar sistemas dinâmicos bem amortecidos pois envolvem funções de transferência racionais com uma forma recursiva simples e completamente parametrizada por um único polo de valor real, o polo de Laguerre p (DA ROSA; CAMPELLO; AMARAL, 2009).

Um dos principais objetivos desse trabalho será calcular o valor ótimo para polo de Laguerre do sistema em estudo. Já que a principal proposta do trabalho é abordar o Modelo de Volterra até a 3° ordem, optou-se por estimar o polo ótimo por meio de uma busca exaustiva simples.

Quando os kernels do Modelo de Volterra são representados por funções ortonormais, dá-se o nome para esse modelo de OBF-Volterra. Visando simplificar esse modelo foram criadas versões das funções de Laguerre filtradas pela entrada do sistema, assim o cálculo dos coeficientes das combinações das funções de Laguerre para representar os kernels torna-se mais simples e pode ser feito a partir do método dos mínimos quadrados.

Por fim, comparou-se os erros entra saída real e as saídas dos Modelos de Volterra de 2° e 3° ordens, visando verificar a viabilidade de se utilizar a 3° ordem, já que há um aumento do custo computacional envolvido nas soluções numéricas do problema. Como parâmetro de comparação utilizou-se o erro quadrático médio (EQM) e o erro quadrático médio normalizado (EQN).

É importante comentar que foi escrito um artigo científico como consequência desse trabalho , em conjunto com o meu orientador, que será publicado no final do ano no Congresso Brasileiro de Automática (SILVA; DA ROSA, 2024).

1.2 Organização do texto

Este trabalho está organizado da seguinte maneira:

- No capítulo 2 apresentam-se: as representações contínuas e discretas do Modelo de Volterra, as funções ortonormais, o Modelo OBF-Volterra e por último são citados diversos trabalhos que se baseiam nesse modelo;

- No capítulo 3 apresentam-se: o sistema escolhido, os algoritmos feitos para simular o sistema e por fim os resultados dos Modelos de Volterra de 2° e 3° ordens;
- No capítulo 4 são apresentadas as conclusões do trabalho e as propostas para trabalhos futuros.

2 O MODELO DE VOLTERRA

2.1 Representações

Dentre os modelos que permitem uma caracterização sistemática de um sistema não-linear, o Modelo de Volterra é um dos mais consolidados. Essencialmente esse modelo é uma extensão da convolução tradicional, que é aplicável somente aos casos de sistemas lineares (CHENG *et al.*, 2016).

Esse modelo é caracterizado por utilizar uma sequência de operações, similares a convolução, entre o sinal de entrada $u(t)$ e as funções h_n conhecidas como kernels. Segue a representação do Modelo de Volterra para o tempo contínuo (DRONGELEN, 2010):

$$\begin{aligned} y(t) = & y_0 + \int_{-\infty}^{+\infty} h_1(\tau_1)u(t - \tau_1)d\tau_1 + \\ & + \iint_{-\infty-\infty}^{+\infty+\infty} h_2(\tau_1, \tau_2)u(t - \tau_1)u(t - \tau_2) d\tau_1 d\tau_2 + \dots \\ & + \iiint_{-\infty-\infty-\infty}^{+\infty+\infty+\infty} \dots \int_{-\infty}^{+\infty} h_n(\tau_1, \tau_2, \dots, \tau_n)u(t - \tau_1)u(t - \tau_2) \dots u(t - \tau_n) d\tau_1 d\tau_2 \dots d\tau_n \end{aligned} \quad (2.1)$$

Nesse modelo, $y(t)$ representa a saída do sistema, n indica a ordem dos kernels e o offset do sistema y_0 será considerado nulo pois as variáveis de estado sistema têm condições iniciais nulas. Com vistas a possibilitar a simulação computacional do sistema em análise, optou-se por utilizar a versão discreta do Modelo de Volterra neste trabalho, que tem a seguinte representação (SCHETZEN, 2006):

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

Será adotado um limite de truncamento ϵ_n na representação e os kernels de ordem n serão considerados nulos. O sistema que será analisado nesse trabalho é causal, ou seja, sua saída depende somente de valores da entrada no instante presente ou passados ou de valores passados da própria saída, isto é (BOYD; CHUA, 1985):

$$y(k) = 0; k < 0$$

Sabe-se que a versão truncada representa com ótima precisão qualquer sistema não-linear que seja causal, estável, invariante no tempo cuja entrada $u(k)$ possua energia limitada. Assim, o limite inferior dos somatórios será $\tau_n = 0$ e o Modelo será truncado (BOYD; CHUA, 1985):

$$\begin{aligned} y(k) = & y_0 + \sum_{\tau_1=0}^k h_1(\tau_1)u(k - \tau_1) + \\ & + \sum_{\tau_1=0}^k \sum_{\tau_2=0}^k h_2(\tau_1, \tau_2)u(k - \tau_1)u(k - \tau_2) + \dots \\ & + \sum_{\tau_1=0}^k \sum_{\tau_2=0}^k \dots \sum_{\tau_n=0}^k h_n(\tau_1, \tau_2, \dots, \tau_n)u(k - \tau_1)u(k - \tau_2) \dots u(k - \tau_n) \end{aligned} \quad (2.3)$$

Segue a representação do Modelo de Volterra de 3º ordem:

$$\begin{aligned} y(k) = & \sum_{\tau_1=0}^k h_1(\tau_1)u(k - \tau_1) + \\ & + \sum_{\tau_1=0}^k \sum_{\tau_2=0}^k h_2(\tau_1, \tau_2)u(k - \tau_1)u(k - \tau_2) + \\ & + \sum_{\tau_1=0}^k \sum_{\tau_2=0}^k \sum_{\tau_3=0}^k h_3(\tau_1, \tau_2, \tau_3)u(k - \tau_1)u(k - \tau_2)u(k - \tau_3) \end{aligned} \quad (2.4)$$

O sistema em questão é do tipo SISO (uma entrada e uma saída), portanto a função $u(k - \tau_n)$ representa diferentes amostras de uma mesma entrada. Assim, os kernels estão associados a um mesmo fator e podem ser considerados simétricos (SCHETZEN, 2006):

$$\begin{aligned} h_2(\tau_1, \tau_2) &= h_2(\tau_2, \tau_1) \\ h_3(\tau_3, \tau_2, \tau_1) &= h_3(\tau_3, \tau_1, \tau_2) = h_3(\tau_2, \tau_1, \tau_3) = h_3(\tau_2, \tau_3, \tau_1) = h_3(\tau_1, \tau_2, \tau_3) = \\ &= h_3(\tau_1, \tau_3, \tau_2) \end{aligned} \quad (2.5)$$

2.2 Bases de Funções Ortonormais

Uma família de funções ortonormais pode ser definida matematicamente da seguinte maneira:

$$\sum_{k=0}^{\infty} \psi_i(k) \psi_j(k) = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases} \quad (2.6)$$

Sendo $\psi_i(k)$ e $\psi_j(k)$ funções que pertencem à mesma base ortonormal. Uma propriedade importante dessas funções é a completude, que significa que qualquer função, dentro do espaço de Lebesgue $L^2[0, \infty)$, pode ser aproximada por uma combinação linear das funções que compõem a mesma base de funções ortonormais. Essa aproximação é baseada em uma precisão definida pelo modelo escolhido (OLIVEIRA *et al.*, 2011). Para que uma função $f(x)$ faça parte desse espaço de Lebesgue é necessário que a integral do quadrado do módulo de $f(x)$ sobre o intervalo $[0, \infty)$ seja finita, ou seja (GUIMARÃES *et al.*, 2006):

$$\int_0^{\infty} |f(x)|^2 dx < \infty \quad (2.7)$$

Geralmente as bases de funções ortonormais são definidas no domínio da frequência, logo será necessário calcular a transformada Z inversa dessas funções. Para que essas funções possam ser utilizadas na identificação ou representação de modelos, necessariamente seus polos

devem ter módulo menor do que 1 pois a região de convergência (ROC) da Transformada Z é a parte interna do círculo unitário como segue na Figura 3.1 (OLIVEIRA *et al*, 2012).

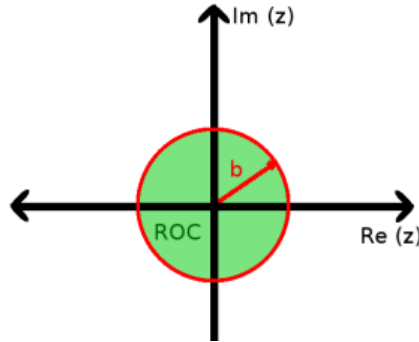


Figura 3.1. Região de convergência da transformada Z
Fonte: JUNIOR, et al., 2010

2.2.1 Funções de Laguerre

As funções de Laguerre possuem como característica ter somente polos reais múltiplos, p é chamado de polo de Laguerre e para que seja cumprida a condição de convergência, $|p| < 1$. Essa família de funções geralmente é representada da seguinte maneira (OLIVEIRA E SILVA, 1994):

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

Neste trabalho serão utilizadas as 6 primeiras funções de Laguerre e o cálculo detalhado da Transformada Z inversa dessas funções encontra-se no apêndice B. Apesar das funções de Laguerre serem mais adequadas para representar sistemas de 1° ordem, mostraram-se adequadas, quando comparadas com as funções de Kautz e as bases de funções ortonormais generalizadas GOBF, para o sistema em análise que é de 4° ordem. A Figura 3.2 mostra o comportamento das 6 funções para $p = 0,5$ (SILVA; DA ROSA, 2024).

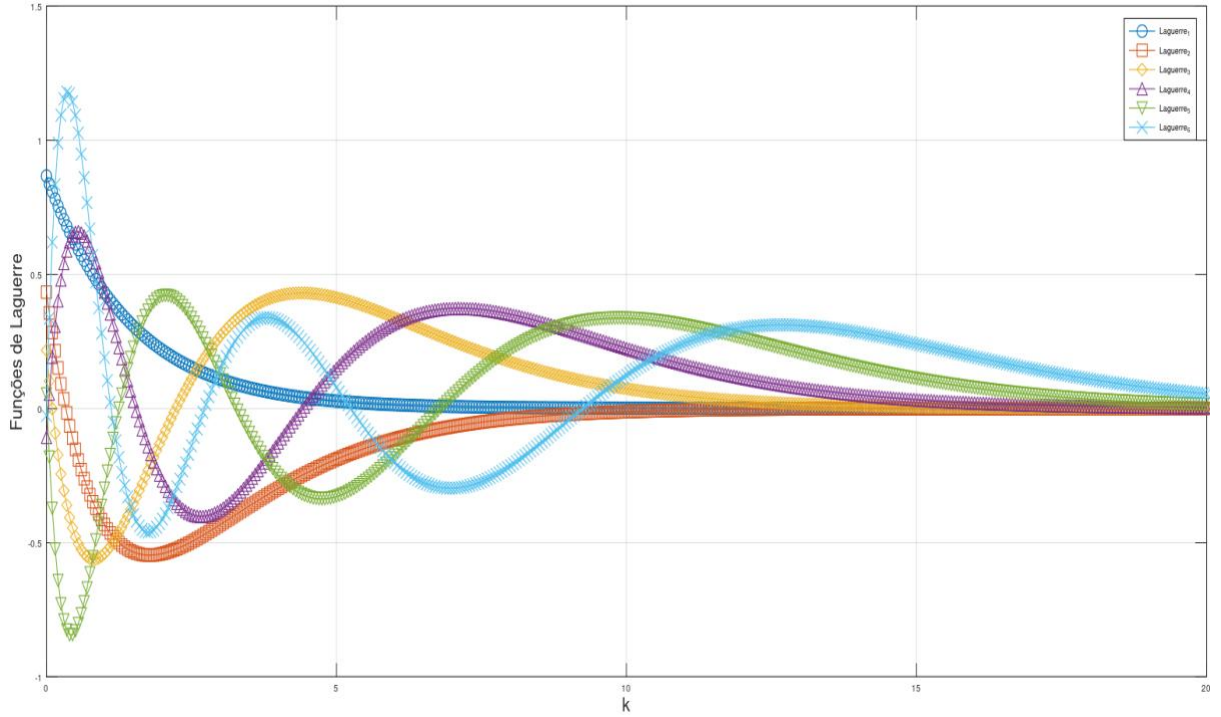


Figura 3.2 Comportamento das funções de Laguerre

2.3 Modelo OBF-Volterra

Esse Modelo é simplesmente o Modelo de Volterra tradicional, porém com a especificidade de que os kernels são expandidos através de uma base de funções ortonormais. Essa abordagem é vantajosa porque a utilização dessas funções reduz significativamente a quantidade de parâmetros a serem estimados (DOYLE et al., 2012).

Como consequência o custo computacional será muito menor e após ter calculado as Z^{-1} das funções de Laguerre, os kernels para o modelo de 3º ordem serão representados por uma combinação dessas funções da seguinte maneira (SCHETZEN, 2006; RUGH, 1981)):

$$\begin{aligned}
 h_1(\tau_1) &= \sum_{i=1}^n \beta_i \psi_i(\tau_1) \\
 h_2(\tau_1, \tau_2) &= \sum_{i=1}^n \sum_{j=1}^n \beta_{ij} \psi_i(\tau_1) \psi_j(\tau_2) \\
 h_3(\tau_1, \tau_2, \tau_3) &= \sum_{i=1}^n \sum_{j=1}^n \sum_{s=1}^n \beta_{ijs} \psi_i(\tau_1) \psi_j(\tau_2) \psi_s(\tau_3)
 \end{aligned}
 \tag{2.18}$$

Onde os coeficientes β são números reais que serão determinados computacionalmente e n é o número de funções de Laguerre. Ao aumentar a ordem do Modelo, a quantidade de termos dessas combinações cresce exponencialmente, em contrapartida a propriedade de simetria dos kernels diminui significativamente esse número. Ao combinar as expressões (2.4) e (2.18):

$$\begin{aligned}
y_{modelo}(k) = & \sum_{\tau_1=0}^k \sum_{i=1}^n \beta_i \psi_i(\tau_1) u(k - \tau_1) + \\
& + \sum_{\tau_1=0}^k \sum_{\tau_2=0}^k \sum_{i=1}^n \sum_{j=1}^n \beta_{ij} \psi_i(\tau_1) \psi_j(\tau_2) u(k - \tau_1) u(k - \tau_2) + \\
& + \sum_{\tau_1=0}^k \sum_{\tau_2=0}^k \sum_{\tau_3=0}^k \sum_{i=1}^n \sum_{j=1}^n \sum_{s=1}^n \beta_{ijs} \psi_i(\tau_1) \psi_j(\tau_2) \psi_s(\tau_3) u(k - \tau_1) u(k - \tau_2) u(k - \tau_3)
\end{aligned} \tag{2.19}$$

A fim de facilitar a visualização do Modelo e os cálculos computacionais, cria-se uma versão das funções de Laguerre filtrada pela entrada do sistema $l_i(k)$ e em seguida é feita a substituição dessa função em (2.19) (SCHETZEN, 2006; RUGH, 1981):

$$\begin{aligned}
\text{Onde, } l_i(k) = & \sum_{\tau=0}^k \psi_i(\tau) u(k - \tau) \\
y_{modelo}(k) = & \sum_{i=1}^6 \beta_i l_i(k) + \sum_{i=1}^6 \sum_{j=1}^6 \beta_{ij} l_i(k) l_j(k) + \sum_{i=1}^6 \sum_{j=1}^6 \sum_{s=1}^6 \beta_{ijs} l_i(k) l_j(k) l_s(k)
\end{aligned} \tag{2.20}$$

2.4 Otimização dos kernels e aplicações

O número de termos para representar os kernels é sempre muito grande, tornando sua identificação muito árdua. Se considerarmos um sistema com memória de N amostras, o Modelo de Volterra irá requerer N^p coeficientes para representar um kernel de ordem p (CHENG *et al.*, 2016).

Existem diversas técnicas que possibilitam diminuir consideravelmente esse número de coeficientes, dentre elas a expansão dos kernels em termos de bases de funções ortonormais. Campello e Favier abordaram um método de identificação de sistemas não-lineares que utiliza

as funções de Laguerre para chegar em uma expansão ótima das funções de resposta ao impulso de um sistema de Volterra (CAMPELLO; FAVIER; DO AMARAL, 2006). Zhang e Tischenko propuseram um novo algoritmo de controle adaptativo e utilizaram o Modelo de Volterra e expandiram os kernels por meio das funções ortonormais de Laguerre (ZHANG; TISCHENKO; YU, 2009).

É extremamente relevante levar em consideração os fatores não-lineares para atingir uma performance ideal de controle. O Modelo de Volterra é amplamente utilizado em problemas de controle em processos químicos, comunicação eletrônica, máquinas elétricas, compensação de distorções não-lineares, cancelamento de eco em canais de comunicação, modelagem de fala e até mesmo em processamento de imagens. Nesse contexto, inúmeros pesquisadores têm desenvolvido uma grande variedade de métodos efetivos de controle para sistemas não-lineares (CHENG *et al.*, 2016).

Yoon e Sun desenvolveram um método de identificação de sistemas não-lineares e utilizaram o Modelo de Volterra para o controle de rastreamento robusto de um atuador de um motor de válvula livre ou sem came (YOON, SUN, 2014). Gruber e Guzmán investigaram o design do controle não-linear preditivo da temperatura de uma estufa utilizando ventilação natural a partir do Modelo de Volterra de 2º ordem identificado experimentalmente a partir dos dados de entrada e saída da estufa (GRUBER *et al*, 2011).

Kumar e Banerjee apresentaram um novo sistema de classificação de imagens e utilizaram-no para reconhecimento facial. Os resultados desse estudo mostraram que a abordagem do Modelo de Volterra superou consistentemente inúmeros métodos de última geração da mesma categoria (KUMAR; BANERJEE; VEMURI, 2012). Le Caillec e Garello utilizaram uma decomposição baseada no Modelo de Volterra para investigar a modelagem de um radar de abertura sintética utilizado no processo de mapeamento da superfície do oceano (LE CAILLEC; GARELLO; CHAPRON, 2002).

Neste capítulo foram detalhados o Modelo de Volterra bem como as funções de Laguerre, além disso discutiu-se sobre: como otimizar os kernels, o modelo OBF-Volterra e aplicações do Modelo. No próximo capítulo o sistema a ser estudado será apresentado e serão expostos os resultados da representação desse sistema pelos Modelos de Volterra de 2º e 3º ordens.

3 APRESENTAÇÃO DO SISTEMA E RESULTADOS

Neste capítulo será detalhada a representação no espaço de estados de uma reação de polimerização em um reator de tanque agitado, em inglês *Continuous Stirred Tank Reactor* (CSTR), que será o sistema a ser simulado. Em seguida serão discutidos os resultados das simulações dos modelos de 2° e 3° ordens por meio de gráficos e Tabelas.

3.1 Reação de polimerização em um CSTR

Polimerização é uma reação em que pequenas moléculas denominadas monômeros se combinam quimicamente formando macromoléculas chamadas de polímeros. A polimerização pode ser feita por adição ou por condensação. Este trabalho trata da polimerização por adição, isto é: os polímeros são formados a partir da reação de adição de um único tipo de monômero (CANEVAROLO JR, 2002).

Por motivos de amplo estudo já documentado, o sistema escolhido para ser representado pelo Modelo de Volterra será uma reação de polimerização de radical livre de metacrilato de metila (MMA) como monômero com Azobisisobutironitrila (AIBN) como iniciador e Tolueno como solvente em um CSTR.

Visando simplificar a reação de polimerização, serão adotadas as seguintes simplificações: volume constante do Reator, operação Isotérmica, capacidade de calor constante, mistura perfeita, hipóteses de estado quase estacionário e de cadeia longa adotadas, sem reação de auto aceleração (efeito gel) e sem polímeros no fluxo de entrada (DOYLE *et al*, 1995).

Valendo-se dessas premissas, a representação no espaço de estados desse modelo foi obtida por (DOYLE *et al.*, 1995) tendo como entrada $u(t)$ o quociente de vazão do iniciador em $[m^3/h]$ e saída o peso molecular médio em número (M_n), $y_{real}(t)$ (kg/mol). As variáveis de estado são: z_1 : Concentração do monômero (MMA), z_2 : Concentração do iniciador (AIBN), z_3 : Peso total de todas as moléculas poliméricas da amostra e z_4 : Número total de moléculas da amostra.

Foi feita a normalização do modelo no espaço de estados por (DOYLE *et al.*, 1995), tornando as condições iniciais das variáveis de estado nulas, e uma aproximação por séries de Taylor truncadas em 2° ordem para fins de simplificação do cálculo numérico durante a simulação do sistema e obteve-se:

$$\left\{ \begin{array}{l} \dot{z}_1(t) = -10.8957z_1(t) - 0.447837z_2(t) - 0.447837z_1(t)z_2(t) + 0.111959z_2^2(t), \\ \dot{z}_2(t) = -10.1022z_2(t) + 10.1022u(t), \\ \dot{z}_3(t) = 2.4162z_1(t) + 8.7744z_2(t) - 10z_3(t) + 1.22581z_1(t)z_2(t) - 0.306453z_2^2(t), \\ \dot{z}_4(t) = 10z_1(t) + 5.00001z_2(t) - 10z_4(t) + 5.00001z_1(t)z_2(t) - 1.25z_2^2(t), \\ y_{real}(t) = -z_3(t) + z_4(t) - z_3(t)z_4(t) + z_3^2(t). \end{array} \right. \quad (4.1)$$

3.2 Algoritmos

Um dos motivos que tornam o Modelo de Volterra extremamente vantajoso computacionalmente é o fato dele ser linear em seus parâmetros, ou seja, apesar da saída ser não-linear em relação à entrada, os kernels são lineares em relação a saída. O que permite o uso de métodos computacionais mais simples, como o método dos mínimos quadrados, para as soluções de cálculo numérico (VAN DEN HOF *et al.*, 2000). Para a simulação do sistema a presença de ruídos na entrada e saída foram desconsiderados.

O custo computacional é extremamente relevante para esse trabalho. A máquina utilizada possui as seguintes configurações: modelo Acer Swift 3, sistema operacional: 64 bits, baseado em x64, Windows 11 Home Single Language, processador: 11ª geração intel® core™ i7-1165G7 @ 2.8GHz 2.8GHz, memória volátil: RAM 16GB e memória não volátil: SSD 1TB.

3.2.1 Modelo de Espaço de Estados e funções de Laguerre

O primeiro passo do trabalho é simplesmente obter a saída real do sistema y_{real} a partir do Modelo no Espaço de Estados. Foi criada uma função que calcula as derivadas das variáveis de estados a partir dos estados atuais do sistema e da entrada. É importante destacar o solver de Equações Diferenciais Ordinárias (EDO's) *ode45* selecionado para a solução do sistema que é baseado em um aprimoramento do Método de Runge-Kutta de 4º e 5º ordem. Esse solver retorna os valores das variáveis de estados para todo k. e o algoritmo dessa etapa pode ser visto no apêndice A.1.

As funções de Laguerre foram calculadas no apêndice B, assim foi possível gerá-las diretamente no ambiente OCTAVE e em seguida foram geradas as funções de Laguerre

filtradas pela entrada $l_i(k)$ que irão compor a matriz de regressores $H(k)$ e o algoritmo dessa etapa pode ser visto em A.2.

3.2.2 Estimação dos coeficientes e cálculo dos kernels

Essa é uma das partes mais relevantes do trabalho pois os kernels , considerados a essência do sistema, serão calculados. Utilizou-se o operador ‘\’ para a solução do sistema linear sobredeterminado $A.x = B$ que minimiza $|B - Ax|$. Para o problema, A será a matriz de regressores H , x o vetor de coeficientes β e B o vetor Y . O algoritmo dessa 1ª etapa pode ser visto em A.3.

$$H_{k+1,258} = \begin{bmatrix} l_1(0) & \dots & l_6(0) & l_1(0)^2 & l_1(0)l_2(0) & \dots & l_6(0)^2 & l_1(0)^3 & l_1(0)l_1(0)l_2(0) & \dots & l_6(0)^3 \\ l_1(1) & \dots & l_6(1) & l_1(1)^2 & l_1(1)l_2(1) & \dots & l_6(1)^2 & l_1(1)^3 & l_1(1)l_1(1)l_2(1) & \dots & l_6(1)^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_1(k) & \dots & l_6(k) & l_1(k)^2 & l_1(k)l_2(k) & \dots & l_6(k)^2 & l_1(k)^3 & l_1(k)l_1(k)l_2(k) & \dots & l_6(k)^3 \end{bmatrix} \quad (4.2)$$

$$\beta_{1,258} = [c_1 \quad \dots \quad c_6 \quad c_{1,1} \quad c_{1,2} \quad \dots \quad c_{6,6} \quad c_{1,1,1} \quad c_{1,1,2} \quad \dots \quad c_{6,6,6}] \quad (4.3)$$

$$Y_{k+1,1} = \begin{bmatrix} y_{real}(0) \\ y_{real}(1) \\ \vdots \\ y_{real}(k) \end{bmatrix} \quad (4.4)$$

Uma vez que β foi estimado, é possível calcular os kernels h_1 , h_2 e h_3 a partir das expressões (2.18). Cabe destaque a necessidade transformar cada coluna da matriz H em tensores para que seja possível calcular h_3 de maneira otimizada. O algoritmo dessa etapa pode ser visto em A.4.

3.2.3 Cálculo da saída do Modelo de Volterra

O código que foi utilizado nessa etapa foi desenvolvido por (GOULART, 2024) e está disponível no MATLAB Central File Exchange. Para que seja possível o cálculo da saída do Modelo de Volterra é necessário que os kernels e a entrada do sistema sejam fornecidos e por motivos de otimização, esse algoritmo realiza os cálculos no domínio da frequência.

3.2.4 Seleção do valor ótimo para o polo de Laguerre

Nessa etapa optou-se pelo método de busca exaustiva variando p no intervalo $]0,1[$ com precisão de duas casas decimais. O código para essa etapa não foi inserido no trabalho pois basicamente foi feito um loop externo a todos os cálculos citados nas seções anteriores para cada valor de p .

Utilizou-se o erro quadrático médio (EQM) e o erro quadrático médio normalizado (EQN) como parâmetro para seleção do valor ótimo de p , y_{real} é a saída real do sistema y_{modelo} é a saída do modelo (OLIVEIRA et al., 2012):

$$EQM = \frac{1}{N_d} \sum_{k=1}^{N_d} (y_{real}(k) - y_{modelo}(k))^2, \quad (4.5)$$

$$EQN \triangleq 10 \log \frac{\sum_{k=1}^{N_d} (y_{real}(k) - y_{modelo}(k))^2}{\sum_{k=1}^{N_d} (y_{real}(k))^2}, \quad (4.6)$$

3.3 Apresentação e discussão de resultados

Nesta seção serão apresentadas as saídas dos Modelos de Volterra de 2º e 3º ordens para duas entradas distintas: impulso $\delta(k)$ e uma sequência de degraus. Ao aumentar a ordem do Modelo de Volterra é esperado que o erro também diminua. O principal objetivo desse trabalho é fazer essa confirmação e verificar se essa diminuição é significativa em contrapartida de um custo computacional mais elevado. Além disso os resultados apresentados poderão nortear estudos de outros sistemas não-lineares representados pelo Modelo de Volterra.

Para que o cálculo da saída do Modelo de Volterra seja possível é necessário obter: a saída real do sistema y_{real} , as funções de Laguerre no domínio k , a solução do sistema linear $H.\beta = Y$ e os kernels h_1, h_2 e h_3 . Quanto maior p , mais lentamente os kernels irão convergir, isso é justificado pela maneira como as funções de Laguerre são construídas no domínio k e pode ser facilmente percebido graficamente.

Os kernels h_1 e h_2 podem ser facilmente representados, já a representação de h_3 torna-se muito mais complexa pois não é possível representá-lo em 2 ou 3 dimensões de forma direta. Existe a possibilidade de optar por fixar uma das variáveis e obter a função resultante em termos das outras 2 variáveis. Essa abordagem é conhecida como visualização por seções ou planos de corte e é útil para entender a variação da função em relação a diferentes combinações de variáveis.

O polo de Laguerre foi calculado simultaneamente para h_1, h_2 e h_3 e obteve-se: $p_{ótimo} = 0,05$ para o Modelo de 2º ordem e $p_{ótimo} = 0,08$ para o Modelo de 3º ordem. Destaca-se que houve um aumento considerável no custo computacional ao incluir o kernel $h_3(k_1, k_2, k_3)$. Assim, ao escolher o Modelo deve-se considerar se a precisão fornecida pelo de 2º ordem já é suficiente para o sistema em estudo.

A seguir serão apresentadas 2 seções, uma para cada entrada, onde serão mostrados gráficos da entrada e saída reais do sistema, representação gráfica dos kernels, gráfico simultâneo de y_{real} e y_{modelo} e por fim os cálculos dos erros EQM, EQN e $E(k)$. Define-se como $E(k)$:

$$E(k) = y_{real} - y_{modelo}$$

Para fins de comparação nas figuras 5.4, 5.5, 5.8 e 5.9 colocou-se os resultados dos dois modelos lado a lado, onde a imagem da esquerda representa o Modelo de 2º ordem e a da direita o Modelo de 3º ordem.

3.3.1 Resposta a sequência de degraus

É importante ressaltar que para fins de simulação a escolha da entrada $u(t)$ é arbitrária desde que possua energia limitada. Optou-se por escolher a entrada como um sinal aleatório com amplitude variando uniformemente no intervalo $[-0.6, 0.8]$ ao longo de 100 horas, com passo de amostra de 0,5 horas. Para a representação gráfica dos kernels utilizou-se um outro

tempo de simulação pois como $p_{ótimo} = 0,05$ então os kernels convergem muito rápido e não teriam amostras suficientes para ter uma boa representação gráfica.

Seguem abaixo:

- A Figura 5.1 que mostra o comportamento da entrada e saída durante a simulação;
- Os kernels h_1 e h_2 do Modelo de 3º ordem estão representados respectivamente pelas Figuras 5.2 e 5.3;
- A Figura 5.4 que mostra o gráfico simultâneo de y_{real} e y_{modelo} para os 2 modelos;
- O erro ao longo do tempo $E(k)$ para os 2 modelos na Figura 5.5;
- A Tabela 5.1 que mostra o comportamento de EQM e EQN para $p_{ótimo}$.

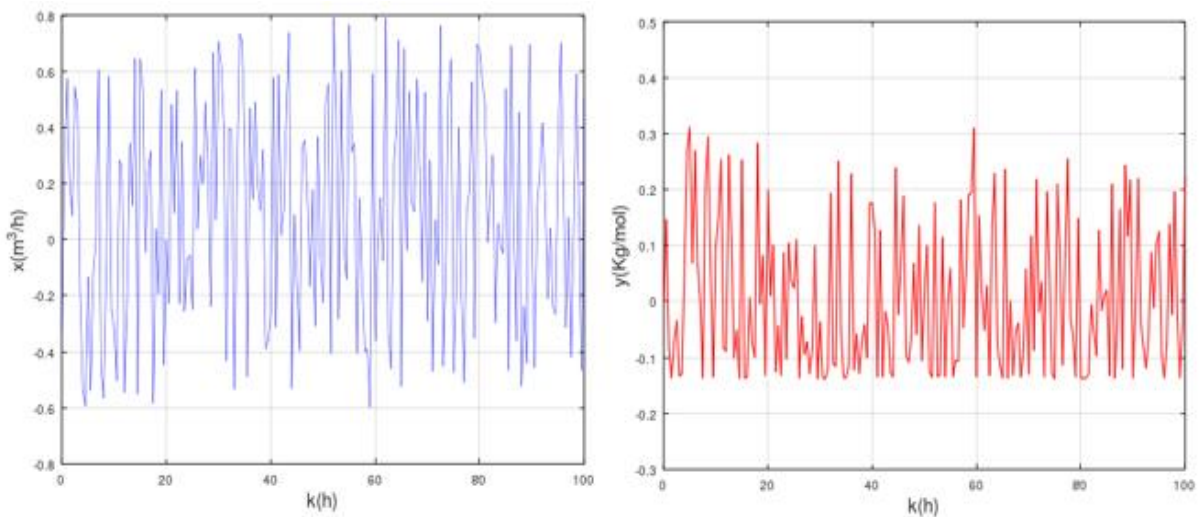


Figura 5.1- Entrada e saída $u(k)$ e $y(k)$

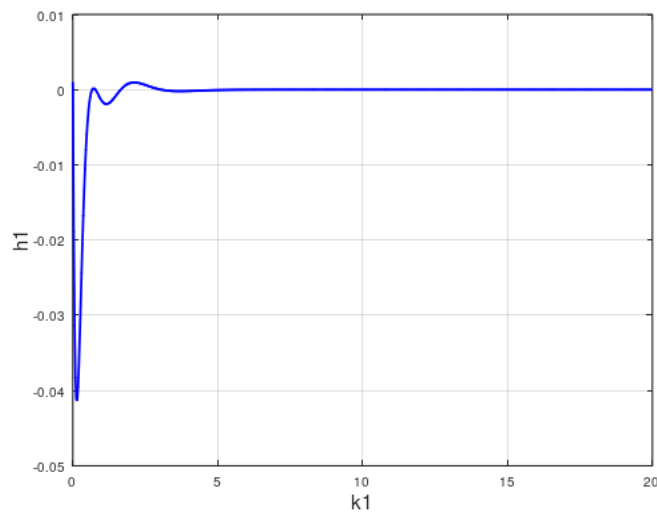


Figura 5.2 – Representação de h_1 para $p_{ótimo}$

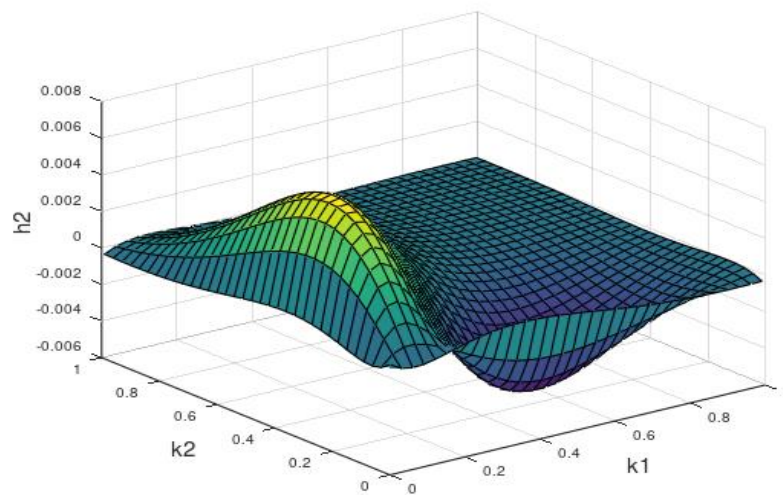


Figura 5.3 – Representação de h_2 para $p_{\text{ótimo}}$

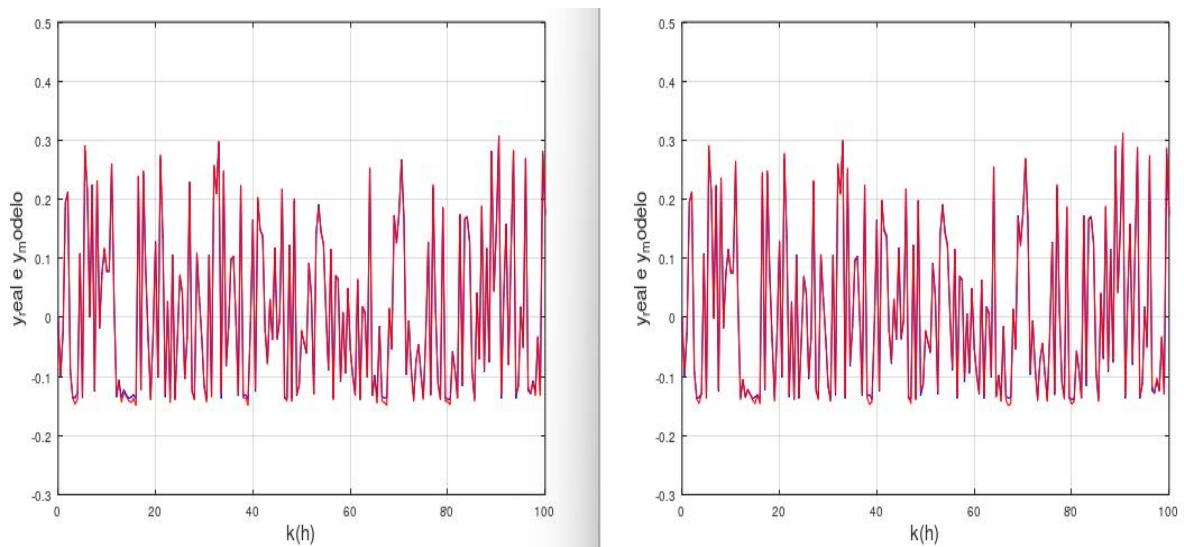


Figura 5.4 - y_{real} (em vermelho) e y_{modelo} (em azul) para os Modelos de 2° e 3° ordens

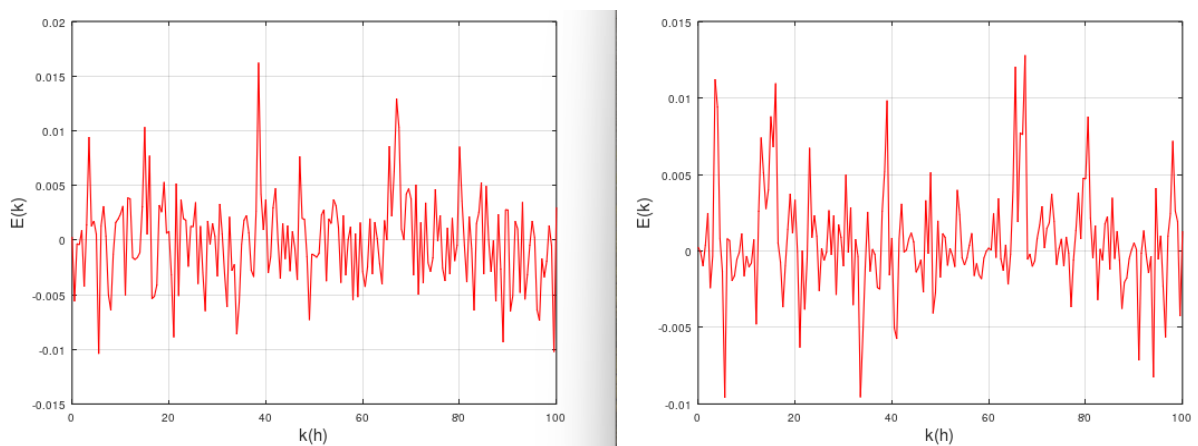


Figura 5.5 - Representação de $E(k)$ para os Modelos de 2° e 3° ordens

Tabela 5.1 - EQM e EQN para $p_{\text{ótimo}}$

Modelo	EQM	EQN
2° ordem	$1,3856 \times 10^{-5}$	-31,0259
3° ordem	$5,4752 \times 10^{-6}$	-35,0584

Foram feitas diversas simulações para a construção da Tabela e utilizou-se um valor médio de EQM e EQN . Quanto mais negativo for o valor EQN melhor será a precisão do modelo, isso se deve ao fato de EQN ser calculado utilizando uma escala logarítmica.

Por meio da Tabela e dos gráficos observa-se que o Modelo de 3° ordem teve uma performance melhor tanto para EQM quanto para EQN . Por outro lado, houve um aumento no custo computacional mas que não impossibilitou a simulação.

3.3.2 Resposta ao impulso $\delta(k)$

O Modelo OBF-Volterra é do tipo NFIR(*Non-linear finite impulse response*), possui resposta finita ao impulso $\delta(k)$. A resposta ao impulso $\delta(k)$ fornece uma visão parcial de como um sistema não-linear invariante no tempo reage a diferentes entradas e pode ser aplicado a uma ampla gama de sistemas (VAN DEN HOF *et al.*, 2000).

O impulso é uma função que pode ser facilmente implementada pois é simplesmente um pico de amplitude em uma única amostra. Essa simulação foi feita ao longo de 5 horas, com passo de amostra de 0,025 horas

Seguem abaixo:

- Os kernels h_2 e h_1 estão representados respectivamente pelas Figuras 5.6 e 5.7;
- A Figura 5.8 que mostra o gráfico simultâneo de y_{real} e y_{modelo} para os 2 modelos;
- O erro ao longo do tempo $E(k)$ para os 2 modelos na Figura 5.9;
- A Tabela 5.2 que mostra o comportamento de EQM e EQN para $p_{\text{ótimo}}$.

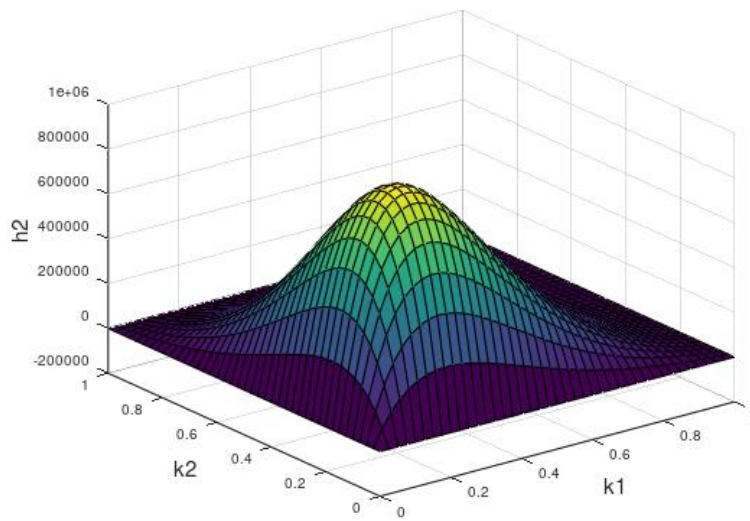


Figura 5.6- h_2 do Modelo de 3° ordem para $p_{ótimo}$

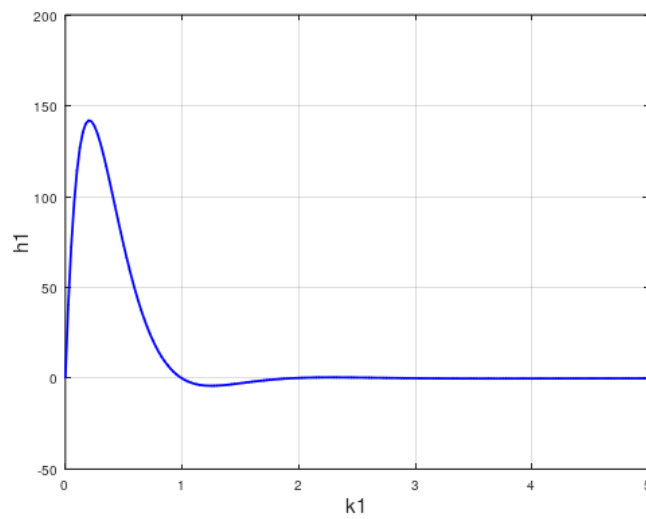


Figura 5.7- h_1 do Modelo de 3° ordem para $p_{ótimo}$

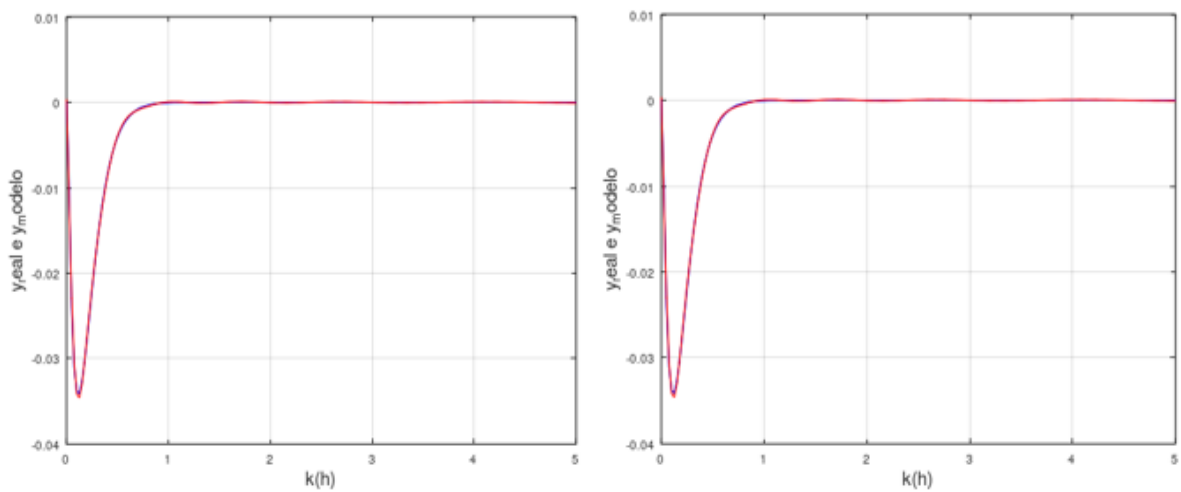


Figura 5.8 - y_{real} (em vermelho) e y_{modelo} (em azul) para os Modelos de 2° e 3° ordens

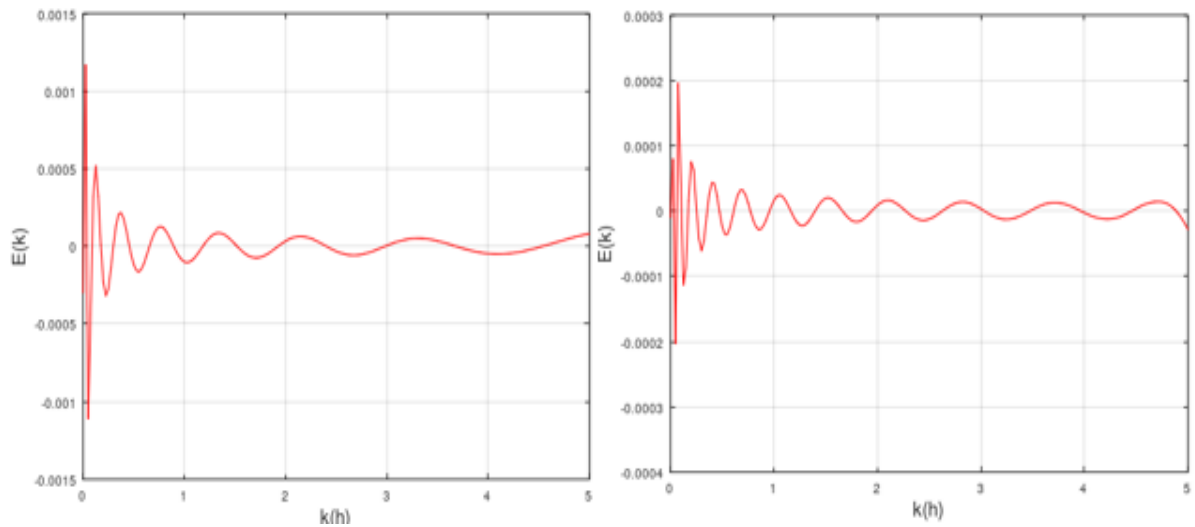


Figura 5.9 - Representação de $E(k)$ para os Modelos de 2° e 3°

Tabela 5.2 - EQM e EQN para $p_{ótimo}$

Modelo	EQM	$EQN(dB)$
2° ordem	$2,1381 \times 10^{-8}$	-33,2795
3° ordem	$8,1005 \times 10^{-10}$	-47,4946

É possível observar claramente a propriedade de simetria de h_2 graficamente. Assim como na resposta ao degrau, o modelo de 3° ordem apresentou uma performance melhor do que o modelo de 2° ordem. Visualmente o erro dos dois modelos parece ser muito próximo para a resposta ao $\delta(k)$, porém diferença entre os dois modelos foi muito maior do que na resposta ao degrau já que EQ43M foi cerca de 100 vezes menor para o modelo de 3° ordem.

3.3.3 Comparação dos resultados

Segue abaixo a Tabela 5.3 que resume EQM e EQN para ambas entradas e modelos. A tabela 5.4 mostra a comparação do custo computacional para os dois modelos variando a quantidade de amostras a partir da entrada degrau.

Tabela 5.3 - EQM e EQN

$u(k)$	2° ordem	2° ordem	3° ordem	3° ordem
	EQM	$EQN(dB)$	EQM	$EQN(dB)$
degrau	$1,3856 \times 10^{-5}$	-31,0259	$5,4752 \times 10^{-6}$	-35,0584
$\delta(k)$	$2,1381 \times 10^{-8}$	-33,2795	$8,1005 \times 10^{-10}$	-47,4946

Tabela 5.4 – Custo computacional para ambos os modelos

Número de amostras	2° ordem (<i>segundos</i>)	3° ordem (<i>segundos</i>)
100	2	8
150	9	36
200	15	91
250	19	167
300	23	199
350	27	353
400	36	605

De modo geral o Modelo de 3° ordem apresentou uma performance superior ao Modelo de 2° ordem apesar do aumento do custo computacional para as duas entradas simuladas. No próximo capítulo serão apresentadas as conclusões do trabalho assim como as propostas para trabalhos futuros.

4 CONCLUSÕES E PROPOSTAS FUTURAS

O principal objetivo desse trabalho é verificar se há uma melhoria na precisão do sistema ao utilizar o Modelo de Volterra de 3° ordem para representar sua saída. Além disso, analisar o aumento do custo computacional envolvido, uma vez que esse tema ainda foi pouco explorado. Como já foi mencionado, as funções de Laguerre geralmente são indicadas para representar sistemas de 1° ordem, entretanto obteve-se um resultado muito satisfatório utilizando-as para representar os kernels do sistema de 4° ordem analisado.

Por ser um *software* livre, a simulação do sistema foi totalmente feita na linguagem OCTAVE, a função *lsqr*, disponível no matlab, não está disponível para OCTAVE. Assim foi necessário utilizar como alternativa o operador ‘\’ para a solução do sistema linear sobredeterminado $H.\beta = Y$. Essa abordagem também foi muito satisfatória para a determinação dos coeficientes β e consequentemente para o cálculo dos kernels $h_1(\tau_1)$, $h_2(\tau_1, \tau_2)$ e $h_3(\tau_1, \tau_2, \tau_3)$.

Ao comparar os resultados obtidos pelos dois Modelos, *EQM*, *EQN* e $E(k)$ são de fato menores para o modelo de 3° ordem, todavia o custo computacional tornou-se mais elevado, o que fica evidenciado pela tabela 5.4. Assim conclui-se que é primordial considerar se o modelo de 2° ordem é suficiente para atender os requisitos de precisão para um determinado problema antes de se implementar um modelo de ordem superior.

Para obter conclusões mais sólidas sobre o assunto uma sugestão para futuros trabalhos é simular outros sistemas além do CSTR utilizando o Modelo de Volterra de 3° ordem e fazer a mesma comparação feita neste trabalho. Além disso sugere-se o uso de simuladores de processos químicos para validação dos resultados como, por exemplo, o *AVEVA Process Simulation*.

Outra proposta futura é utilizar outras bases de funções ortonormais, como a de Kautz e GOBF, para a representação dos kernels do Modelo de Volterra de 3° ordem e verificar se há uma melhoria na precisão ao comparar com os resultados obtidos utilizando as funções de Laguerre.

O cálculo do $p_{ótimo}$ também pode ser aprimorado. Foi feita apenas uma busca exaustiva simples neste trabalho, uma sugestão é utilizar o método de Levenberg-Marquardt que pode superar os métodos do gradiente descendente e gradiente conjugado para problemas de tamanho médio e é uma escolha usual em problemas de otimização não-linear (DA ROSA; CAMPELLO; AMARAL, 2009).

REFERÊNCIAS

- BOYD, Stephen; CHUA, Leon. Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on circuits and systems*, v. 32, n. 11, p. 1150-1161, 1985.
- CAMPELLO, Ricardo JGB; DO AMARAL, Wagner C.; FAVIER, Gérard. A note on the optimal expansion of Volterra models using Laguerre functions. *Automatica*, v. 42, n. 4, p. 689-693, 2006.
- CANEVAROLO JR, Sebastião V. *Ciência dos polímeros*. Artiber editora, São Paulo, v. 24, 2002.
- CHENG, C. M. et al. Volterra-series-based nonlinear system modeling and its engineering applications: A state-of-the-art review. *Mechanical Systems and Signal Processing*, v. 87, p. 340-364, 2017.
- DA ROSA, Alex; CAMPELLO, Ricardo JGB; AMARAL, Wagner C. Exact search directions for optimization of linear and nonlinear models based on generalized orthonormal functions. *IEEE Transactions on Automatic Control*, v. 54, n. 12, p. 2757-2772, 2009.
- DA SILVA, Gustavo Vitorino Monteiro. *Controlo não linear*. Escola superior de tecnologia de Setúbal, 2006.
- DOYLE III, Francis J.; OGUNNAIKE, Babatunde A.; PEARSON, Ronald K. Nonlinear model-based control using second-order Volterra models. *Automatica*, v. 31, n. 5, p. 697-714, 1995.
- DOYLE, Francis J.; PEARSON, Ronald K.; OGUNNAIKE, Babatunde Ayodeji. *Identification and control using Volterra models*. London: Springer, 2002.
- GRUBER, J. K. et al. Nonlinear MPC based on a Volterra series model for greenhouse temperature control using natural ventilation. *Control Engineering Practice*, v. 19, n. 4, p. 354-366, 2011.
- GUIMARÃES, Cícero Januário et al. *Sobre os espaços de Lebesgue e Sobolev generalizados e aplicações envolvendo op (x)-laplaciano*. 2006.
- José Goulart (2024). Fast Volterra Filtering (<https://www.mathworks.com/matlabcentral/fileexchange/32248-fast-volterra-filtering>), MATLAB Central File Exchange. Retrieved September 3, 2024.
- JUNIOR, Wilson Arnaldo Artuzi; CRUZ, Ricardo Rodrigo Wolf. *Introdução à Transformada Z*. Universidade Federal do Paraná, Paraná, Relatório, 2010.
- KUMAR, Ritwik et al. Trainable convolution filters and their application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, v. 34, n. 7, p. 1423-1436, 2011.

LE CAILLEC, J.-M.; GARELLO, René; CHAPRON, Bertrand. Analysis of the SAR imaging process of the ocean surface using Volterra models. *IEEE journal of oceanic engineering*, v. 27, n. 3, p. 675-699, 2002.

OLIVEIRA, Gustavo HC et al. 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, n. 1-2, p. 121-132, 2011.

OLIVEIRA, Gustavo HC et al. An introduction to models based on Laguerre, Kautz and other related orthonormal functions—Part II: non-linear models. *International Journal of Modelling, Identification and Control*, v. 16, n. 1, p. 1-14, 2012.

RUGH, Wilson John. *Nonlinear system theory*. Baltimore: Johns Hopkins University Press, 1981.

SCHETZEN, Martin. Nonlinear system modelling and analysis from the Volterra and Wiener perspective. In: *Block-oriented Nonlinear System Identification*. London: Springer London, 2010. p. 13-24.

SILVA, A.O.; DA ROSA, A. Identificação paramétrica de sistemas usando modelos de Volterra com bases de funções ortonormais generalizadas. XXV Congresso Brasileiro de Automática, Rio de Janeiro-RJ, Brasil (a ser publicado em 2024).

SILVA, T. Oliveira E. Optimality conditions for truncated Laguerre networks. *IEEE Transactions on Signal Processing*, v. 42, n. 9, p. 2528-2530, 1994.

VAN DEN HOF, Paul et al. Modelling and identification with rational orthogonal basis functions. *IFAC Proceedings Volumes*, v. 33, n. 15, p. 445-455, 2000.

VAN DRONGELEN, Wim. *Signal processing for neuroscientists, a companion volume: advanced topics, nonlinear techniques and multi-channel analysis*. Elsevier, 2010.

VIANA, Gabriel Santos. Representação de sistemas não-lineares invariantes no tempo através do modelo de Volterra de segunda ordem com base de funções ortonormais. 2021.

YOON, Yongsoon; SUN, Zongxuan. Nonlinear identification and robust tracking control of a camless engine valve actuator based on a Volterra series representation. In: *2014 American Control Conference*. IEEE, 2014. p. 1535-1540.

ZHANG, Hai-Tao; TISCHENKO, Larissa; YU, Pin-Ze. A novel adaptive control algorithm based on non-linear Laguerre—Volterra observer. *Transactions of the Institute of Measurement and Control*, v. 31, n. 2, p. 129-151, 2009.

APÊNDICE A – CÓDIGOS

Todos os códigos que estão anexados nesse apêndice são de autoria do autor e foram feitos na plataforma do OCTAVE.

A.1 Representação do sistema utilizando o Modelo em Espaço de Estados

```
% Condições iniciais
```

```
x0 = [0;0;0;0];
```

```
% Tempo de Simulação
```

```
k = 0:0.03:32;
```

```
% Entrada do sistema
```

```
num_aleatorios = rand(1, length(k));
```

```
x = -0.6 + 1.4* num_aleatorios;
```

```
% Definição da função que retorna o vetor de derivadas das variáveis de estado do sistema
```

```
function dxdk = sistema_nao_linear(ksim, xsim, x)
```

```
    if nargin < 2
```

```
        xsim = zeros(4, 1);
```

```
    end
```

```
    dxdk = zeros(4, 1);
```

```
    dxdk(1) = -10.8957*xsim(1) - 0.447837*xsim(2) - 0.447837*xsim(1)*xsim(2)
```

```
    +0.111959 * xsim(2)^2;
```

```
    dxdk(2) = -10.1022*xsim(2) + 10.1022*x;
```

```
    dxdk(3) = 2.4162*xsim(1) + 8.7744*xsim(2) - 10*xsim(3) + 1.22581* xsim(1)*xsim(2) -
```

```
    0.306453*xsim(2)^2;
```

```
    dxdk(4) = 10*xsim(1) + 5.00001*xsim(2) - 10*xsim(4) + 5.00001*xsim(1)*xsim(2) -
```

```
    1.25* xsim(2)^2;
```

```
end
```

```
% Simulação usando ode45
```

```
[ksim, xsim] = ode45(@(ksim, xsim) sistema_nao_linear(ksim, xsim, interp1(k, x, ksim, 'previous',  
0)), k, x0);
```

% Saída do sistema

y_real = -xsim(:, 3) + xsim(:, 4) - xsim(:, 3) .* xsim(:, 4) + xsim(:, 3).^2;

A.2 Geração das funções de Laguerre e suas respectivas versões filtradas

% p pode assumir qualquer valor no intervalo]0,1[inicialmente

p = 0.09;

% Definição das funções de Laguerre

Laguerre_1 = ((1-p^2)^(1/2))*(p^k);

Laguerre_2 = ((1-p^2)^(1/2)).*((k+1).*(p.^(k+1))-k.*(p.^(k-1)));

Laguerre_3 = ((1-p^2)^(1/2)).*(p.^(k+2).*((k.^2+3.*k+2)/2)- p^k.*(k.^2+k)+ p.^(k-2).*(k.*(k-1)/2));

Laguerre_4 = ((1-p^2)^(1/2)).*(-p.^(k+3).*((k.^3+6.*k.^2+11.*k+6)/6)+

p.^(k+1).*((k.^3+3.*k.^2+2.*k)/2)+p.^(k-1).*(-k.^3+k)/2+ p.^(k-3).*(k.^3-3.*k.^2+2.*k)/6);

Laguerre_5 = ((1-p^2)^(1/2)).*1/24.*p.^(k-4).*((k+1).*(k+2).*(k+3).*(k+4).*p.^8 - 4.*k.*(k+1).*(k+2).*(k+3).*p.^6 + 6.*(k-1).*k.*(k+1).*(k+2).*p.^4 - 4.*(k-2).*(k-1).*k.*(k+1).*p.^2 + (k-3).*(k-2).*(k-1).*k);

Laguerre_6 = ((1 - p.^2)^(1/2)).*1/120.*p.^(k - 5).*(10.*k.*(k.^4 - 5.*k.^2 + 4).*p.^4 - (k + 1).*(k + 2).*(k + 3).*(k + 4).*(k + 5).*p.^10 + 5.*k.*(k + 1).*(k + 2).*(k + 3).*(k + 4).*p.^8 - 10.*(k - 1).*k.*(k + 1).*(k + 2).*(k + 3).*p.^6 - 5.*(k - 3).*(k - 2).*(k - 1).*k.*(k + 1).*p.^2 + (k - 4).*(k - 3).*(k - 2).*(k - 1).*k);

Laguerres = [Laguerre_1', Laguerre_2', Laguerre_3', Laguerre_4', Laguerre_5', Laguerre_6'];

% Funções de Laguerre Filtradas pela entrada x

L1 = zeros(size(k));

L2 = zeros(size(k));

L3 = zeros(size(k));

L4 = zeros(size(k));

L5 = zeros(size(k));

L6 = zeros(size(k));

for i = 1:length(k)

for tau = 0:i-1

L1(i) = L1(i) + Laguerre_1(tau+1)*x(i - tau);

L2(i) = L2(i) + Laguerre_2(tau+1)*x(i - tau);

L3(i) = L3(i) + Laguerre_3(tau+1)*x(i - tau);

```

    L4(i) = L4(i) + Laguerre_4(tau+1)*x(i - tau);
    L5(i) = L5(i) + Laguerre_5(tau+1)*x(i - tau);
    L6(i) = L6(i) + Laguerre_6(tau+1)*x(i - tau);
end
end

```

```

Laguerres_f = [L1', L2', L3', L4', L5', L6'];
M = size(Laguerres_f, 2);
N = length(k);

```

A.3 Construção dos regressores e estimação dos coeficientes

% Regressores de primeira ordem

```
H1 = Laguerres_f;
```

% Regressores de segunda ordem

```

H2 = zeros(N, M^2);
count = 1;
for i = 1:M
    for j = 1:M
        H2(:, count) = Laguerres_f(:, i).*Laguerres_f(:, j);
        count = count + 1;
    end
end

```

% Regressores de terceira ordem

```

H3 = zeros(N, M^3);
count = 1;
for i = 1:M
    for j = 1:M
        for l = 1:M
            H3(:, count) = Laguerres_f(:, i) .* Laguerres_f(:, j) .* Laguerres_f(:, l);
            count = count + 1;
        end
    end
end

```

```

    end
end

% Concatenação dos Regressores
H = [H1 H2 H3];

% Estimação utilizando o operador '\'
coef = zeros(size(H, 2), 1);
coef = H \ y_real;

% coeficientes para cada ordem
coef_1_ordem = coef(1:M);
coef_2_ordem = coef(M+1:M+M^2);
coef_3_ordem = coef(M+M^2+1:end);

```

A.4 Cálculo dos kernels

```

% kernel de primeira ordem
K1 = coef_1_ordem' * [Laguerre_1; Laguerre_2; Laguerre_3; Laguerre_4; Laguerre_5;
Laguerre_6];

% kernel de segunda ordem
K2 = zeros(length(k), length(k));
count = 1;
for i = 1:M
    for j = 1:M
        K2 = K2 + coef_2_ordem(count)*(Laguerres(:, i) * Laguerres(:, j)');
        count = count + 1;
    end
end
end

```

```

% kernel de terceira ordem
K3 = zeros(length(k), length(k), length(k));
count = 1;
for i = 1:M
    for j = 1:M
        for l = 1:M
            % Expansão tensorial
            Li = reshape(Laguerres(:, i), [length(k), 1, 1]);
            Lj = reshape(Laguerres(:, j), [1, length(k), 1]);
            Ll = reshape(Laguerres(:, l), [1, 1, length(k)]);
            K3 = K3 + coef_3_ordem(count) * (Li .* Lj .* Ll);
            count = count + 1;
        end
    end
end
end

```

APÊNDICE B – CÁLCULO DAS FUNÇÕES DE LAGUERRE

Serão calculadas as Z^{-1} das 6 primeiras funções de Laguerre e será necessário utilizar as seguintes propriedades da transformada Z: (adicionar referência)

$$\frac{z}{(z-a)} \xleftrightarrow{Z} a^k u(k) \quad (1)$$

$$\frac{z}{(z-a)^2} \xleftrightarrow{Z} k a^{k-1} u(k) \quad (2)$$

$$\frac{z}{(z-a)^3} \xleftrightarrow{Z} \frac{k(k-1)}{2} a^{k-2} u(k) \quad (3)$$

$$\frac{z}{(z-a)^4} \xleftrightarrow{Z} \frac{(k-2)(k-1)k}{6} a^{k-3} u(k) \quad (4)$$

$$\frac{z}{(z-a)^5} \xleftrightarrow{Z} \frac{(k-3)(k-2)(k-1)k}{24} a^{k-4} u(k) \quad (5)$$

$$\frac{z}{(z-a)^6} \xleftrightarrow{Z} \frac{(k-4)(k-3)(k-2)(k-1)k}{120} a^{k-5} u(k) \quad (6)$$

Uma vez que o sistema será avaliado somente para $k > 0$, então $u(k) = 1$. A seguir serão calculadas as Z^{-1} .

1) $\psi_2(z)$

$$\psi_1(z) = z \frac{\sqrt{1-p^2}}{z-p} \left(\frac{1-pz}{z-p} \right)^0 = z \frac{\sqrt{1-p^2}}{z-p} \quad (2.9)$$

Ao utilizar a propriedade (1) teremos:

$$\begin{aligned} Z^{-1}[\psi_1(z)] &= \sqrt{1-p^2} p^k \\ Z^{-1}[\psi_1(z)] &= \psi_1(k) \end{aligned} \quad (2.10)$$

Onde $Z^{-1}[\psi_1(z)]$ simboliza a transformada inversa de $\psi_1(z)$.

2) $\psi_2(z)$

$$\begin{aligned}\psi_2(z) &= z \frac{\sqrt{1-p^2}}{z-p} \left(\frac{1-pz}{z-p} \right) = z \frac{\sqrt{1-p^2} (1-pz)}{(z-p)^2} \\ \psi_2'(z) &= \frac{\psi_2(z)}{\sqrt{1-p^2}} = z \frac{(1-pz)}{(z-p)^2}\end{aligned}\tag{2.11}$$

Expandindo em frações parciais:

$$\begin{aligned}\psi_2'(z) &= z \frac{(1-pz)}{(z-p)^2} = z \left(\frac{A}{z-p} \right) + z \left(\frac{B}{(z-p)^2} \right) = z \left(\frac{A(z-p)+B}{(z-p)^2} \right) \\ \psi_2'(z) &= z \left(\frac{A(z-p)+B}{(z-p)^2} \right)\end{aligned}\tag{2.12}$$

Comparando (2.11) e (2.12) temos que:

$$\begin{aligned}\begin{cases} A = -p \\ B = 1 - p^2 \end{cases} \\ \psi_2'(z) = -p \left(\frac{z}{z-p} \right) + (1 - p^2) \left(\frac{z}{(z-p)^2} \right)\end{aligned}$$

Utilizando as propriedades (1) e (2), teremos:

$$Z^{-1}[\psi_2'(z)] = -p(p^k) + (1 - p^2)(kp^{k-1}) = -((k+1)p^{k+1} - kp^{k-1})$$

$$Z^{-1}[\psi_2(z)] = -\sqrt{1-p^2}((k+1)p^{k+1} - kp^{k-1})\tag{2.13}$$

3) $\psi_3(z)$

$$\begin{aligned}\psi_3(z) &= z \frac{\sqrt{1-p^2}}{z-p} \left(\frac{1-pz}{z-p} \right)^2 = z \frac{\sqrt{1-p^2} (1-pz)^2}{(z-p)^3} \\ \psi_3'(z) &= z \frac{(1-pz)^2}{(z-p)^3} = z \left(\frac{A}{z-p} \right) + z \left(\frac{B}{(z-p)^2} \right) + z \left(\frac{C}{(z-p)^3} \right) \\ \psi_3'(z) &= z \left(\frac{A(z-p)^2 + B(z-p) + C}{(z-p)^3} \right)\end{aligned}$$

$$\begin{cases} A = p^2 \\ B = 2p^3 - 2p \\ C = p^4 - 2p^2 + 1 \end{cases}$$

$$\psi'_3(z) = p^2 \left(\frac{z}{z-p} \right) + (2p^3 - 2p) \left(\frac{z}{(z-p)^2} \right) + (p^4 - 2p^2 + 1) \left(\frac{z}{(z-p)^3} \right)$$

Ao utilizar as propriedades (1), (2) e (3):

$$Z^{-1}[\psi'_3(z)] = p^2(p^k) + (2p^3 - 2p)(kp^{k-1}) + (p^4 - 2p^2 + 1) \left(\frac{k(k-1)p^{k-2}}{2} \right)$$

$$Z^{-1}[\psi_3(z)] = \sqrt{1-p^2} \left(p^{k+2} \left(\frac{k^2 + 3k + 2}{2} \right) - p^k(k^2 + k) + p^{k-2} \left(\frac{k(k-1)}{2} \right) \right) \quad (2.14)$$

4) $\psi_4(z)$

$$\begin{aligned} \psi_4(z) &= z \frac{\sqrt{1-p^2}}{z-p} \left(\frac{1-pz}{z-p} \right)^3 = z \frac{\sqrt{1-p^2} (1-pz)^3}{(z-p)^4} \\ \psi'_4(z) &= z \frac{(1-pz)^3}{(z-p)^4} = z \left(\frac{A}{z-p} \right) + z \left(\frac{B}{(z-p)^2} \right) + z \left(\frac{C}{(z-p)^3} \right) + z \left(\frac{D}{(z-p)^4} \right) \\ \psi'_4(z) &= z \left(\frac{A(z-p)^3 B(z-p)^2 + C(z-p) + D}{(z-p)^4} \right) \end{aligned}$$

$$\begin{cases} A = -p^3 \\ B = 3p^2(1-p^2) \\ C = 3p(-p^4 + 2p^2 - 1) \\ D = -p^6 + 3p^4 - 3p^2 + 1 \end{cases}$$

$$\begin{aligned} \psi'_4(z) &= -p^3 \left(\frac{z}{z-p} \right) + 3p^2(1-p^2) \left(\frac{z}{(z-p)^2} \right) + 3p(-p^4 + 2p^2 - 1) \left(\frac{z}{(z-p)^3} \right) + \\ &+ (-p^6 + 3p^4 - 3p^2 + 1) \left(\frac{z}{(z-p)^4} \right) \end{aligned}$$

Ao utilizar as propriedades (1), (2), (3) e (4):

$$\begin{aligned} Z^{-1}[\psi'_4(z)] &= -p^3(p^k) + 3p^2(1-p^2)(kp^{k-1}) + 3p(-p^4 + 2p^2 - 1) \left(\frac{k(k-1)}{2} p^{k-2} \right) + \\ &+ (-p^6 + 3p^4 - 3p^2 + 1) \left(\frac{(k-2)(k-1)k}{6} p^{k-3} \right) \end{aligned}$$

$$Z^{-1}[\psi_4(z)] = \sqrt{1-p^2} \left(\begin{aligned} & -p^{k+3} \left(\frac{k^3 + 6k^2 + 11k + 6}{6} \right) + p^{k+1} \left(\frac{k^3 + 3k^2 + 2k}{2} \right) + \\ & + p^{k-1} \left(\frac{-k^3 + k}{2} \right) + p^{k-3} \left(\frac{k^3 - 3k^2 + 2k}{6} \right) \end{aligned} \right) \quad (2.15)$$

5) $\psi_5(z)$

$$\begin{aligned} \psi_5(z) &= z \frac{\sqrt{1-p^2}}{z-p} \left(\frac{1-pz}{z-p} \right)^4 = z \frac{\sqrt{1-p^2} (1-pz)^4}{(z-p)^5} \\ \psi'_5(z) &= z \frac{(1-pz)^4}{(z-p)^5} = z \left(\frac{A}{z-p} \right) + z \left(\frac{B}{(z-p)^2} \right) + z \left(\frac{C}{(z-p)^3} \right) + z \left(\frac{D}{(z-p)^4} \right) + z \left(\frac{E}{(z-p)^5} \right) \\ \psi'_5(z) &= z \left(\frac{A(z-p)^4 + B(z-p)^3 + C(z-p)^2 + D(z-p) + E}{(z-p)^5} \right) \end{aligned}$$

$$\begin{cases} A = p^4 \\ B = 4p^3(p^2 - 1) \\ C = 6p^2(p^4 - 2p^2 + 1) \\ D = 4p(p^6 - 3p^4 + 3p^2 - 1) \\ E = p^8 - 4p^6 + 6p^4 - 4p^2 + 1 \end{cases}$$

$$\begin{aligned} \psi'_5(z) &= p^4 \left(\frac{z}{z-p} \right) + 4p^3(p^2 - 1) \left(\frac{z}{(z-p)^2} \right) + 6p^2(p^4 - 2p^2 + 1) \left(\frac{z}{(z-p)^3} \right) + \\ &+ 4p(p^6 - 3p^4 + 3p^2 - 1) \left(\frac{z}{(z-p)^4} \right) + (p^8 - 4p^6 + 6p^4 - 4p^2 + 1) \left(\frac{z}{(z-p)^5} \right) \end{aligned}$$

Ao utilizar as propriedades (1), (2), (3), (4) e (5):

$$\begin{aligned} Z^{-1}[\psi'_5(z)] &= p^4(p^k) + 4p^3(p^2 - 1)(kp^{k-1}) + 6p^2(p^4 - 2p^2 + 1) \left(\frac{k(k-1)}{2} p^{k-2} \right) + \\ &+ 4p(p^6 - 3p^4 + 3p^2 - 1) \left(\frac{(k-2)(k-1)k}{6} p^{k-3} \right) + \\ &+ (p^8 - 4p^6 + 6p^4 - 4p^2 + 1) \left(\frac{(k-3)(k-2)(k-1)k}{24} p^{k-4} \right) \end{aligned}$$

Expandindo os termos:

$$\begin{aligned}
 Z^{-1}[\psi_5(z)] &= \sqrt{1-p^2} \left(p^{k+4} \left(\frac{k^4+10k^3+35k^2+50k+24}{24} \right) - p^{k+2} \left(\frac{k^4+6k^3+11k^2+6k}{6} \right) + \right. \\
 &\quad \left. + \left(\frac{k^4+2k^3-k^2-2k}{4} \right) + p^{k-2} \left(\frac{-k^4+2k^3+k^2-2k}{6} \right) + p^{k-4} \left(\frac{k^4-6k^3+11k^2-6k}{24} \right) \right)
 \end{aligned} \tag{2.16}$$

6) $\psi_6(z)$

$$\begin{aligned}
 \psi_6(z) &= z \frac{\sqrt{1-p^2}}{z-p} \left(\frac{1-pz}{z-p} \right)^5 = z \frac{\sqrt{1-p^2} (1-pz)^5}{(z-p)^6} \\
 \psi'_6(z) &= z \frac{(1-pz)^5}{(z-p)^6} = z \left(\frac{A}{z-p} \right) + z \left(\frac{B}{(z-p)^2} \right) + z \left(\frac{C}{(z-p)^3} \right) + z \left(\frac{D}{(z-p)^4} \right) + z \left(\frac{E}{(z-p)^5} \right) + \\
 &\quad z \left(\frac{F}{(z-p)^6} \right) \\
 \psi'_6(z) &= z \left(\frac{A(z-p)^5 + B(z-p)^4 + C(z-p)^3 + D(z-p)^2 + E(z-p) + F}{(z-p)^6} \right)
 \end{aligned}$$

$$\begin{cases}
 A = -p^5 \\
 B = 5p^4(1-p^2) \\
 C = 10p^3(-p^4+2p^2-1) \\
 D = 10p^2(1-3p^2+3p^4-p^6) \\
 E = -5p(1-4p^2+6p^4-4p^6+p^8) \\
 F = -p^{10}+5p^8-10p^6+10p^4-5p^2+1
 \end{cases}$$

$$\begin{aligned}
 \psi'_6(z) &= -p^5 \left(\frac{z}{z-p} \right) + 5p^4(1-p^2) \left(\frac{z}{(z-p)^2} \right) + 10p^3(-p^4+2p^2-1) \left(\frac{z}{(z-p)^3} \right) + \\
 &\quad + 10p^2(1-3p^2+3p^4-p^6) \left(\frac{z}{(z-p)^4} \right) - 5p(1-4p^2+6p^4-4p^6+p^8) \left(\frac{z}{(z-p)^5} \right) + (-p^{10}+5p^8- \\
 &\quad -10p^6+10p^4-5p^2+1) \left(\frac{z}{(z-p)^6} \right)
 \end{aligned}$$

Ao utilizar as propriedades (1), (2), (3), (4), (5) e (6):

$$\begin{aligned}
 Z^{-1}[\psi'_6(z)] &= -p^5(p^k) + 5p^4(1-p^2)(kp^{k-1}) + \\
 &\quad + 10p^3(-p^4+2p^2-1) \left(\frac{k(k-1)}{2} p^{k-2} \right) + \\
 &\quad + 10p^2(1-3p^2+3p^4-p^6) \left(\frac{(k-2)(k-1)k}{6} p^{k-3} \right) + \\
 &\quad - 5p(1-4p^2+6p^4-4p^6+p^8) \left(\frac{(k-3)(k-2)(k-1)k}{24} p^{k-4} \right) +
 \end{aligned}$$

$$+(-p^{10} + 5p^8 - 10p^6 + 10p^4 - 5p^2 + 1) \left(\frac{(k-4)(k-3)(k-2)(k-1)k}{120} p^{k-5} \right)$$

Expandindo os termos:

$$\begin{aligned} Z^{-1}[\psi_6(z)] = & \sqrt{1-p^2} \left(-p^{k+5} \left(\frac{k^5+15k^4+85k^3+225k^2+274k+120}{120} \right) + \right. \\ & + p^{k+3} \left(\frac{k^5+10k^4+35k^3+50k^2+24k}{24} \right) - p^{k+1} \left(\frac{k^5+5k^4+5k^3-5k^2-6k}{12} \right) + p^{k-1} \left(\frac{k^5-5k^3+4k}{12} \right) + \\ & \left. + p^{k-3} \left(\frac{-k^5+5k^4+5k^3-5k^2+6k}{24} \right) + p^{k-5} \left(\frac{k^5+10k^4+35k^3+50k^2+24k}{120} \right) \right) \end{aligned}$$

(2.17)

Excel, PowerPoint e PDF. Obter o aplicativo Microsoft 365:
<https://aka.ms/GetM365>