



**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS**  
**UNIDADE ARAXÁ**

**RANGEL RESENDE ALVES**

**DESENVOLVIMENTO DE SOFTWARE DE IDENTIFICAÇÃO DE  
SISTEMAS NÃO LINEARES UTILIZANDO ALGORITMO DE  
GOLUB-HOUSEHOLDER**

**ARAXÁ/MG**

**2016**

**RANGEL RESENDE ALVES**

**DESENVOLVIMENTO DE SOFTWARE DE IDENTIFICAÇÃO DE  
SISTEMAS NÃO LINEARES UTILIZANDO ALGORITMO DE  
GOLUB-HOUSEHOLDER**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Automação Industrial, do Centro Federal de Educação Tecnológica de Minas Gerais - CEFET/MG, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Automação Industrial.

Orientador: Prof. Me. Luís Paulo Fagundes

**ARAXÁ/MG**

**2016**



Serviço Público Federal  
MINISTÉRIO DA EDUCAÇÃO  
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS  
COORDENAÇÃO DO CURSO DE ENGENHARIA DE AUTOMAÇÃO INDUSTRIAL/ ARAXÁ

TRABALHO DE CONCLUSÃO DE CURSO – TCC – ATA DE DEFESA


ATA DA DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO DE ENGENHARIA DE AUTOMAÇÃO INDUSTRIAL do aluno *Rangel Resende Alves*

Às 18h do dia 05 de agosto de 2016, reuniu-se, no Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG/ Unidade Araxá, a Comissão Examinadora de Trabalho de Conclusão de Curso para julgar, em exame final, o trabalho intitulado **Desenvolvimento de Software de Identificação de Sistemas Não Lineares Utilizando Algoritmo de Golub-Householder** como parte dos requisitos para a obtenção do Título de Engenheiro de Automação Industrial. Abrindo a sessão, o Presidente da Comissão, Prof. Me. Luis Paulo Fagundes, após dar a conhecer aos presentes o teor das Normas Regulamentares do Trabalho Final, passou a palavra ao candidato para apresentação de seu trabalho. Seguiu-se a arguição pelos examinadores, com a respectiva defesa do candidato. Logo após, a Comissão se reuniu, sem a presença do candidato e do público, para julgamento e expedição do resultado final.

Após a reunião da Banca Avaliadora, o candidato foi considerado: Aprovado Com nota final de: 82,33 / 100.

O resultado final foi comunicado publicamente ao candidato pelo Presidente da Comissão. O aluno abaixo-assinado declara que o trabalho ora identificado é da sua autoria material e intelectual, excetuando-se eventuais elementos, tais como passagens de texto, citações, figuras e datas, desde que as mesmas identifiquem claramente a fonte original, explicitando as autorizações obtidas dos respectivos autores, quando necessárias. Declara ainda, neste âmbito, não estar a violar direitos de terceiros.


*Rangel Resende Alves*  
Nome do aluno

  
\_\_\_\_\_ *assinatura*

Araxá, 05 de agosto de 2016.  
*local e data*

Nada mais havendo a tratar, o Presidente encerrou os trabalhos e lavrou a presente ATA, que será assinada por todos os membros participantes da Comissão Examinadora.

Araxá, 05 de Agosto de 2016.

  
\_\_\_\_\_ Prof. Me. Luis Paulo Fagundes (Orientador)

  
\_\_\_\_\_ Prof. Dr. Henrique José Avelar

  
\_\_\_\_\_ Prof. Me. Frederico Duarte Fagundes

**DEDICO ESTE TRABALHO**

*Aos meus pais, a minha esposa, a meus filhos e a meu irmão,  
que sempre me incentivaram e me fizeram acreditar na realização dos meus sonhos e  
souberam ter paciência em certos momentos durante esta trajetória árdua, mas que  
está chegando a um fim, ou melhor dizendo a novos começos.*

## **AGRADECIMENTOS**

A Deus pela vida e por iluminar os meus caminhos com a presença de pessoas tão especiais. Por me dar forças para vencer os momentos difíceis; coragem para continuar superando as dificuldades e paciência para não me entregar ao desânimo diante das minhas fraquezas.

Aos meus pais Vilma e Geraldo e meu irmão Douglas pelos incentivos e encorajamento durante esse árduo caminho. Agradecer também à minha esposa Karla por me incentivar e muitas vezes ter paciência, muitas das vezes para mim e para ela, e agradecer por aos meus filhos Pedro Henrique e João Victor, mesmo sendo tão novos e pequenos foram uma grande fonte de inspiração para desenvolver este trabalho e nunca desistir mesmo nos momentos difíceis

Aos meus professores de graduação, os quais de forma direta ou indireta me permitiram agregar conhecimentos e crescer tecnicamente quanto pessoalmente, devido à vivência que tive com todos.

Gostaria de agradecer em especial ao professor Luis Paulo Fagundes, o qual aceitou o desafio de orientar este trabalho e pelas valiosas orientações, e à professora Érica Daniela de Araújo, pela paciência e ajuda nas correções ortográficas

*"Cada sonho que você deixa pra trás, é um pedaço do seu futuro que deixa de existir."*

Steve Jobs

## RESUMO

A identificação de sistemas consiste na área do conhecimento que se destaca por buscar maneiras de desenvolver e implementar modelos matemáticos. Tornando-se essencial à engenharia, pois com a obtenção desses modelos pode-se entender e compreender o comportamento de sistemas reais e utilizá-los para solucionar problemas. O presente trabalho teve como objetivo o desenvolvimento de um software didático capaz de realizar a identificação de sistemas não lineares, utilizando algoritmo de Golub-Householder, juntamente com o modelo NARMAX. Durante o desenvolvimento será analisado e apresentado as etapas para a identificação de sistemas não lineares com ênfase na utilização do algoritmo de Golub-Householder. Baseando-se em rotinas computacionais, desenvolvidas no software Matlab, as quais serão ‘alimentadas’ com informações, sinais de entrada e saída de determinados sistemas dinâmicos, com a finalidade de identificar sistemas não lineares. O desenvolvimento deste trabalho possibilitou a obtenção de um software didático para identificação de sistemas não lineares o qual facilita o processo de identificação, realizando tarefas como de tratamento e processamento de dados, e que permita que os alunos apliquem os conhecimentos obtidos em teoria, na prática.

Palavras-chave: Identificação de sistemas. Sistemas não lineares. Algoritmo de Golub-Householder. Modelo NARMAX.

## **ABSTRACT**

Systems identification is a knowledge area that highlights to seek different ways to develop and implement mathematics models. Becoming essential to engineering, because through the obtainment of these models, it can understand real systems behaviour and use it to solve problems. This study has as objective the production of a didactic software able to perform non-linear systems identification, using Golub-Householder algorithm, along with NARMAX model. During the development, will be analyzed and shown the steps to do a non-linear systems identification with emphasis in use of the Golub-Householder algorithm. Based on computational routines and developed in Matlab® software, these routines will be supplied with information, infeed and outfeed signals from determined dynamic systems in order to identify non-linear systems. The development of this study made possible the obtaining of a non-linear systems identification didactic software that facilitates identification process, allowing students to put their theoretical knowledge into practice.

**Keywords:** Systems identification. Non-linear systems. Golub-Householder algorithm. NARMAX model.



## LISTA DE FIGURAS

|   |    |
|---|----|
| Figura 1: Exemplo modelagem caixa-branca .....  | 17 |
| Figura 2 : Exemplo modelagem caixa-preta.....   | 18 |
| Figura 3:Exemplo de modelagem caixa-cinza, sistema de operações em ambientes internos de helimodelismo..... | 19 |
| Figura 4: Quadro ilustrativo da evolução da identificação de sistemas. ....                                 | 20 |
| Figura 5 Representação etapas de identificação de sistemas.....   | 23 |
| Figura 6: Propriedade de ortogonalidade do estimador MQ .....   | 28 |
| Figura 7: Diagrama de blocos e modelo de forno a arco no ambiente Matlab.....                               | 34 |
| Figura 8: Diagrama de blocos da estrutura utilizado o forno elétrico. ....                                  | 35 |
| Figura 9: Representação do trocador de calor .....  | 36 |
| Figura 10 : Layout desenvolvido.....  | 38 |
| Figura 11: Trecho rotina implemetada para acionamento botões da interface:.....                             | 38 |
| Figura 12: Inicialização interface Gráfica com o Usuário GUI.....   | 39 |
| Figura 13: Tela de Inicialização GUIDE.....   | 40 |
| Figura 14: Interface Desenvolvida .....   | 41 |
| Figura 15: Configuração parâmetros teste 1 .....  | 49 |
| Figura 16: Gráfico Dados Utilizados Teste 1 .....   | 50 |
| Figura 17: Gráficos Resultados identificação 1.....   | 51 |
| Figura 18: Graficos Dados Teste 2 e Teste 3.....  | 52 |
| Figura 19: Gráfico resultados teste 2 np=11.....  | 53 |
| Figura 20: Graficos Resultado Teste 3 np = 3 .....  | 53 |

## LISTA DE ABREVIATURAS E SIGLAS

ARMAV – Modelo auto-regressivo, de média móvel

ARV – Modelo auto-regressivo vetorial

CEFET – Centro Federal de Educação Tecnológica de Minas Gerais – Campus Araxá

CGS – Método Clássico de Gram-Schmidt

ERR – Taxa de redução do erro ("Error reduction ratio").

GH – Golub-Householder

GUI – Interface gráfica do usuário ("Graphical User Interface")

MGS – Método Modificado de Gram-Schmidt

MQ – Método dos Mínimos Quadrados

MQO – Método dos Mínimos Quadrados Ortogonal

NARMAX – Modelo não-linear auto-regressivo, de média móvel e entrada externa  
("Non-linear auto-regressive, moving average with exogenous input  
model").

NARX – Modelo não-linear auto-regressivo com entrada externa ("Non-linear  
auto-regressive with exogenous input").

:

# SUMÁRIO

|   |                  |
|---|------------------|
| <b><u>1. INTRODUÇÃO .....</u></b>   | <b><u>12</u></b> |
| <b><u>2. FUNDAMENTAÇÃO TEÓRICA.....</u></b>                               | <b><u>16</u></b> |
| 2.1. Identificação de Sistemas .....                                      | 16               |
| 2.1.1 - Sistema Caixa Branca (fenomenológica): .....                      | 16               |
| 2.1.2 - Sistema caixa-preta.....  | 17               |
| 2.1.3 - Sistema caixa cinza .....   | 18               |
| 2.2 - Conceitos Básicos.....  | 21               |
| 2.2.1 -Linearidade.....   | 21               |
| 2.2.2 - Invariância no Tempo.....   | 21               |
| 2.2.3 - Concentração de Parâmetros.....                                   | 22               |
| <b><u>3. Etapas de Identificação de Sistemas .....</u></b>                | <b><u>23</u></b> |
| 3.1 - Testes Dinâmicos e Coleta de Dados .....                            | 24               |
| 3.2 - Escolha da Representação Matemática a ser usada .....               | 24               |
| 3.3 - Determinação da Estrutura do Modelo.....                            | 25               |
| 3.4 - Estimação de Parâmetros .....                                       | 26               |
| 3.4.1 - Método de Mínimos Quadrados .....                                 | 26               |
| 3.4.2 - Ortogonalidade.....   | 27               |
| 3.5 - Validação do Modelo .....   | 29               |
| <b><u>4. Identificação de Sistemas Não lineares: Algoritmos .....</u></b> | <b><u>30</u></b> |
| 4.1. O Método Golub-Householder .....                                     | 30               |
| 4.2 Taxa de Redução de Erro.....  | 31               |
| <b><u>5. EXEMPLOS E APLICAÇÕES .....</u></b>                              | <b><u>33</u></b> |
| <b><u>6. METODOLOGIA.....</u></b>   | <b><u>37</u></b> |
| <b><u>7. RESULTADOS .....</u></b>   | <b><u>49</u></b> |
| <b><u>8. CONCLUSÃO.....</u></b>   | <b><u>55</u></b> |
| <b><u>REFERÊNCIAS .....</u></b>   | <b><u>56</u></b> |
| <b><u>ANEXO A - ROTINA PARA GERAÇÃO DA MATRIZ GENÉRICA .....</u></b>      | <b><u>59</u></b> |

# 1. INTRODUÇÃO

O homem procura entender, explicar e representar sistemas físicos, físico-químicos, sociais e biológicos, entre outros, há muito tempo. Uma das maneiras de representar um sistema real para sua melhor compreensão é através de um modelo matemático. De posse desse modelo, pode-se entender o comportamento do sistema real e utilizá-lo na solução de problemas. Uma das aplicações desses modelos matemáticos é para prever o comportamento do sistema em um momento posterior ao instante de análise.

Em vista dessas considerações, na identificação de sistemas, existem várias formas de classificar as técnicas de modelagens, uma delas agrupa os métodos em três categorias, denominadas, respectivamente, modelagem caixa-branca, modelagem caixa-cinza e modelagem caixa-preta (AGUIRRE, 2007). Tratar-se-á de cada uma dessas categorias com vistas a circunstanciá-las, dado que este trabalho tem por objetivo desenvolver uma plataforma de identificação de sistemas não lineares utilizando o método de Golub-Householder (doravante GH). O projeto proposto aborda sistemas não lineares devido, na prática, os sistemas dinâmicos em sua maioria serem não lineares. Conforme Aguirre (2007), na modelagem caixa-branca, é necessário conhecer profundamente o sistema em estudo, assim como as leis da física e da química que regem o sistema a ser modelado; por esse motivo, esse tipo de modelagem também é conhecido como modelagem da natureza do processo ou modelagem conceitual. Na modelagem caixa-preta, por sua vez, também conhecida como modelagem empírica, tem por característica marcante que pouco ou nenhum conhecimento prévio do sistema é necessário; nesse modelo, apenas dados de entrada e de saída do sistema são usados durante a identificação. A modelagem caixa-cinza, por seu turno, busca combinar as vantagens dos procedimentos de identificação caixa-preta e caixa-branca; essa técnica baseia-se na aquisição dos dados do sistema, tanto os de entrada quanto os de saída, juntamente com algum outro tipo de informação complementar que é usado na identificação.

De fato, há razões mais fortes para, em dada aplicação, optar por modelos não lineares, como, por exemplo, o fato de que modelos não lineares produzem certos regimes dinâmicos que modelos lineares não conseguem representar (AGUIRRE, 2007).

Entretanto, a escolha por modelos não lineares traz consigo um inevitável aumento na complexidade dos algoritmos a serem utilizados (AGUIRRE, 2007).

O algoritmo de GH foi utilizado neste projeto com a finalidade de estimar os parâmetros do modelo, assim como determinar a sua estrutura. O método desse algoritmo foi desenvolvido no Matlab, ambiente de desenvolvimento projetado para simplificar a manipulação computacional que pode interpretar rotinas codificadas em uma linguagem de programação própria de alto-nível. Essa linguagem de programação foi estruturada e utilizada para implementar, internamente ao ambiente, uma série de ferramentas úteis no processo de sinais e na modelagem e controle de sistemas dinâmicos.

A identificação de sistemas tem apresentado aplicações em diversas áreas, tornando interessante aos alunos de Engenharia e de cursos técnicos vivenciar essa prática, assim como à instituição ensiná-la, dado as demandas de mercado. Vallverduet et al. (1992), por exemplo, apresenta a identificação do Controle Neural do Sistema Cardiovascular Humano utilizando modelo NARMAX. Outro exemplo é demonstrado por Alessandra Teodoro Neves (2006), em sua tese de doutorado, defendida em 2006, que consiste na aplicação de modelos paramétricos ARMAV e ARV na identificação modal de sistemas mecânicos.

Dado o exposto, neste trabalho, foi desenvolvido um sistema específico para aplicação de técnicas e de métodos visando determinar o modelo matemático do sistema que será aplicado no Centro Federal de Educação Tecnológica de Minas Gerais – Campus Araxá (doravante Cefet-MG), com vistas a fornecer aos alunos do curso Técnico em Eletrônica e do curso de graduação em Engenharia de Automação Industrial a oportunidade de vivenciar os desafios inerentes ao processo de identificação de sistemas, dado que, atualmente, o laboratório dessa instituição não apresenta nenhum software específico para a aplicação de identificação de sistemas.

Ademais, com o presente trabalho pretende-se desenvolver mais recursos didáticos, de baixo custo, para a instituição acima citada, visto que há demanda por formação de profissionais qualificados que saibam tratar adequadamente dados de sistema. Nessa medida, em seu processo de formação, o aluno terá vivência com situações específicas em laboratório, o que garantirá rápida tomada de decisão e, conseqüentemente, melhor interpretação e análise na identificação de sistemas. Além disso, a utilização de computadores e informática vem tornando-se cada vez mais indispensável para a vida das pessoas, tanto pessoalmente quanto profissionalmente, em virtude de uma maior agilidade no processo e no armazenamento de informações,

facilitando o desenvolvimento tarefas, otimizando tempo e número de pessoas envolvidas (RAMIRO, 2014). Nas instituições de ensino, conforme verificado por Ramiro (2014), a utilização de computadores tem a função de possibilitar a mediação no processo de ensino-aprendizagem, contribuindo com a construção do conhecimento do aluno, de tal modo a prepará-lo e capacitá-lo para o mercado de trabalho.

Com o avanço tecnológico vivido atualmente, nota-se que a informática está diretamente envolvida com diversas áreas do conhecimento, pois torna-se cada vez mais indispensável na vida das pessoas devido à praticidade e rapidez proporcionada no desenvolvimento de variadas atividades. Como notado em Ramiro (2014), a grande praticidade do computador vem o tornando uma ferramenta essencial para as empresas, principalmente, em áreas relacionadas a projetos, pois o aumento da demanda acompanha o desenvolvimento econômico e social das cidades. Na educação, o computador apresenta grande importância no desenvolvimento e na aprendizagem dos alunos, já que as ferramentas desenvolvidas com a evolução tecnológica preparam o aluno para um concorrido mercado de trabalho.

Diante do exposto, o presente trabalho tem como finalidade proporcionar mais uma ferramenta didática à instituição, a qual irá auxiliar a formação dos alunos de Engenharia de Automação Industrial. Além de apresentar uma maior motivação aos alunos no processo de aprendizagem, o *software* tornará as aulas menos monótonas, por propiciar uma relação mais direta do aluno com o cotidiano profissional. Logo, sendo o Cefet-MG uma instituição federal voltada para a formação técnico-profissional, proporcionar tamanho engajamento é fundamental. Assim, acredita-se que a aplicação proposta permitirá à instituição, além de diminuição nos gastos, a garantia de um recurso didático a mais que proporcione aos alunos, tanto do curso Técnico em Eletrônica quanto da Engenharia de Automação Industrial, vivenciar experiências e confrontar problemas que podem surgir na identificação de um sistema real.

O presente trabalho teve o propósito de desenvolver um *software* que permita empregar técnicas de identificação de sistemas não lineares utilizando algoritmo de Golub-Householder. Garantindo o tratamento adequado aos dados do sistema para que a identificação possa ser efetivada adequadamente. E criar interface de usuário para que testes diversos possam ser realizados no sistema em questão, permitindo a aplicação de técnicas de identificação de sistemas não lineares utilizando método de Golub-Householder.

Em vista das considerações anteriores, aventa-se a seguinte hipótese: se implementado rotinas de automatização do processo de identificação de sistemas não lineares, o mesmo seria executado em um tempo menor, permitindo um teste mais rápido em relação a diversos modelos representativos.

## **2. FUNDAMENTAÇÃO TEÓRICA**

### **2.1. Identificação de Sistemas**

A identificação de sistemas trata da construção de modelos matemáticos para representar os sistemas elaborados a partir de observações, usualmente, dos dados de entrada e saída. Várias formas e técnicas encontram-se disponíveis para a obtenção de modelos dos sistemas, os quais podem apresentar diferentes graus de formalização matemática (LJUNG, 1999; NELLES, 2000 e SOUSA, 2005).

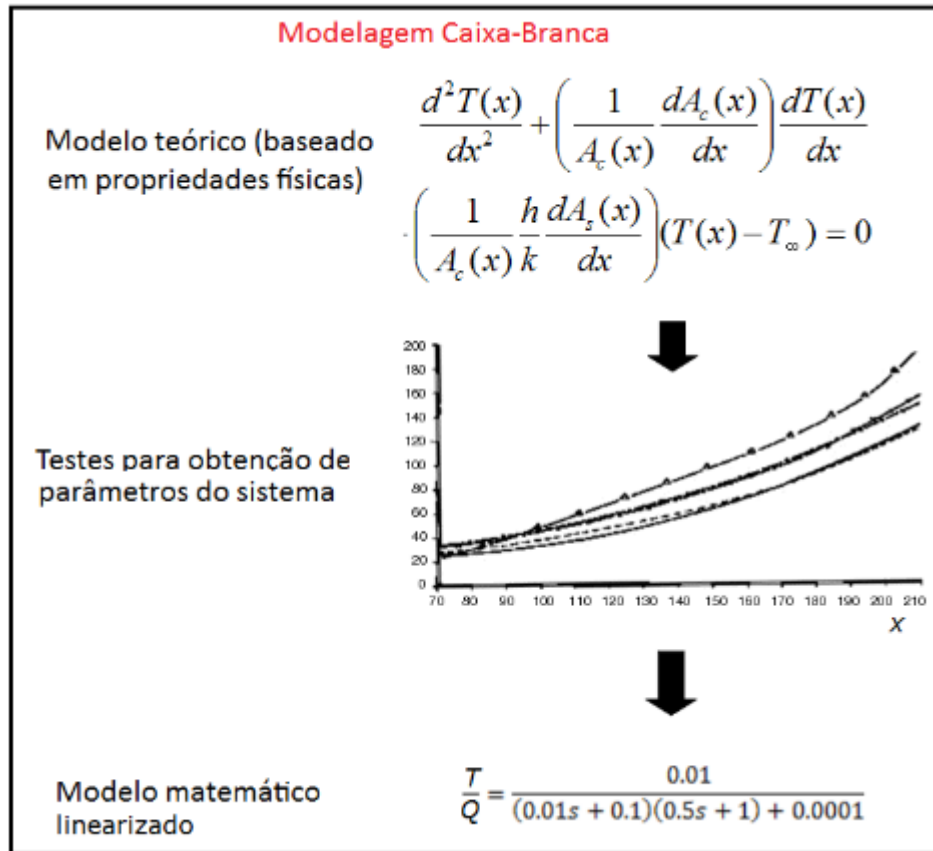
Conforme exposto, dentre as várias formas de classificar as técnicas de modelagem, podem-se agrupá-las em três grandes grupos denominados: modelagem caixa-branca, modelagem caixa-preta e modelagem caixa-cinza (AGUIRRE, 2007).

#### **2.1.1 - Sistema Caixa Branca (fenomenológica):**

Na modelagem caixa-branca, é necessário conhecer profundamente o sistema em estudo, assim como as leis da física e da química que regem o sistema a ser modelado, ou seja “suas estruturas são completamente ajustadas a partir de informações conhecidas "a priori". Nesse caso, a forma da função matemática que descreve o comportamento dinâmico do sistema original é pré-conhecida” (Sousa, 2005).



Figura 1: Exemplo modelagem caixa-branca



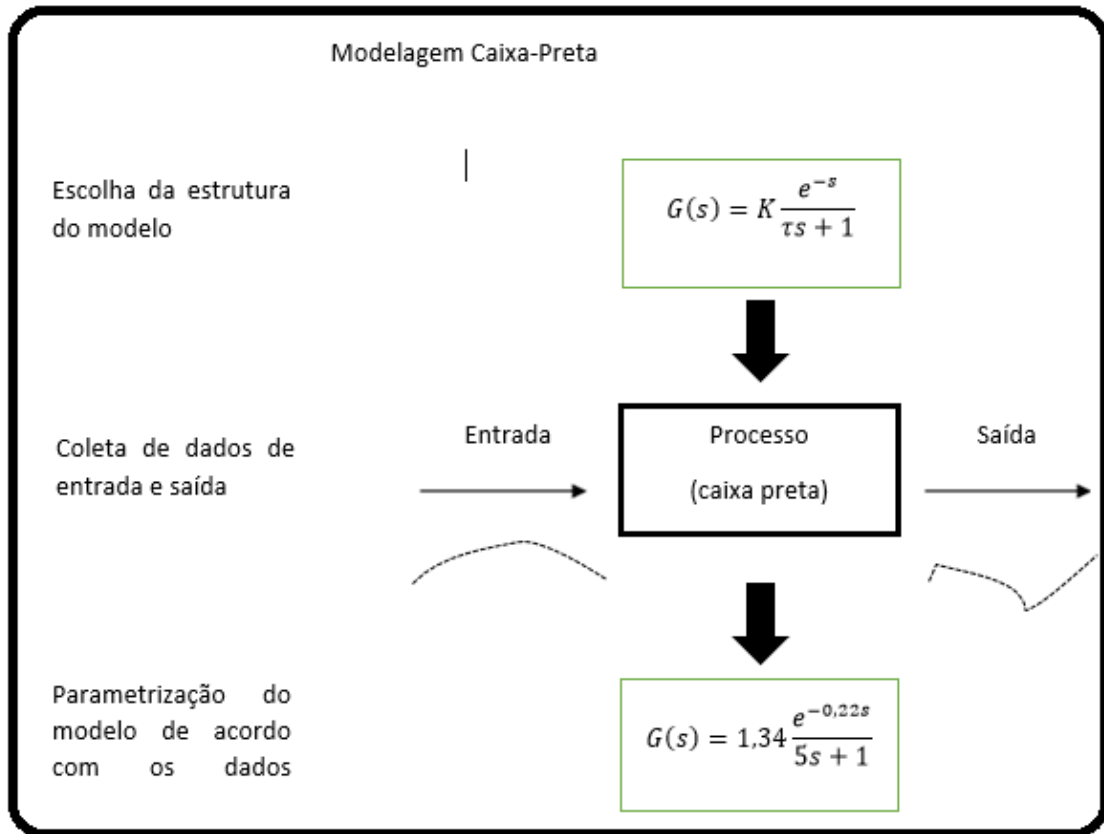
Fonte: Controle de Processos Industriais (2016).

### 2.1.2 - Sistema caixa-preta

A modelagem caixa-preta, por sua vez, também conhecida como modelagem empírica, tem por característica marcante que pouco ou nenhum conhecimento prévio do sistema é necessário. Nesse modelo, apenas dados de entrada e saída do sistema são usados durante a identificação.

Na figura 2 é representado esse método, que consiste na dispensa dos conhecimentos das leis da física ou da química, e baseia-se apenas nas relações de causa-efeito, correlacionando dados de entrada e saída do processo.

Figura 2 : Exemplo modelagem caixa-preta.



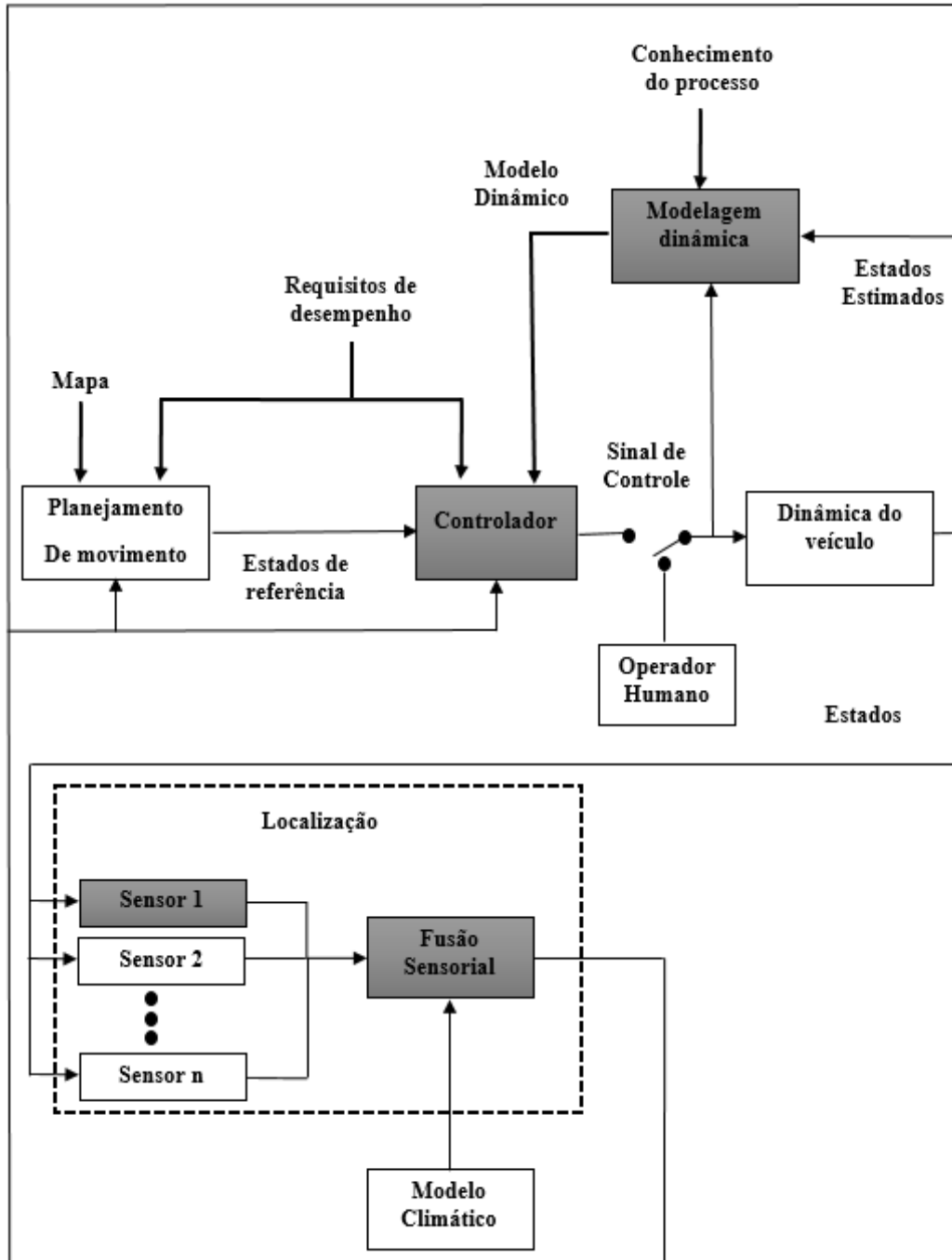
Fonte: Adaptado de Passos (2014).

### 2.1.3 - Sistema caixa cinza

A modelagem caixa-cinza, segundo Aguirre (2007), busca combinar as vantagens dos procedimentos de identificação caixa-preta e caixa-branca. Essa técnica baseia-se na aquisição dos dados do sistema, tanto os de entrada quanto os de saída, juntamente com algum outro tipo de informação complementar que é usado na identificação.

Na figura 3 verifica-se um sistema robótico para operações em ambientes internos a partir de uma plataforma comercial de helimodelo, considerando a tarefa de um voo autônomo. Onde o movimento da aeronave, pode ser tratada como um corpo rígido, descrito por 12 variáveis como: a posição, representada no referencial inercial, a velocidade do helicóptero, a posição angular e a velocidade angular, representada no referencial da aeronave, entre outras como os sinais de comando do piloto (de Moreira et al. (2011). Ou seja é utilizado um conhecimento a priori do sistema mas se faz necessário utilizar os sinais de entrada e saída para a determinação do modelo.

Figura 3: Exemplo de modelagem caixa-cinza, sistema de operações em ambientes internos de helimodelismo.

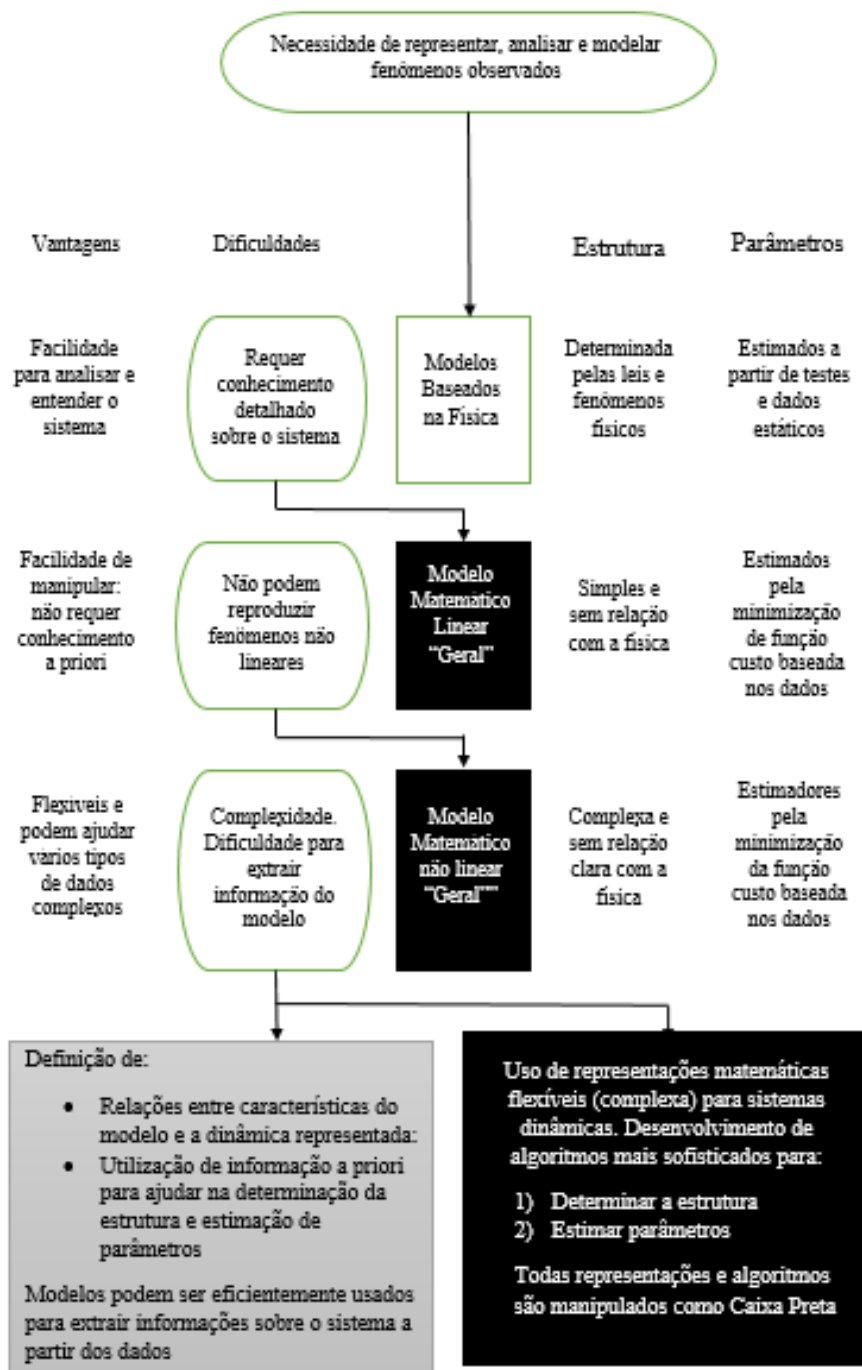


Fonte: Adaptado de Moreira et al. (2011).

Na sequência, baseando-se em Aguirre (2007), apresenta-se um quadro ilustrativo da evolução da identificação de sistemas, onde o quadro da figura 4 ilustra uma maneira

de entender a evolução da área nas últimas décadas. Nesse quadro, os blocos com quinas arredondadas referem-se a problemas cuja solução serviu de motivação para buscar novas ferramentas e técnicas. A linha de tempo é de cima para baixo sendo a parte inferior mais desenvolvida.

Figura 4: Quadro ilustrativo da evolução da identificação de sistemas.



Fonte: Adaptado de Aguirre (2007).

## **2.2 - Conceitos Básicos**

Um modelo matemático de um sistema real é um análogo matemático que representa algumas das características observadas em tal sistema (AGUIRRE, 2007). Sendo assim, o objetivo da identificação de sistemas é, portanto, a determinação de um modelo matemático capaz de representar as características de interesse do sistema, a partir de seus dados de entrada e saída (SOUSA, 2005). Logo, a fim de desenvolver modelos aproximados, algumas considerações devem ser feitas, tais como: linearidade, invariância no tempo e concentração de parâmetros.

### **2.2.1 -Linearidade**

Um sistema é considerado linear quando atende ao princípio da superposição. Esse princípio pode ser compreendido pelo seguinte exemplo: se um sistema excitado por uma entrada  $u_{1(t)}$  produz uma saída  $y_{1(t)}$  e produz uma saída  $y_{2(t)}$  quando excitado por uma entrada  $u_{2(t)}$ , o mesmo sistema, sendo excitado por uma combinação linear das entradas, a saber  $au_{1(t)} + bu_{2(t)}$  apresentará uma saída que corresponde à igual combinação linear das entradas individuais,  $ay_{1(t)} + by_{2(t)}$ , sendo  $a$  e  $b$  constantes reais.

Um sistema linear possui o mesmo tipo de comportamento independente do seu ponto de operação (AGUIRRE, 2007) .

### **2.2.2 - Invariância no Tempo**

Uma das premissas mais importantes da modelagem matemática é a invariância no tempo, pois isso significa que a dinâmica do modelo não se altera durante o período em que se analisa o sistema. Contudo, isso não quer dizer que o modelo estudado é estático, e sim que a dinâmica que está regulando a evolução temporal é a mesma, ou seja a saída do sistema varia com o tempo, mas os parâmetros são invariantes. Sobre esse aspecto, segundo Aguirre (2007), um sistema é invariante se um deslocamento no tempo de entrada provocar um deslocamento no tempo de saída, ou seja, a dinâmica do sistema não se altera com tempo. Aguirre (2007, p. 55) apresenta um exemplo de invariância interessante de ser lembrado, qual seja:

*Seja o sistema  $y(t) = \text{sen}(u(t))$ . Portanto, para uma entrada  $u_1(t)$  a saída será  $y_1 = \text{sen}(u_1(t))$ . Supondo que uma segunda entrada seja gerada como sendo o deslocamento temporal da primeira, ou seja,  $u_2 = u_1(t - t_0)$ .*

*A saída nesse caso é*

$$y_2 = \text{sen}(u_2(t)) = \text{sen}(u_1(t - t_0))$$

*Da definição do sistema, tem-se também*

$$y_1(t - t_0) = \text{sen}(u_1(t - t_0)),$$

*e como  $y_2(t) = y_1(t - t_0)$  o sistema é invariante*

### **2.2.3 - Concentração de Parâmetros**

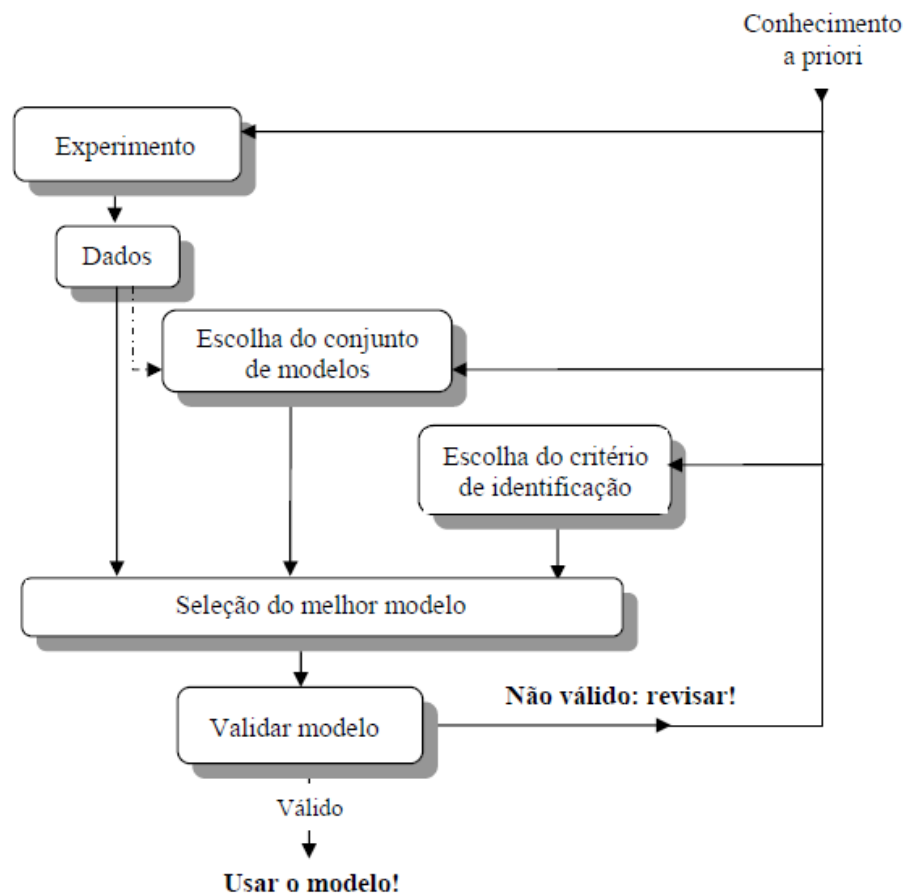
Esse tipo de representação pressupõe que as variáveis de interesse variam apenas com o tempo e não no espaço. "Em outras palavras, modelos com parâmetros concentrados descrevem o comportamento do sistema num único ponto do espaço" (AGUIRRE, 2007).

### 3. Etapas de Identificação de Sistemas

O processo de identificação de um sistema pressupõe, de uma maneira geral, a coleta e o processamento dos sinais de entrada e saída, a escolha de um conjunto de modelos e a seleção, dentre esses, do modelo que melhor representa o sistema (LJUNG, 1999 apud SOUSA, 2005, p.21).

As etapas das técnicas de identificação podem ser resumidas nas seguintes : “Testes dinâmicos e coleta de dados; Escolha da representação matemática a ser usada; Determinação da estrutura do modelo; Estimação de parâmetros; e Validação do modelo” (SOUSA, 2005). A representação da figura 5 ilustra as principais etapas de identificação de sistemas:

Figura 5 Representação etapas de identificação de sistemas.



Fonte: Antunes (2007).

### 3.1 - Testes Dinâmicos e Coleta de Dados

Uma vez que a identificação se propõe a obter modelos a partir de dados, é necessário gerar tais dados. Os dados utilizados para identificar os sistemas dinâmicos são gerados a partir de medições da resposta do sistema, de acordo com o sinal de excitação. Logo, os dados devem conter a informação necessária sobre o sistema a ser modelado (AGUIRRE, 2007). Como verificado por Rodrigues (2000), nesta etapa do processo deve-se garantir que os sinais entrada, os quais irão excitar o sistema, sejam projetados para satisfazer uma série de propriedades a fim de garantir dados condizentes.

### 3.2 - Escolha da Representação Matemática a ser usada

Uma importante questão relacionada com representação matemática a ser usada na modelagem do sistema é a escolha da sua estrutura, pois a estrutura escolhida deverá representar o comportamento de um sistema dinâmico (Rodrigues 2000), principalmente em sistemas não lineares, devido à diversidade do sistema.

“A escolha do tipo de representação dependerá dos objetivos do modelo, das ferramentas disponíveis para sua obtenção e das informações disponíveis a priori sobre o sistema a ser identificado” (Sousa, 2005, p.27). Dentre as várias representações de sistemas não lineares, destacam-se: Série de Volterra; Modelos de Hammerstein e de Wiener, Algumas representações NARX e NARMAX; Modelos polinomiais contínuos; Funções radiais de base; Redes neurais artificiais; Neurônio Neo-Fuzzy, entre outros.

Devido o foco do projeto ser voltado para sistema não lineares, a representação matemática usada será a NARMAX. Como demonstrado por Sousa (2005, p.28) essa estrutura pode ser representada, genericamente, por uma função não linear  $F'(\cdot)$  como,

$$y(k) = F'[y(k-1), \dots, y(k-n), u(k-1), \dots, u(k-n), \dots, e(k-1), \dots, e(k-n)] \quad (5.1)$$



Sendo :

$y(k-1), \dots, y(k-n)$  : representação dos atrasos do sistema em relação aos sinais de saída

$u(k-1), \dots, u(k-n)$  : representação dos atrasos do sistema em relação aos sinais de entrada

$e(k-1), \dots, e(k-n)$  : representação dos atrasos do sistema em relação aos sinais de ruído

### 3.3 - Determinação da Estrutura do Modelo

De acordo com Sousa (2005, p.29),

*escolha da estrutura do modelo é a tarefa mais importante e ao mesmo tempo a mais difícil no processo de identificação de sistemas. É a estrutura do modelo a principal responsável pela representação das características dinâmicas e estáticas do sistema. Algum conhecimento prévio sobre o sistema e sensibilidade por parte do projetista devem ser combinados para se determinar as propriedades requeridas do modelo.*

Como mencionado por Sousa (2005, p.29) o problema da detecção da estrutura está na obtenção dos termos a serem utilizados no sistema pois, segundo Rodrigues (2000, p.14), o número de termos possíveis em modelos polinomiais cresce bastante com o aumento do grau de não linearidade e da ordem do modelo.

Embora o número de elementos, termos obtidos, seja grande, apenas alguns poucos destes são utilizados para detecção da estrutura, os selecionados podem também ser denominados como termos candidatos do modelo (RODRIGUES, 2000, p.14). A fim de escolher os termos mais adequadamente, são utilizadas várias técnicas dentre as quais podemos destacar: taxa de redução de erro (ERR), critério de informação Akaike, agrupamento de termos e “zeroing-and-refitting”.

Em modelos não lineares Aguirre (2004) verifica que a tarefa fica mais complexa, tornando-se aconselhável a combinações de critérios para obtenção de regressores mais significativos.

### 3.4 - Estimação de Parâmetros

A estimação dos parâmetros consiste na determinação do modelo que melhor aproxime o comportamento dinâmico da estrutura escolhida ao sistema. A aproximação é feita através de uma função de custo que minimize a diferença entre a saída do sistema e a saída prevista para o modelo (SOUSA, 2005).

Um dos métodos mais conhecidos e utilizado para a estimação dos parâmetros trata-se do estimador de mínimos quadrados, como mencionado por Gauss (1963) alguns aspectos básicos aplicados a este método são:

- Número de observações é estritamente necessário à determinação das grandezas desconhecidas
- Maior número de observações do que o mínimo são necessários, afim de reduzir os efeitos dos erros gerados pelas medições.
- O problema da modelagem está implícito nas observações do sistema verificado
- E os parâmetros estimados devem satisfazer as observações da forma mais exata possível.

#### 3.4.1 - Método de Mínimos Quadrados

Ao analisar a equação 5.2

$$\theta = X^{-1}y \quad (3.2)$$

Compreende-se que apenas uma única solução poderá atender todas as n restrições do sistema ao mesmo tempo, mas se analisar a equação normal.

$$\theta = [X^T X]^{-1} X^T y \quad (3.3)$$

Nota-se que possui infinitas soluções sobredeterminadas, mas mesmo possuindo infinitas soluções o objetivo será encontrar uma resposta que traga algum significado intuitivo (AGUIRRE , 2007, p. 224), a fim de obter isto, deve-se admitir conhecer o valor de parâmetro  $\hat{\theta}$ , contudo deve-se assumir que os valores observados contém erros  $\xi$ , logo a expressão pode ser dada por:

$$y = X^T \hat{\theta} + \xi \quad (3.4)$$

Sendo:

$X \in R^N$  (Vetor de regressor)

$\hat{\theta} \in R^N$  (Vetor de parâmetro)

$\xi \in R^N$  (Resíduo, erro de estimação da variável  $y$ )

Ao analisar-se a equação 3.4 a mesma pode ser reescrita como:

$$\xi(k) = y(k) - X^T \hat{\theta} = y(k) - \hat{y}(k) \quad (3.5)$$

Com isso, ao aplicar a técnica dos mínimos quadrados, admite-se a estimação de parâmetros através da minimização da função custo (SOUSA , 2005, p.35).

$$J = \frac{1}{2} \sum_{k=1}^N \xi^2(N) = \frac{1}{2} \xi^T \xi = \frac{1}{2} \|\xi\|^2 \quad (3.6)$$

Como mencionado anteriormente, há presença de ruídos nas medições e, para reduzir este problema, torna-se necessário um maior número de amostras em relação à quantidade de regressores. Segundo Sousa (2005, p.36) “com isso, a matriz de regressores  $X$  não será quadrada, o que impede a determinação do vetor de parâmetro 3.5. Mas após algumas interações matemáticas a solução é obtida através do método dos mínimos quadrados através da expressão:”

$$\hat{\theta} = (X^T X)^{-1} X^T y(k) \quad (3.7)$$

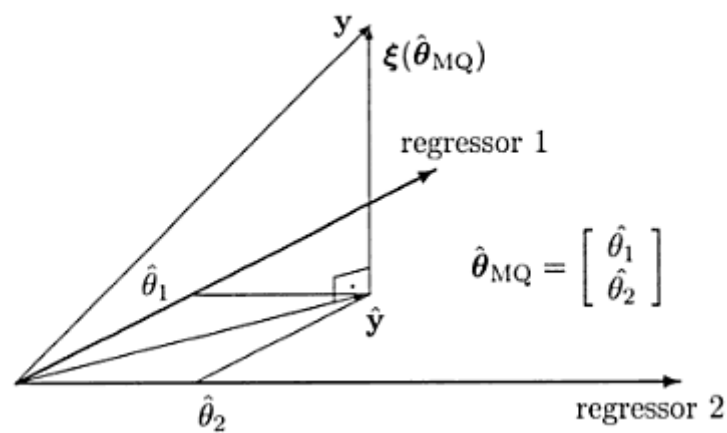
### 3.4.2 - Ortogonalidade

Como analisado por Sousa (2005, p.36) “quando a estimação de parâmetro de um modelo qualquer é feita utilizando todos os possíveis termos candidatos, a sua matriz de regressores é normalmente mal condicionada, indicando que as colunas de regressores da matriz  $X$  são correlacionadas”. Com o propósito de evitar este problema técnicas de ortogonalização devem ser utilizadas, pois com o emprego destes artifícios a coluna de

regressores torna-se não correlacionada fazendo com que a base fique ortogonal para o espaço vetorial da solução.

Como demonstrado por Aguirre (2007, p.231) as propriedades de ortogonalidade podem ser explicadas ilustrativamente como na figura 6

Figura 6: Propriedade de ortogonalidade do estimador MQ



Fonte: Aguirre (2007)

Segundo Aguirre (2007, p.231) o esquema da figura 6 pode ser interpretado da seguinte forma: Os regressores da equação (3.4) originam uma matriz de regressores, a qual pode ser analisada geometricamente como sendo uma hipersuperfície no espaço representadas pelos regressores 1 e 2. A variável dependente,  $y(k)$  origina um vetor de dados observados,  $y$ , o qual forma uma projeção ortogonal aos vetores de regressão 1 e 2,  $\hat{y}$ , ou seja,

$$\hat{y} = X\hat{\theta} \quad (3.8)$$

E a diferença entre os vetores de dados observados  $y$ , gerados pela variável dependente  $y(k)$ , e a sua projeção ortogonal ao plano representado por  $\hat{y}$ , gerado pelos regressores, é denominado vetor resíduo, representado por  $\xi$ .

Logo, a figura 6 deve ser interpretada da seguinte maneira. Ao analisar um vetor de dados  $y$ , juntamente com a matriz de regressão,  $X$ , será criado um vetor  $\hat{y}$  de projeção ortogonal ao vetor de dados sobre a hipersuperfície originada pelas colunas dos regressores,  $X$ , sendo a matriz de projeção  $P(X)$  descrita pela equação 3.9  $\hat{y} = X\hat{\theta} =$

$$X[X^T X]^{-1}X^T y = P(X)y \quad (3.9)$$

### 3.5 - Validação do Modelo

Tendo obtido um grupo de modelos, é necessário verificar se eles incorporam ou não as características de interesse do sistema original. Além de comparar os modelos entre si e decidir se há algum candidato significativamente melhor que os demais, o resultado da validação dependerá da aplicação pretendida para o modelo e da quantidade de informação disponível sobre o sistema original (AGUIRRE, 2007).

A validação dos modelos podem ser divididas em duas partes: validação estatística e validação dinâmica. A primeira foi descrita da seguinte forma em Sousa (2005, p.38)

A validação estatística utiliza as funções de correlação para verificar a existência de alguma dinâmica não modelada nos resíduos. Para sistemas lineares é necessário verificar se os resíduos  $\xi(k)$  são brancos e não correlacionados com a entrada. Isto pode ser feito calculando-se as funções de auto correlação dos resíduos e correlação cruzada dos resíduos com a entrada. Para sistemas não lineares esses testes não são adequados por não detectarem termos cruzados de ruído.

Sendo necessário a este tipo de validação a utilização de funções de auto-correlação não lineares (Billings et al. 1983).

Este tipo de validação é denominada desta forma devido os critérios utilizados se basearem em parâmetros estatísticos, mas este método pode não representar corretamente o comportamento de modelos dinâmicos, sendo necessário empregar outro critério de validação: a validação dinâmica, o qual é explicado por Sousa (2005, p.39)

A validação dinâmica verifica se o modelo identificado apresenta características dinâmicas semelhantes à dinâmica do sistema original, comparando-se a saída do sistema à saída do modelo. Para este tipo de validação a massa de dados usada na excitação do sistema e do modelo deve ser diferente dos dados utilizados nas demais fases de identificação

## 4. Identificação de Sistemas Não lineares: Algoritmos

Um problema comum a todas as representações de sistemas dinâmicos não lineares é a necessidade de determinar a estrutura, a topologia de cada modelo (AGUIRRE, 2007). Logo, os algoritmos de identificação de sistemas têm o objetivo de resolver o problema da escolha de estrutura e da estimação de parâmetros. Como se sabe, há diversos algoritmos para a estimação de parâmetros de modelos matemáticos. Vários deles podem ser agrupados na família de estimadores derivados a partir de estimador MQ. Tais algoritmos são robustos e, normalmente, fáceis de implementar (AGUIRRE, 2007).

Os principais métodos aplicados nos algoritmos MQ Ortogonais são:

- O método clássico de Gram-Schmidt (CGS)
- O método modificado de Gram-Schmidt (MGS)
- O método de Golub-Householder (GH)

Durante o desenvolvimento deste trabalho será utilizado apenas o algoritmo de Golub-Householder, devido ser um método de fácil implementação e é o mais utilizado.

### 4.1. O Método Golub-Householder

Para realizar a estimação dos parâmetros do modelo foi utilizado o método dos mínimos quadrados e o algoritmo de Golub-Householder (AGUIRRE, 2007) para a determinação da taxa de redução do erro. Definindo-se a equação matricial

$$y = \Psi\hat{\theta} + \xi \quad (3.10)$$

Onde,

$$y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}; \Psi = [\Psi_1 \quad \dots \quad \Psi_{n_\theta}]; \hat{\theta} = \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_{n_\theta} \end{bmatrix}; \xi = \begin{bmatrix} \xi(1) \\ \xi(2) \\ \vdots \\ \xi(N) \end{bmatrix}$$

e

$$\Psi = \begin{bmatrix} \Psi_i(1) \\ \Psi_i(2) \\ \vdots \\ \Psi_i(N) \end{bmatrix}, i = 1, 2, \dots, n_\theta$$

Na primeira equação,  $\Psi$  representa a matriz de regressores,  $\hat{\theta}$  representa os parâmetros estimadores e  $\xi$  representa o erro de estimação, isto é, o quanto o  $\hat{y}$  estimado difere do  $y$  real. Assim, considerando uma matriz de regressores candidatos  $\Psi$ , como,

$$\Psi = [y(k-1), y(k-2), u(k-1), \dots, y(k-1)u(k-1), y(k-2)u(k-2)] \quad (3.11)$$

A matriz de regressores  $\Psi$  é montada utilizando as saídas medidas do sistema, e seria dada como:

$$\Psi = \begin{bmatrix} y(0) & y(-1) & u(0) & \dots & y(0)u(0) & y(-1)u(0) \\ y(1) & y(0) & u(1) & \dots & y(1)u(1) & y(0)u(1) \\ y(2) & y(1) & u(2) & \dots & y(2)u(2) & y(1)u(2) \\ y(3) & y(2) & u(3) & \dots & y(3)u(3) & y(2)u(3) \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \end{bmatrix}$$

Onde as saídas, ao instante anterior a 0, são consideradas 0.

Considerando a equação

$$y = \Psi\theta \quad (3.12)$$

Para se determinar  $\hat{\theta}$ , pode-se fazer.

$$\Psi^T y = \Psi^T \Psi \theta \quad (3.13)$$

$$\hat{\theta} = [\Psi^T \Psi]^{-1} \Psi^T y \quad (3.14)$$

## 4.2 Taxa de Redução de Erro

A taxa de redução de erro trata-se de um critério para determinação dos regressores de modelo. Uma observação importante a respeito dos algoritmos apresentados é que os métodos MQO assumem que todas as variáveis independentes foram determinadas (AGUIRRE, 2007). Ou seja, este critério mensura a redução do erro de saída à adição de termos no modelo.

Utilizando a equação

$$\vec{y} = W\vec{\theta} + E\{\xi(k)\} \quad (3.15)$$

Onde:

$y$  = vetor referente às saídas do sistema

$W$  = matriz dos regressores do modelo

$\theta$  = vetor de parâmetros

$E\{\vec{\xi}(k)\}$  = esperança matemática do erro de modelagem

Segundo Rodrigues (2000, p.21) “a redução no valor da variância pode ser normalizada com relação ao erro quadrático médio do sinal de saída”. Com isso taxa de redução de erro, o ERR, pode ser definida como:

$$[ERR]_i = \frac{\theta_i^2 w_i^T T_i}{Y^T Y} \quad i, \dots, n_\theta$$

Ao ser verificado por Sousa (2005, p.31), através do valor de ERR pode-se fazer algumas considerações como: quanto maior o valor de ERR maior será sua importância para o modelo e desta forma os termos de baixo poderão ser descartados, evitando termos desnecessários.

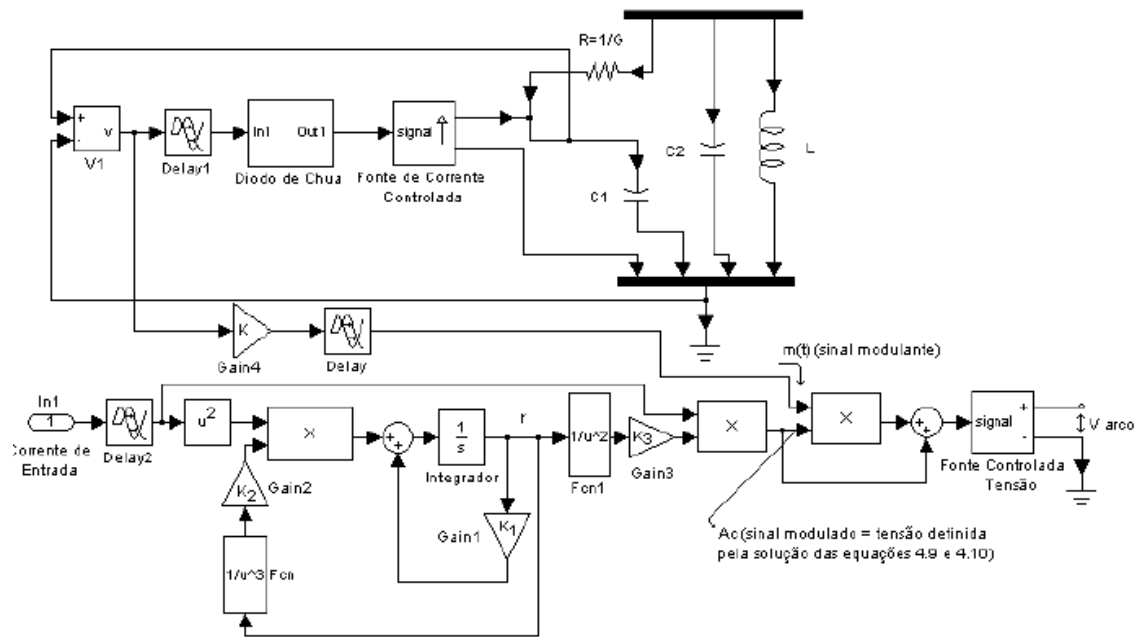


## 5. EXEMPLOS E APLICAÇÕES

A ideia básica da identificação de sistemas é permitir a construção de modelos matemáticos de um sistema dinâmico baseado nos dados obtidos, entrada e saída. A obtenção do modelo mais adequado para a representação do sistema é feita através da escolha da estrutura e da estimação dos parâmetros do modelo (AGUIRRE, 2007). Para tanto, a identificação de sistemas é constituída de cinco etapas, quais sejam: teste dinâmico e coleta de dados, escolha da representação matemática a ser usada, determinação da estrutura do modelo, estimação de parâmetros e validação do modelo.

Em Sousa (2005), o autor analisa dois modelos para um forno elétrico a arco. No primeiro, busca descrever o comportamento dinâmico em função das variações corrente-tensão associadas às variações do arco elétrico, reproduzindo a consequente flutuação de tensão na rede de distribuição de energia. Já no segundo modelo, baseia-se em técnicas de identificação não lineares, através da representação NARMAX polinomial. Esses modelos são estruturas paramétricas construídas a partir dos dados de entrada e saída da rede de distribuição elétrica, no ponto de conexão do forno. Logo, o objetivo principal do autor era reproduzir diretamente os efeitos da modulação de baixa frequência sobre a componente de frequência fundamental da tensão e corrente na rede de distribuição elétrica. Na figura 7 verificamos a implementação das equações dinâmicas do forno a arco associadas ao oscilador de chua.

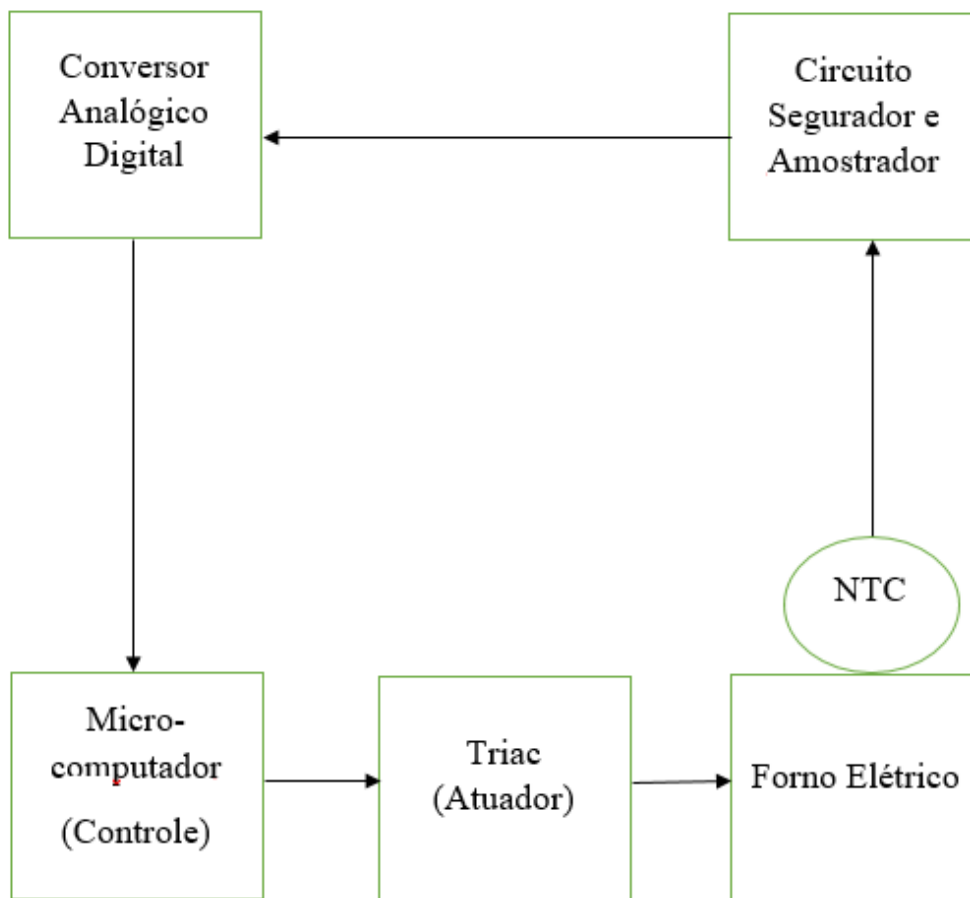
Figura 7: Diagrama de blocos e modelo de forno a arco no ambiente Matlab



Fonte: Sousa 2005

Já Rodrigues (2006) utiliza os modelos NARMAX polinomiais para a representação de sistemas com dinâmicas não lineares, o qual o diagrama de blocos demonstra a estrutura utilizada na figura 8. Os conceitos de agrupamentos de termos e de coeficientes de agrupamentos são utilizados para derivar um procedimento auxiliar de seleção de estrutura de modelos não lineares polinomiais. Conforme o autor, os regimes dinâmicos desse sistema não podem ser reproduzidos por modelos lineares convencionais, portanto, identificá-los constituem um bom teste para avaliar a qualidade dos modelos NARMAX polinomiais na representação de sistemas não lineares reais.

Figura 8: Diagrama de blocos da estrutura utilizado o forno elétrico.



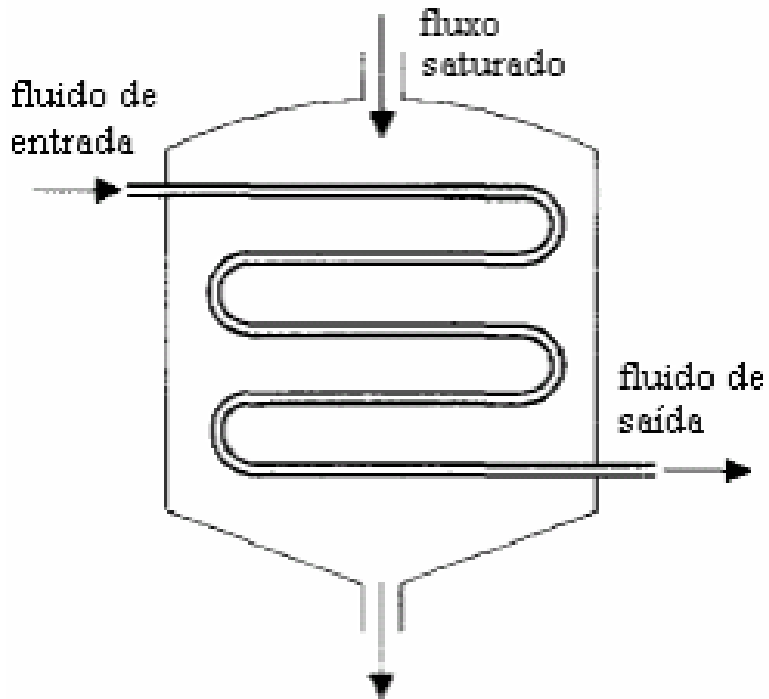
Fonte: Adaptado Rodrigues (2006)

Em Neves (2006), a autora realiza um estudo sobre as técnicas paramétricas de identificação de sistemas no domínio do tempo utilizando Auto-Regressivo de Média Móvel Vetorial (ARMAV) e o modelo Auto-Regressivo Vetorial (ARV). Em ambos os modelos, os procedimentos de identificação dos parâmetros auto-regressivos, responsáveis pela dinâmica do sistema, são estimados utilizando a aproximação dos mínimos quadrados.

Coelho et al. (2006), por sua vez, realizou uma análise sobre a capacidade que o modelo NARMAX polinomial tem de reproduzir o comportamento de sistemas não lineares usando dados reais. Segundo esses autores, esses dados não apresentaram variações abruptas. A utilização de modelos polinomiais NARMAX permitiu verificar a linearidade nos parâmetros, com isso, foi possível aplicar o algoritmo de estimação linear dos mínimos quadrados, verificando também que as propriedades estatísticas e dinâmicas desse modelo são facilmente calculadas, na figura 9 verificamos a representação do

sistema utilizado.. Devido a essas características, o modelo NARMAX foi de utilidade nas representações não lineares, mas mostrou dificuldade nos ajustes dos parâmetros das ordens de atraso para a aplicação do método dos MQO.

Figura 9: Representação do trocador de calor



Fonte: Coelho 2006

Diante do exposto, os trabalhos mencionados demonstram as vastas aplicações e abordam as várias etapas do processo de identificação de sistemas não lineares, essa que se mostra mais adequada e eficiente em sistemas dinâmicos reais. Logo, autores como Aguirre (2007), Neves (2006), Rodriguez (2006), dentre outros, utilizam essas técnicas para a validação dos modelos propostos.

## 6. METODOLOGIA

O presente projeto de pesquisa foi dividido em duas etapas: a primeira etapa focalizou o estudo teórico de trabalhos relacionados com o tema de identificação de sistema não lineares, já na segunda etapa o foco foi a aplicação do fato em estudo.

O estudo prático constituiu de etapas fundamentais para o desenvolvimento do projeto, dentre as quais se destacam o desenvolvimento da programação do algoritmo e a criação da interface com o usuário. Tais etapas possibilitaram a aquisição e tratamento dos dados inseridos no sistema.

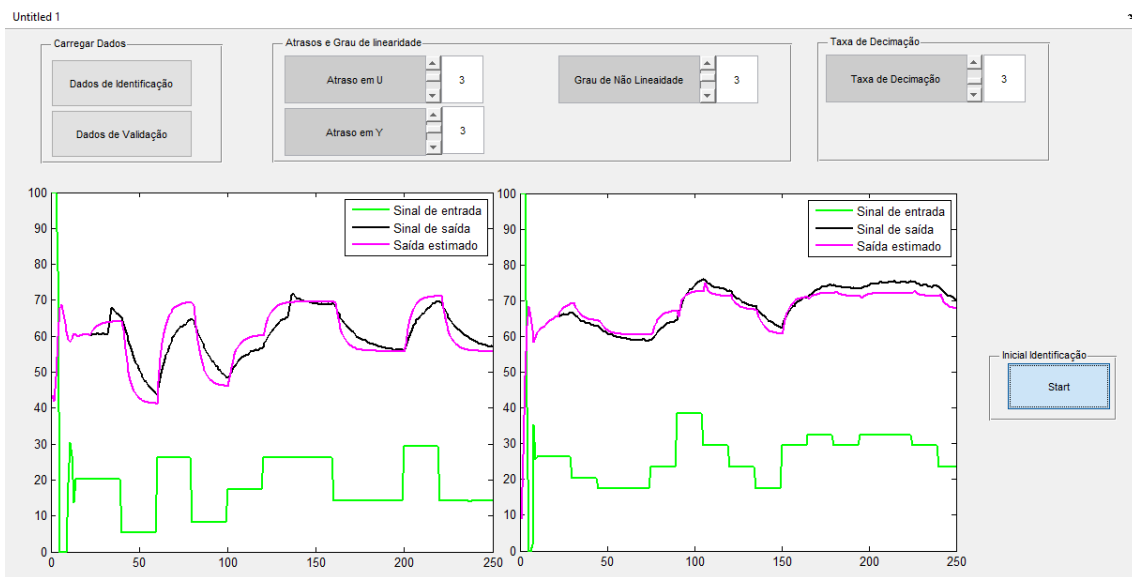
Na etapa de implementação do algoritmo foram desenvolvidas rotinas computacionais que foram executadas pelo software do Matlab, software de alto nível que permite implementação de algoritmos e construções de gráficos.

Já a etapa de criação da interface também foi concebida dentro do software Matlab, mas através de uma interface gráfica com o usuário, a GUI, ambiente idealizado para facilitar e tornar prática a utilização de ferramenta computacional a qual permite interações do software com o usuário.

Logo, foi a partir da relação entre o embasamento teórico e o estudo prático do fato que este trabalho foi realizada, o qual permitiu a obtenção de um software de identificação de sistemas não lineares.

No desenvolvimento do projeto foi utilizado o software Matlab, o qual possibilita a implementação de rotinas e sub-rotinas de lógicas. Além do desenvolvimentos de programas, comandos lógicos o software permite a criação de interfaces com o usuário a GUI interface gráfica do usuário. Desta forma foi desenvolvido tanto a interface do usuário, desenho do *layout* da aplicação e a implementação das rotinas computacionais no Matlab, como demonstrado na figura 10.

Figura 10 : Layout desenvolvido



Fonte: Autor

Figura 11: Trecho rotina implementada para acionamento botões da interface:

```

Editor - C:\Users\Rangel Resende\Desktop\Interface.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
79 % --- Executes on button press in Carregar dados de identificação.
80 function pushbutton2_Callback(hObject, eventdata, handles)
81 global entrada saida tam_amostra_sai tam_amostra_entr
82 % hObject handle to pushbutton2 (see GCBO)
83 % eventdata reserved - to be defined in a future version of MATLAB
84 % handles structure with handles and user data (see GUIDATA)
85 cla(handles.axes1);
86 uiopen('*.mat');
87 axes(handles.axes1);
88 entrada;
89 saida;
90 tam_amostra_entr = length(entrada);
91 tam_amostra_sai = length(saida);
92 title('Grafico de Dados de Identificação');
93 handle = plot (entrada,'g');
94 set(handle,'LineWidth',[1.5]);
95 hold on
96 handle = plot (saida,'k');
97 set(handle,'LineWidth',[1.5]);
98 legend ('Sinal de entrada','Sinal de saída');
99
100
101 % --- Executes on button press in Carregar dados de validação.
102 function pushbutton3_Callback(hObject, eventdata, handles)
103 global entrada2 saida2 tam_amostra_sai2 tam_amostra_entr2
104 % hObject handle to pushbutton3 (see GCBO)
105 % eventdata reserved - to be defined in a future version of MATLAB
106 % handles structure with handles and user data (see GUIDATA)
107 cla(handles.axes2);
108 uiopen('*.mat');
109 axes(handles.axes2);
110 entrada2=entrada;
111 saida2=saida;

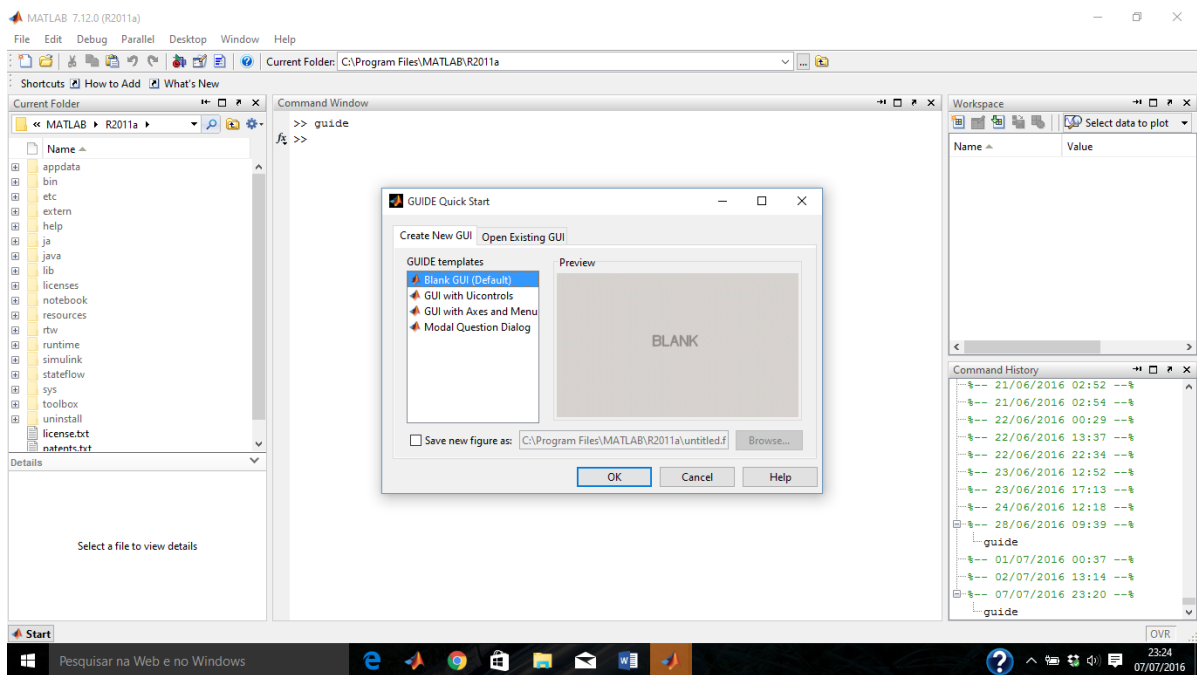
```

Fonte: Autor

## 6.1 - Interface

Como mencionado anteriormente o desenvolvimento da interface passa pelo *layout* e pelo desenvolvimento das rotinas de cada elemento criado, dentro do seu *layout*, nas chamadas *callback functions*. O acesso ao ambiente de desenvolvimento ocorre ao se digitar no *workspace* do matlab o comando “guide”, figura 12.

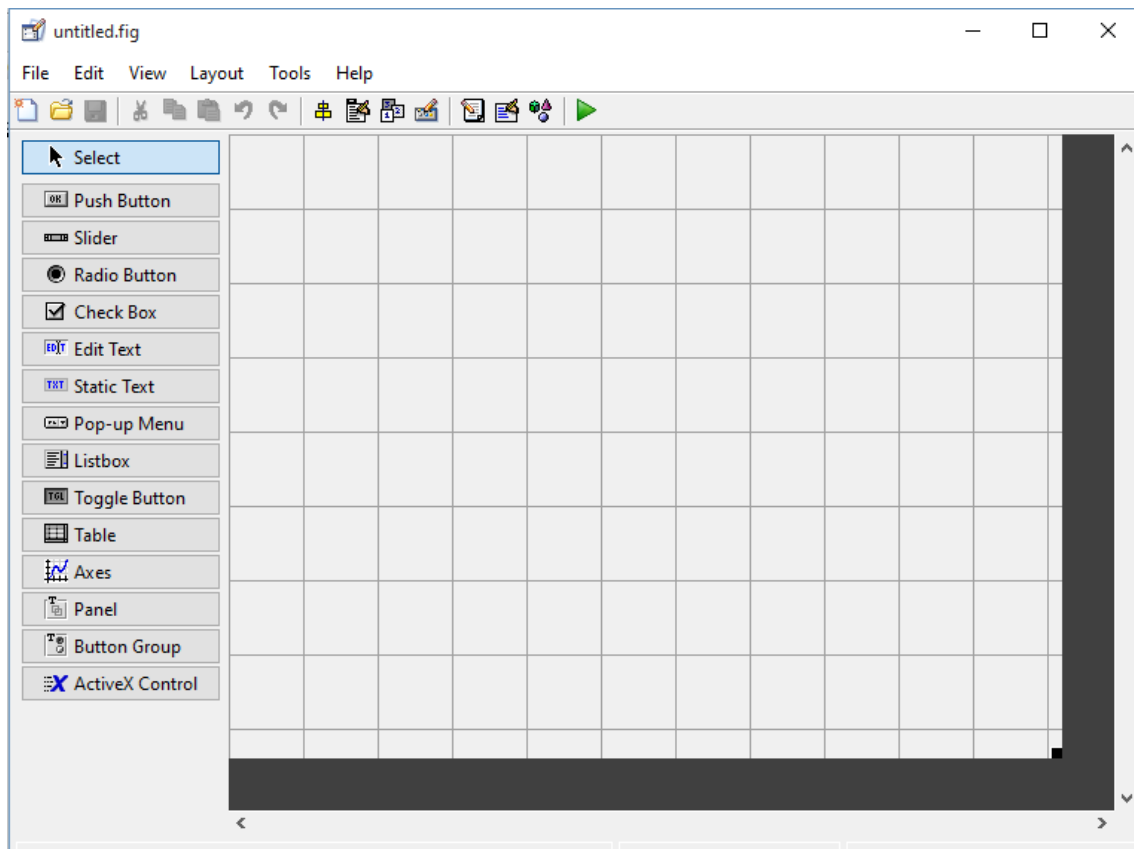
Figura 12: Inicialização interface Gráfica com o Usuário GUI



Fonte: Autor

Na tela inicial de projetos, figura 13, são apresentados os elementos de desenvolvimento da tela, desde botões, gráficos, caixa de textos entre outros elementos.

Figura 13: Tela de Inicialização GUIDE



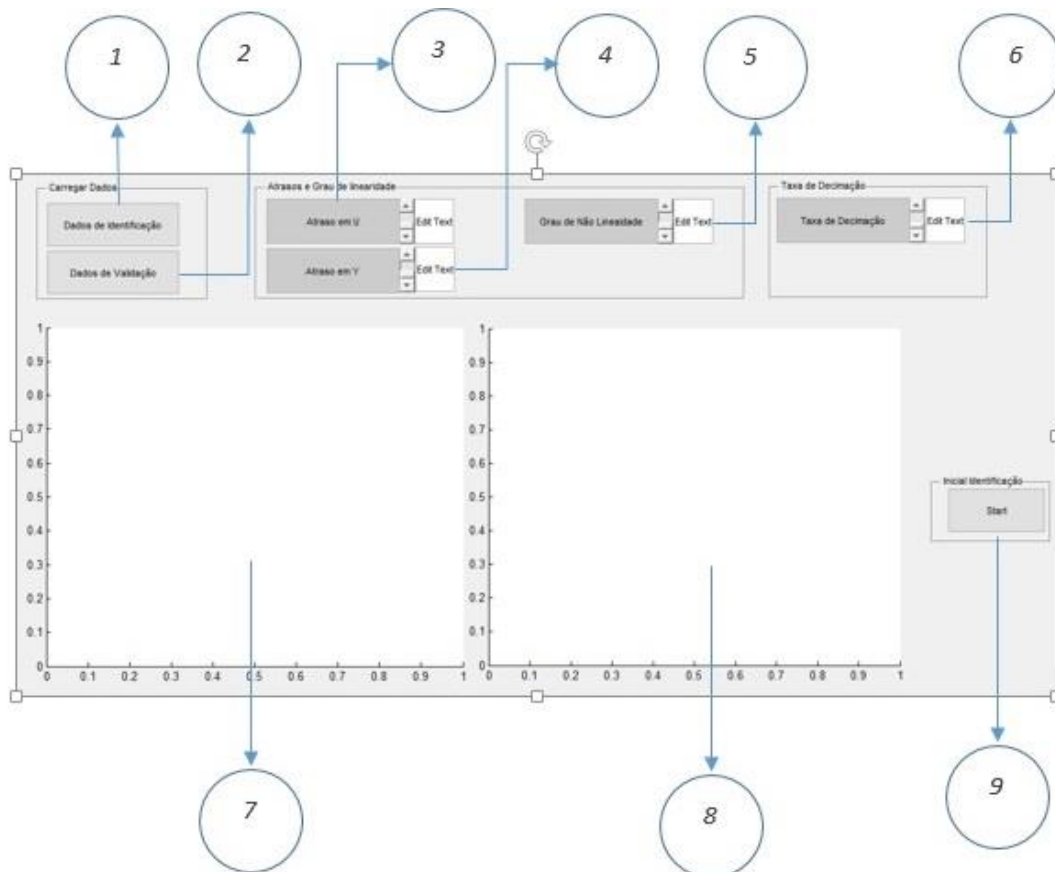
Fonte: Autor

Todo elemento inserido na tela, ao ser selecionado realizará uma chamada de rotina, ou seja, ao selecionar algum elemento da tela automaticamente “chama sua função”, lógica desenvolvida, e retorna algo à interface como um valor, plota um gráfico, apresenta uma informação.

Com a finalidade da criação do software foi desenvolvido o seguinte layout, figura 14, com suas respectivas funções.



Figura 14: Interface Desenvolvida



Fonte: Autor

Descrição dos botões e elementos da interface de identificação:

- 1- Dados de Identificação – Botão de seleção de dados a serem identificados, a seleção ocorre através da abertura de uma janela de dados armazenados, os quais podem alimentar o sistema.
- 2- Dados de Validação – Botão de seleção de dados os quais validam a identificação obtida, a seleção ocorre através da abertura de uma janela de dados armazenados, os quais podem alimentar o sistema.
- 3- Atraso U – Permite determinar o atraso U, atraso do sinal de entrada, a ser utilizado no programa, essa variável é definida por meio de um slider limitado de zero a quatro.

- 4- Atraso Y – Permite determinar o atraso Y, atraso do sinal de saída, a ser utilizado no programa, essa variável é definida por meio de um slider limitado de zero a quatro.
- 5- Grau de Não Linearidade – Determina o grau de não linearidade que o sistema será submetido, possuindo uma faixa, com range de um a quatro.
- 6- Taxa de Decimação – Taxa de amostragem utilizada no *software*, ou seja, os intervalos de uma amostra a outra dos elementos selecionados.
- 7- Gráfico de Identificação – A área deste gráfico é reservada para plotar os dados de entrada e saída juntamente com a saída estimada pelo sistema identificado.
- 8- Gráfico de Validação – A área deste gráfico é reservada para plotar os dados de entrada e saída juntamente com a saída estimada, além de comparar a saída estimada do método proposto, sistema não linear, com a saída estimada de um modelo linear.
- 9- Start – Inicializa a identificação após ser definido os valores das variáveis já citadas (itens de 1 a 6).

## **6.2 - Desenvolvimento da rotinas e sub-rotinas para a implementação do algoritmo de Householder**

### **6.2.1 - Coleta de dados**

O presente trabalho não foi utilizado o procedimento de coleta de dados e sim o de seleção de dados, ou seja, o foco será na análise e interpretação da massa de dados fornecida. Sendo utilizado neste trabalho dados obtidos de Aguirre (2007).

### **6.2.2 - Escolha da representação matemática**

A escolha da representação é uma importante questão na modelagem dos sistemas, pois esta escolha influenciará significativamente na determinação do modelo resultante. Algumas características devem ser observadas para a escolha da representação, como por exemplo a linearidade do sistema, mas como o objetivo deste trabalho é a análise de sistemas não lineares utilizando os sinais obtidos de entrada e saída, a representação mais conveniente encontrada foi a escolha da representação NARMAX (Nonlinear

autoregressive moving average model with exogenous input), cuja representação é determinada pela combinação de dados de entrada, saída e pelo erro do sistema modelado.

### 6.2.3 - Determinação da Estrutura do Modelo a ser Utilizado

A concepção da estrutura do modelo pode ser dividida em algumas partes importantes: o desenvolvimento da matriz de regressores (termos candidatos) e a implementação do método de Golub-Householder, que possibilitará encontrar os melhores termos para o sistema, ou seja, determinar os termos que melhor representarão as características do sistema.

#### 6.2.3.1 - Desenvolvimento Matriz de Regressores

Com a finalidade tornar o *software* mais ágil essa etapa é dividida em duas, uma desenvolvida nas linhas de programação da aplicação e a outra externamente a isso, a separação dessas subetapas é importante, pois ao analisarmos, que os termos da matriz genérica serão os mesmos ao utilizar a mesma configuração, valores de parâmetros (atraso U, atraso Y e grau de linearidade), diferenciando-os pelos dados utilizados. Desta forma o *software* não precisa gastar tempo para ficar calculando e recalculando os termos da matriz genérica, deixando assim de gastar alguns minutos.

Exemplos:

1- Atraso X: 2; Atraso Y: 2; Grau de Não Linearidade: 3

Combinações possíveis (Termos Candidatos): 35 combinações distintas

1,  $u(k-1)$ ,  $u(k-2)$ ,  $y(k-1)$ ,  $y(k-2)$ ,  $u(k-1)*u(k-2)^2$ ,  $u(k-1)^2*u(k-2)$ ,  $u(k-1)*y(k-1)^2$ ,  $u(k-1)^2*y(k-1)$ ,  $u(k-1)*y(k-2)^2$ ,  $u(k-2)*y(k-1)^2$ ,  $u(k-1)^2*y(k-2)$ ,  $u(k-2)^2*y(k-1)$ ,  $u(k-2)*y(k-2)^2$ ,  $u(k-2)^2*y(k-2)$ ,  $y(k-1)*y(k-2)^2$ ,  $y(k-1)^2*y(k-2)$ ,  $u(k-1)^2$ ,  $u(k-1)^3$ ,  $u(k-2)^2$ ,  $u(k-2)^3$ ,  $y(k-1)^2$ ,  $y(k-1)^3$ ,  $y(k-2)^2$ ,  $y(k-2)^3$ ,  $u(k-1)*u(k-2)$ ,  $u(k-1)*y(k-1)$ ,  $u(k-1)*y(k-2)$ ,  $u(k-2)*y(k-1)$ ,  $u(k-2)*y(k-2)$ ,  $y(k-1)*y(k-2)$ ,  $u(k-1)*u(k-2)*y(k-1)$ ,  $u(k-1)*u(k-2)*y(k-2)$ ,  $u(k-1)*y(k-1)*y(k-2)$ ,  $u(k-2)*y(k-1)*y(k-2)$

2- Atraso X: 4; Atraso Y: 4; Grau de Não Linearidade: 2

Combinações possíveis (Termos Candidatos): 45 combinações distintas

1,  $u(k - 1)$ ,  $u(k - 2)$ ,  $u(k - 3)$ ,  $u(k - 4)$ ,  $y(k - 1)$ ,  $y(k - 2)$ ,  $y(k - 3)$ ,  $y(k - 4)$ ,  $u(k - 1)^2$ ,  $u(k - 2)^2$ ,  $u(k - 3)^2$ ,  $u(k - 4)^2$ ,  $y(k - 1)^2$ ,  $y(k - 2)^2$ ,  $y(k - 3)^2$ ,  $y(k - 4)^2$ ,  $u(k - 1)u(k - 2)$ ,  $u(k - 1)u(k - 3)$ ,  $u(k - 1)u(k - 4)$ ,  $u(k - 2)u(k - 3)$ ,  $u(k - 2)u(k - 4)$ ,  $u(k - 3)u(k - 4)$ ,  $u(k - 1)y(k - 1)$ ,  $u(k - 1)y(k - 2)$ ,  $u(k - 2)y(k - 1)$ ,  $u(k - 1)y(k - 3)$ ,  $u(k - 2)y(k - 2)$ ,  $u(k - 3)y(k - 1)$ ,  $u(k - 1)y(k - 4)$ ,  $u(k - 2)y(k - 3)$ ,  $u(k - 3)y(k - 2)$ ,  $u(k - 4)y(k - 1)$ ,  $u(k - 2)y(k - 4)$ ,  $u(k - 3)y(k - 3)$ ,  $u(k - 4)y(k - 2)$ ,  $u(k - 3)y(k - 4)$ ,  $u(k - 4)y(k - 3)$ ,  $u(k - 4)y(k - 4)$ ,  $y(k - 1)y(k - 2)$ ,  $y(k - 1)y(k - 3)$ ,  $y(k - 1)y(k - 4)$ ,  $y(k - 2)y(k - 3)$ ,  $y(k - 2)y(k - 4)$ ,  $y(k - 3)y(k - 4)$

No desenvolvimento e cálculo de todas as combinações dos termos genéricos possíveis foi desenvolvida em uma rotina à parte e posteriormente inserido em uma subrotina do software. Tendo assim uma maior agilidade devido não ser necessário ficar recalculando e determinando os termos

### 6.2.3.2 - Matriz de Regressores Genérica

A criação da matriz de termos candidatos segue o seguinte princípio: cria-se primeiramente um vetor base, formado pelas variáveis atraso X e atraso Y, as quais irão de  $u(k-1)$  à  $u(k-\text{atraso X})$  e  $y(k-1)$  à  $y(k-\text{atraso Y})$  onde as letras **u** e **y** representam respectivamente os dados de entrada e saída que serão inseridos na aplicação, e juntamente com o número 1, o qual representa um constante para o sistema. Já o grau de não linearidade fará com que seja criado mais dois vetores base iguais a esse, após isso será feita a multiplicação termo a termo desses três vetores.

Exemplo3:

Atraso X: 2; Atraso Y: 2; Grau de Não Linearidade: 3

Vetor base = [ $u(k - 1)$ ,  $u(k - 2)$ ,  $y(k - 1)$ ,  $y(k - 2)$ , 1]

Multiplicação de vetores base = Vetor base<sub>1</sub> x Vetor base<sub>2</sub> x Vetor base<sub>3</sub>

Quantidade de termos gerados = 125

Termos distintos= 35

Termos candidatos = [ 1, u(k - 1), u(k - 2), y(k - 1), y(k - 2), u(k - 1)\*u(k - 2)^2, u(k - 1)^2\*u(k - 2), u(k - 1)\*y(k - 1)^2, u(k - 1)^2\*y(k - 1), u(k - 1)\*y(k - 2)^2, u(k - 2)\*y(k - 1)^2, u(k - 1)^2\*y(k - 2), u(k - 2)^2\*y(k - 1), u(k - 2)\*y(k - 2)^2, u(k - 2)^2\*y(k - 2), y(k - 1)\*y(k - 2)^2, y(k - 1)^2\*y(k - 2), u(k - 1)^2, u(k - 1)^3, u(k - 2)^2, u(k - 2)^3, y(k - 1)^2, y(k - 1)^3, y(k - 2)^2, y(k - 2)^3, u(k - 1)\*u(k - 2), u(k - 1)\*y(k - 1), u(k - 1)\*y(k - 2), u(k - 2)\*y(k - 1), u(k - 2)\*y(k - 2), y(k - 1)\*y(k - 2), u(k - 1)\*u(k - 2)\*y(k - 1), u(k - 1)\*u(k - 2)\*y(k - 2), u(k - 1)\*y(k - 1)\*y(k - 2), u(k - 2)\*y(k - 1)\*y(k - 2)]

Segue abaixo trecho programação para gerar matriz genérica para atraso x igual a 2

```
% atraso 2
elseif a==2
    if grau==1
        R=V;
    elseif grau==2
        for j=1:f
            for j1=1:f
                R(i,m)=V(i,j)*V(i,j1);
                m=m+1;
            end
        end
    elseif grau==3
        for j=1:f
            for j1=1:f
                for j2=1:f
                    R(i,m)=V(i,j)*V(i,j1)*V(i,j2);
                    m=m+1;
                end
            end
        end
    end
else
    for j=1:f
        for j1=1:f
            for j2=1:f
                for j3=1:f
                    R(i,m)=V(i,j)*V(i,j1)*V(i,j2)*V(i,j3);
                    m=m+1;
                end
            end
        end
    end
end
```

Após a geração de todas as possíveis combinações do sistema, as matrizes foram inseridas nas linhas de programação dentro de uma função, a qual preenche a matriz com os dados selecionados pelo usuários.

#### **6.2.4 - Determinação da estrutura do modelo**

Devido haver uma alta complexidade para a inclusão de termos candidatos no modelo de sistemas não lineares, mesmo considerando sistemas de baixa ordem, tendo em vista o elevado número de termos. Com o intuito de resolver esta questão foi adotado o critério da taxa de redução de erro (ERR).

A taxa de redução de erro, Erro Reduction Ratio (ERR), corresponde à soma dos erros quadráticos gerados por cada termo, ou seja, consiste em mensurar a redução do erro de saída devido à inserção de cada termo. Possibilitando desta forma determinar termos relevantes e irrelevantes, sendo possível escolher, selecionar os termos mais significativos ao modelo. Sendo que quanto maior o índice ERR maior será sua importância no modelo.

Sendo assim, a estrutura do modelo foi determinada da seguinte forma: de posse da matriz genérica de termos candidatos, preenche-se a matriz com os dados selecionados, campo, dado de identificação da interface, e o critério ERR seleciona os termos com maior coeficiente, ou seja, os termos mais revelantes,

Como demonstrado no trecho abaixo, parte da programação desenvolvida no software matlab, pelo autor Aguirre, o qual possibilitará o cálculo do ERR

```

for j=1:np % Opera por colunas, ate o numero de termos final

    % Determina err para demais regressores e volta a escolher
    % o de maior valor

    for k=j:n-1 % ate completar o numero de termos candidatos

        c(k) = ((A(j:m,k)'*A(j:m,n))^2) / ((A(j:m,k)'*A(j:m,k))*yy); %
        err do regressor
    end;

    [ans,aux]= max(c(j:n-1));
    jm=j+aux-1;
    err(j)=ans;
    aux=A(:,jm); % coluna do regressor com maior err
    A(:,jm)=A(:,j);
    A(:,j)=aux;
    aux=piv(jm); % indice do regressor com maior err
    piv(jm)=piv(j);
    piv(j)=aux;

x=A(j:m,j);
% v=house(x)
% Do livro Matrix Computations 2a Ed. pg 196
% Dado um vetor x, volta-se um vetor v de tal forma
% que (I-2vv'/v'v)x é zero à excecao do primeiro elemento

nx=length(x);
u=norm(x,2);
v=x;
if u ~= 0
    b=x(1) + sign(x(1))*u;
    v(2:nx) = v(2:nx)/b;
end;
v(1)=1;
% fim house(x)

a=A(j:m,j:n);

% a=rowhouse(a,v)
% Dada uma matriz A (m,n), e um vetor de comprimento m, v,
% cujo primeiro elemento é 1, este algoritmo substitui
% A por P*A onde P=I-2vv'/v'v

b=-2/(v'*v);
w=b*a'*v;
a=a+v*w';
% fim rowhouse(a,v)

A(j:m,j:n)=a;

end;
% fim myhouse(A)
Piv=piv(1:np

```

### **6.2.5 - Validação**

Como verificado em Aguirre (2007), após inúmeras etapas no processo de identificação deseja-se constatar que o modelo obtido possui características relevantes do sistema, ou seja, o modelo determinado apresenta propriedades interessantes do sistema observado.

Com o intuito de tornar válido o modelo obtido, o modelo encontrado é submetido a uma nova massa de dados, ou seja, é efetuado dois testes no sistema, o sistema é excitado com impulsos diferentes, a ser observado, os quais são independentes um do outro. Sendo a primeira amostra de dados utilizada para a identificação do sistema e a segunda massa de dados utilizada na validação do sistema. Importante lembrar que os dados extraídos devem ser obtidos de um planta em operação em condições semelhantes.

Desta forma pode se garantir que o modelo matemático obtido representa as características do sistema em estudo.

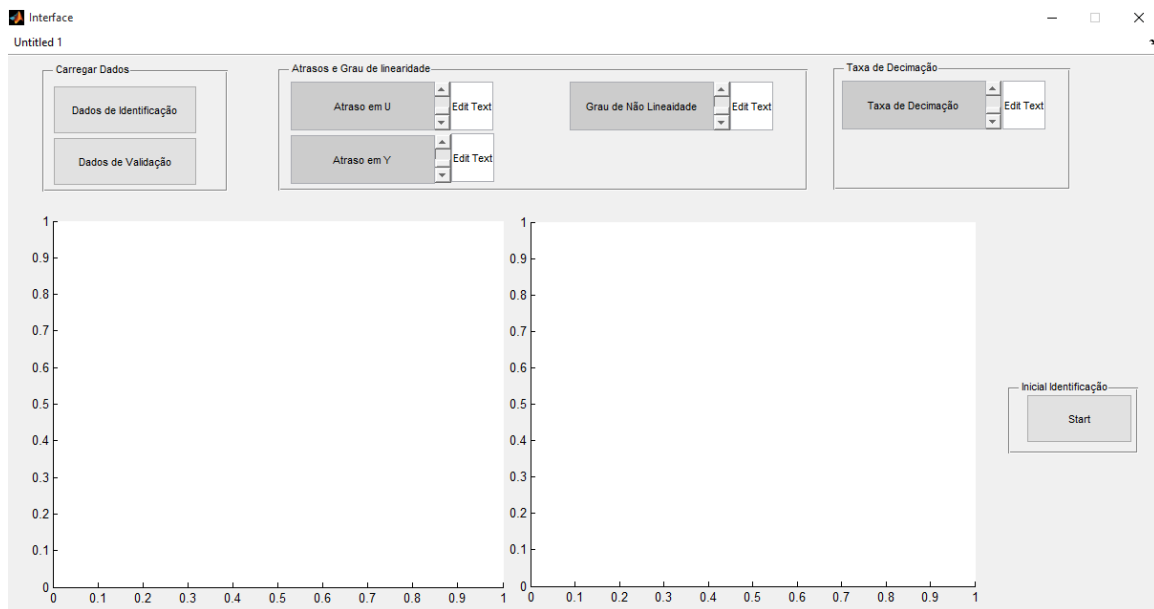


## 7. RESULTADOS

No presente trabalho, ao se desenvolver o software de identificação de sistemas utilizou-se dados já experimentados por outros autores, dados de entrada e saída de determinadas plantas. Com o intuito de demonstrar a funcionalidade e aplicabilidade do trabalho segue os resultados do *software* a ser aplicado a dois sistemas distintos.

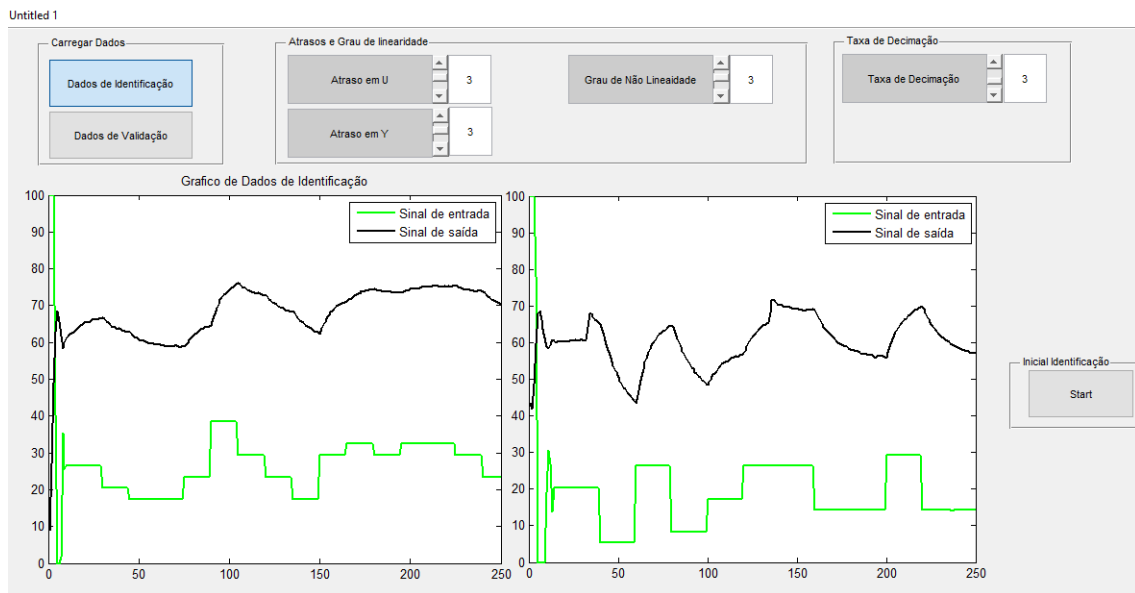
No primeiro exemplo, ao se configurar o *software* como mostra a figura 15 e selecionar os arquivos .mat (arquivos de extensão do Matlab), testgrafico3 e testgrafico2, respectivamente nos campos dados de identificação e dados de validação, o programa irá apresentar, a princípio, apenas o gráficos destas informações selecionadas, vide figura 16

Figura 15: Configuração parâmetros teste 1



Fonte: Autor

Figura 16: Gráfico Dados Utilizados Teste 1



Fonte: Autor

Ao iniciar o sistema nestas configurações, será apresentada algumas informações relevantes ao usuário, tais como: termos candidatos, estimação parâmetro e o próprio modelo do sistema.

Como mencionado no tópico anterior, o algoritmo de Householder juntamente com a taxa de redução de erro ERR determinam os termos que serão utilizados, ou seja, dos 84 termos da matriz de regressores o algoritmo irá selecionar os melhores termos, neste exemplo dos 84 termos será escolhida apenas 9 .

[ 1,  $u(k - 1)$ ,  $u(k - 2)$ ,  $u(k - 3)$ ,  $y(k - 1)$ ,  $y(k - 2)$ ,  $y(k - 3)$ , ... ,  $u(k - 1)*y(k - 2)*y(k - 3)$ ,  $u(k - 2)*y(k - 1)*y(k - 3)$ ,  $u(k - 3)*y(k - 1)*y(k - 2)$ ,  $u(k - 2)*y(k - 2)*y(k - 3)$ ,  $u(k - 3)*y(k - 1)*y(k - 3)$ ,  $u(k - 3)*y(k - 2)*y(k - 3)$ ,  $y(k - 1)*y(k - 2)*y(k - 3)$ ]

Destes se seleciona apenas nove termos, os quais representam mais significativamente as características do sistema.

[  $y(k-1)$ ,  $y(k-2)$ ,  $u(k-2)$ ,  $u(k-1)^3$ ,  $u(k-2)^2*y(k-3)$ ,  $u(k-1)^2$ ,  $y(k-1)*y(k-3)^2$ ,  $u(k-1)*u(k-3)^2$ ,  $u(k-2)*y(k-2)^2$ ]

Sendo:

$u$  = dados referentes à entrada do sistema

y= dados referente à saída do sistema.

(k-n) = argumento que indica que os regressores são tomados k a n instante.

Ainda com esta configuração podemos observar os parâmetros estimados, logo o modelo do sistema a ser analisado.

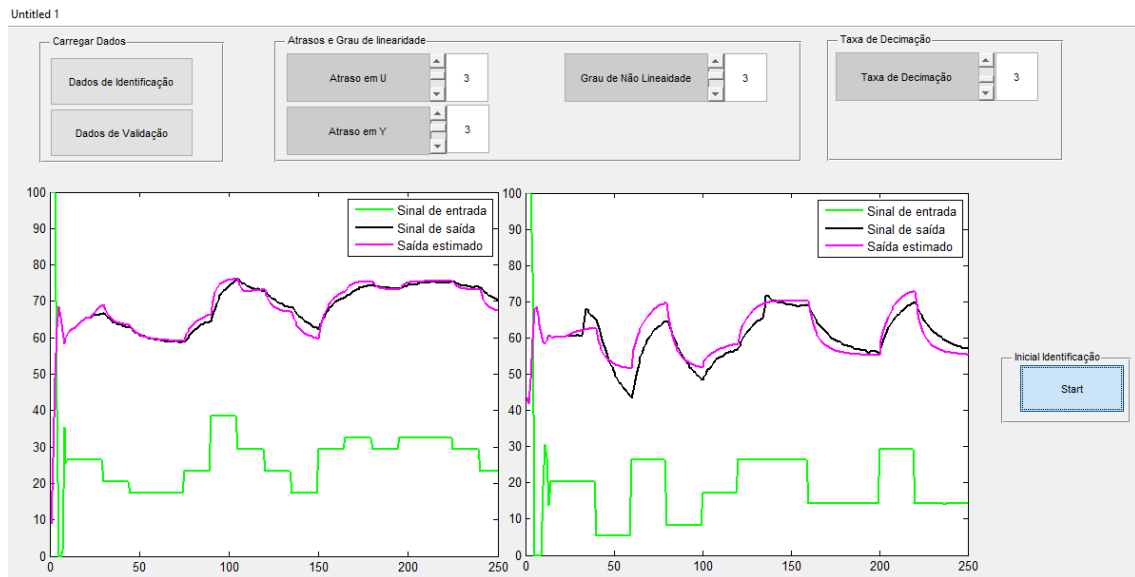
$\theta$

= [0,9963 ; 0,072; -0,25; - 1,76x10<sup>-5</sup>; 3,74x10<sup>-6</sup>; 12,85x10<sup>-3</sup>; -2,28x10<sup>-5</sup>; +5,27x10<sup>-5</sup>]

$$\text{Modelo} = -1,76x10^{-4}u(k-1)^3 + 12,85x10^{-3}u(k-1)^2 + 5,27x10^{-5}u(k-1)u(k-3)^2 + 4,69x10^{-7}y(k-1)u(k-2)^2 + 3,74x10^{-6}u(k-2)y(k-2)^2 - 0,25u(k-2) + 0,072y(k-2) - 2,28x10^{-5}y(k-1)y(k-3)^2 + 0,9963y(k-1) \quad (7.1)$$

Através do modelo matemático 7.1 podemos observar o comportamento do sistema ao longo do intervalo, durante a etapa de validação, figura 17.

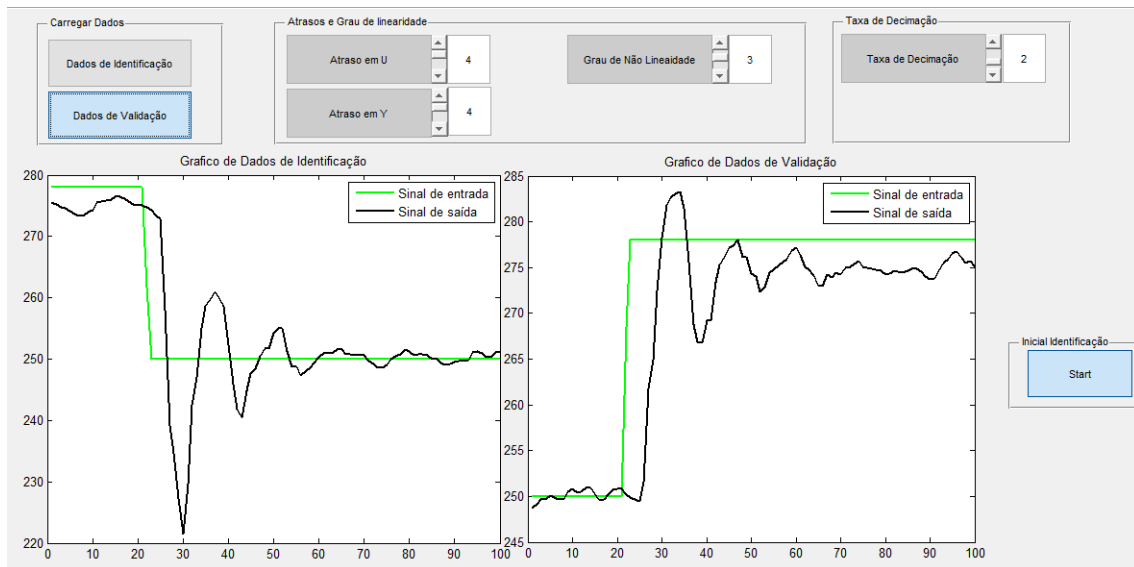
Figura 17: Gráficos Resultados identificação 1



Fonte: Autor

No segundo exemplo proposto utiliza-se as informações de uma outra planta, a qual apresenta o comportamento demonstrado na figura 18 e será parametrizada com os valores mostrados na mesma imagem.

Figura 18: Graficos Dados Teste 2 e Teste 3



Fonte: Autor

Apresentando as seguintes características:

Número de termos candidatos: 165

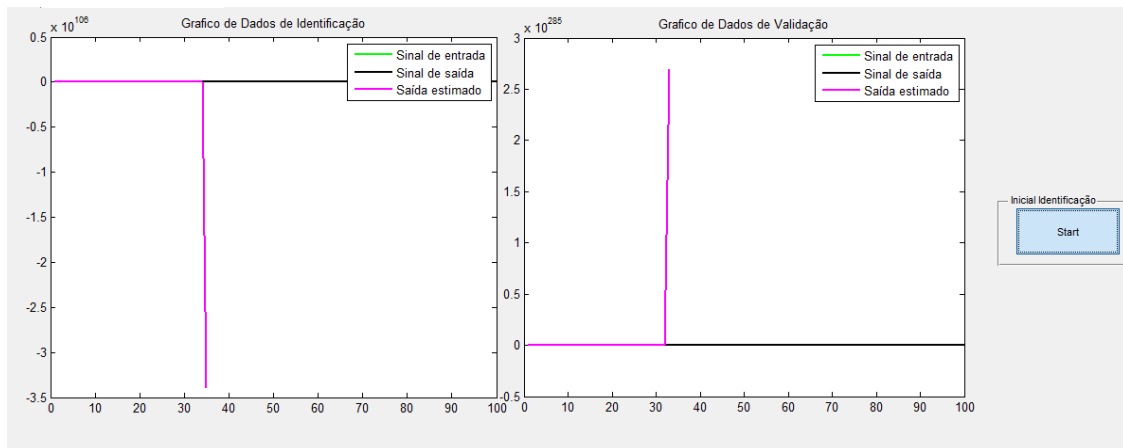
Número de parâmetros, número de termos no modelo final: 11

Termos do modelo final:

[  $u_2$ ,  $y_3*y_4$ ,  $u_3*y_2*y_3$ ,  $y_1^3$ ,  $u_1$ ,  $u_4*y_1*y_4$ ,  $u_3^2$ ,  $u_4*y_3$ ,  $u_3*y_1^2$ ,  $u_1*u_2*u_3$ ,  $u_3*y_2^2$  ]

Comportamento da saída estimada ao longo do tempo, figura 19.

Figura 19: Gráfico resultados teste 2  $n_p=11$

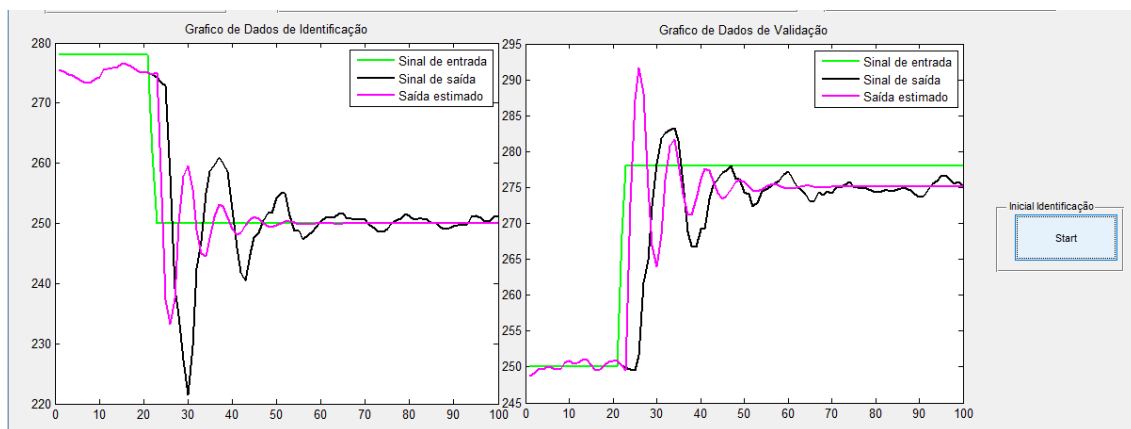


Fonte: Autor

Como demonstrado na figura 19 o resultado encontrado não é satisfatório, mas ele possui este desempenho devido à quantidade de parâmetros no modelo final, isto porque o algoritmo escolheu termos espúrios, ou seja, o modelo final possui elementos desnecessários.

De forma a representar isto, foi alterado o número de parâmetros do modelo para três e foi mantida a configuração anterior, apresentando respostas diferentes como exposto na figura 20

Figura 20: Graficos Resultado Teste 3  $n_p = 3$



Fonte: Autor

Possuindo assim as seguintes características:

Termos do modelo final:

[  $u^2$ ,  $y^3 \cdot y^4$ ,  $u^3 \cdot y^2 \cdot y^3$  ]

$$\theta = [1,272; 1,541 \times 10^{-3}; 1,813 \times 10^{-5} ]$$

Modelo final:  $1,272u(k-2) - 1,541 \times 10^{-3}y(k-3)y(k-4) + 1,813 \times 10^{-5}u(k-3)y(k-2)y(k-3)$

(7.2)

## 8. CONCLUSÃO

Com o desenvolvimento do *software*, observou-se que o tempo para a identificação de sistemas mostrou-se relativamente baixo, sempre realizando a tarefa proposta na casa dos segundos, comprovando que o sistema possui uma boa agilidade, visto que para calcular tais tarefas manualmente levaria um tempo maior, principalmente dependendo da sua complexidade, além da possibilidade de erros humanos durante os cálculos.

Devido aos diversos testes, pode-se verificar que os objetivos propostos foram atendidos, como desenvolvimento de interface que possibilitaria a identificação de sistemas, tratamento adequado aos dados utilizados e utilização do *software* para diversos sistemas.

No decorrer do projeto observou algumas limitações e algumas possíveis melhorias que podem ser feitas no trabalho, dentre estas ressalta-se a criação de uma nova interface com o usuário, que permitiria a seleção do número de parâmetros contidos no modelo final, o qual foi comprovado na seção anterior.

Um mudança de maior impacto seria o desenvolvimento de um sistema de aquisição de dados, o qual poderia ser acionado por meio da interface com o usuário, coletando as respostas do sistema excitado com os sinais estipulados pelo usuário.

Além do desenvolvimento de mais uma janela, ou aba, contendo as respostas obtidas, proporcionando uma melhor visualização das respostas do sistema.

## REFERÊNCIAS

AGUIRE, L. A. **Introdução à identificação de sistemas**: técnicas lineares e não-lineares aplicadas a sistemas reais. Belo Horizonte: Editora UFMG, 2007. 728 p.

ALMEIDA, A. X. C.; RODRIGUES, J. R.; MOREIRA, A. H.; MARTINS, C. A. Identificação e Controle de um servomecanismo utilizando linguagem gráfica. In: **CONEM VI CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA**. Disponível em: <<http://www.abcm.org.br/pt/wp-content/anais/conem/2010/PDF/CON10-0907.pdf>>. Acesso em: 09 nov. 2014.

ANTUNES, F. Uma Nova Abordagem para Representações e Identificações de Classes de Sistemas Dinâmicos Não Lineares. 2007.f.76. Tese (Mestrado em Engenharia Elétrica) – Universidade de Itajúba. Itajúba, 2007. Disponível em: <<http://saturno.unifei.edu.br/bim/0031985.pdf>>. Acesso em 12 jul. 2016.

BARROS, A. J. S. S.; LEHFELD, N. A. S. **Fundamentos de Metodologia Científica**. São Paulo: Editora Prentice Hall, 2007.

Billings, S. A. et al. Structure Detection and Model Validity tests in the identification of nonlinear systems. *Int. J. Control*, Vol. 130, N<sup>o</sup>. 4, July, 1983.

Controle de Processos Industriais Consultoria e Treinamentos. Disponível em: <<http://www.tdps.com.br/aulas/modelagem-e-identificacao/#mod>>. Acesso em 12 ago. 2016

CRUZ, C.; RIBEIRO, U. **Metodologia Científica – Teoria e Prática**. Curitiba: Editora Axcel Books, 2004.

COELHO, L.S.; PESSOA, M. W.; SUGUIMOTO, R. M. Identificação de um sistema dinâmico caótico utilizando modelo NARMAX e mínimos quadrados ortogonais. In: **XIV Seminário iniciação científica e VIII Mostra de pesquisa da PUCRR**. Disponível em: <<http://www2.pucpr.br/reol/index.php/PIBIC2006?dd1=396&dd99=view>>. Acesso em: 02 jan. 2015.



GAUSS, K. G., Theory of Motion of the Heacently Bodies. New York: Dover, 1963.

LJUNG, L. **System identification, Theory for theuser**. New Jersey: Prentice Hall, 1999.

MOREIRA, M. A. G.; TORRES, L. A. B.; PEREIRA, G. A. S.; ISCOLD, P. Localização, modelagem e controle de um mini-helicóptero em ambientes internos. **Sba Controle & Automação** [online], 2011, v. 22, n. 3, pp. 238-255. Disponível em: <<http://dx.doi.org/10.1590/S0103-17592011000300003>>. Acesso em: 09 nov. 2014.

NELLES, O. **Nonlinear system identification**. Berlin/Heidelberg, Springer-Verlag, 2001.

NEVES, A. T. **Aplicação dos modelos paramétricos ARMAV e ARV na identificação modal de sistemas mecânicos**. 2006. 152 f. Tese (Doutorado em Engenharia Mecânica) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2006. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18135/tde-17012011-111718/pt-br.php>>. Acesso: 02 jan. 2015.

PASSOS, L. **Notas de aula de controle de processos industriais**. Disponível em: <[https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=12&cad=rja&uact=8&ved=0CD4QFjABOAO&url=http%3A%2F%2Fwww.portalengrenagens.com.br%2Ftdps%2F%3Fwpdmact%3Dprocess%26did%3DMTAuaG90bGluaw%3D%3D&ei=KtVfVMqE\\_HbsATNz4G4BA&usg=AFQjCNHVCFC12P1EGL8vkWFpQuAEAN7UFw&sig2=AYG\\_f3fek9xq-9b7J0BO7g](https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=12&cad=rja&uact=8&ved=0CD4QFjABOAO&url=http%3A%2F%2Fwww.portalengrenagens.com.br%2Ftdps%2F%3Fwpdmact%3Dprocess%26did%3DMTAuaG90bGluaw%3D%3D&ei=KtVfVMqE_HbsATNz4G4BA&usg=AFQjCNHVCFC12P1EGL8vkWFpQuAEAN7UFw&sig2=AYG_f3fek9xq-9b7J0BO7g)>. Acesso em: 09 nov. 2014.

RAMIRO, F. S.; COSTA, L. A. ;BERNADES J. A . Softwares educacionais – Seu uso e importância no ensino-aprendizagem dos alunos de engenharia civil. In: **COBENGE – ENGENHARIA: MÚLTIPLOS SABERES E ATUAÇÕES** Disponível em:<<http://www.abenge.org.br/cobenge-2014/Artigos/129044.pdf>>. Acesso em: 07 set 2015.

ROCHA, K. P.; PANOEIRO, N. M.; JÚNIOR, L. O. A; CARMO, M. J. Controle de sistemas não lineares: um estudo de caso na graduação. In: **COBENGE - XL**

**CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIAS.** Disponível em: <<http://www.abenge.org.br/CobengeAnteriores/2012/artigos/104270.pdf>>. Acesso em: 02 jan. 2015.

**RODRIGUES, G. G. Identificação de Sistemas Dinâmicos Não-Lineares Utilizando Modelos NARMAX Polinomiais – Aplicação a Sistemas Reais.** 2000. 152 f. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, 2000. Disponível em: <[http://www.bibliotecadigital.ufmg.br/dspace/bitstream/handle/1843/BUDB-8D3GC3/giovani\\_guimaraes\\_rodrigues.pdf?sequence=1](http://www.bibliotecadigital.ufmg.br/dspace/bitstream/handle/1843/BUDB-8D3GC3/giovani_guimaraes_rodrigues.pdf?sequence=1)>. Acesso em: 2 jan. 2015.

**SOUSA, J. P. M. Aplicação das técnicas de identificação de sistemas não-lineares à modelagem de fornos elétricos a arco.** 2005. 126 p. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós-graduação em Engenharia Elétrica, Pontifícia Universidade Católica de Minas Gerais, 2005. Disponível em: <[www.biblioteca.pucminas.br/teses/EngEletrica\\_SousaJP\\_1.pdf](http://www.biblioteca.pucminas.br/teses/EngEletrica_SousaJP_1.pdf)>. Acesso em: 2 de jan. 2015.

**VALLVERDU, M.; KORENBERG M.J; CAMINAL, P. Model Identification of the Neural Control of the cardiovascular System Using NARMAX Models – IEEE.** Paper presented at the Proc. Computer in Cardiology (1991).

## ANEXO A - ROTINA PARA GERAÇÃO DA MATRIZ GENÉRICA

```
clear
clc
syms y k u
a=2;b=2;grau = 3;
c= a+b;
m=1;
i=1;
f=c+1;
V = sym (ones(1,f));

% Cria o vetor base
for j=1:c
    if j<=a
        %V(1,j) = dado_u(atrasoX -(j-1):tam_dado_u - j);
        V(1,j) = sprintf('u(k-%d)',j);
    else
        e=j-a;
        %V(1,j) = dado_y(atrasoY -(e-1):tam_dado_y - e);
        V(1,j) = sprintf('y(k-%d)',e);
    end
end

V

% Atraso 0 em x
if a==0
    if grau==1
        R=V;
    elseif grau==2
        for j=1:f
            for j1=1:f
                R(i,m)=V(i,j)*V(i,j1);
                m=m+1;
            end
        end
    elseif grau==3
        for j=1:f
            for j1=1:f
                for j2=1:f
                    R(i,m)=V(i,j)*V(i,j1)*V(i,j2);
                    m=m+1;
                end
            end
        end
    else
        for j=1:f
            for j1=1:f
                for j2=1:f
                    for j3=1:f
                        R(i,m)=V(i,j)*V(i,j1)*V(i,j2)*V(i,j3);
                        m=m+1;
                    end
                end
            end
        end
    end
end
```

```

% atraso 1
elseif a==1
    if grau==1
        R=V;
    elseif grau==2
        for j=1:f
            for j1=1:f
                R(i,m)=V(i,j)*V(i,j1);
                m=m+1;
            end
        end
    elseif grau==3
        for j=1:f
            for j1=1:f
                for j2=1:f
                    R(i,m)=V(i,j)*V(i,j1)*V(i,j2);
                    m=m+1;
                end
            end
        end
    end
else
    for j=1:f
        for j1=1:f
            for j2=1:f
                for j3=1:f
                    R(i,m)=V(i,j)*V(i,j1)*V(i,j2)*V(i,j3);
                    m=m+1;
                end
            end
        end
    end
end

```

```

% atraso 2
elseif a==2
    if grau==1
        R=V;
    elseif grau==2
        for j=1:f
            for j1=1:f
                R(i,m)=V(i,j)*V(i,j1);
                m=m+1;
            end
        end
    elseif grau==3
        for j=1:f
            for j1=1:f
                for j2=1:f
                    R(i,m)=V(i,j)*V(i,j1)*V(i,j2);
                    m=m+1;
                end
            end
        end
    end
else
    for j=1:f
        for j1=1:f

```

```

        for j2=1:f
            for j3=1:f
                R(i,m)=V(i,j)*V(i,j1)*V(i,j2)*V(i,j3);
                m=m+1;
            end
        end
    end
end

end

end

end

% Atraso 3
elseif a==3
    if grau==1
        R=V;
    elseif grau==2
        for j=1:f
            for j1=1:f
                R(i,m)=V(i,j)*V(i,j1);
                m=m+1;
            end
        end
    elseif grau==3
        for j=1:f
            for j1=1:f
                for j2=1:f
                    R(i,m)=V(i,j)*V(i,j1)*V(i,j2);
                    m=m+1;
                end
            end
        end
    end

else
    for j=1:f
        for j1=1:f
            for j2=1:f
                for j3=1:f
                    R(i,m)=V(i,j)*V(i,j1)*V(i,j2)*V(i,j3);
                    m=m+1;
                end
            end
        end
    end

end

end

end

% Atraso 4
else
    if grau==1
        R=V;
    elseif grau==2
        for j=1:f
            for j1=1:f
                R(i,m)=V(i,j)*V(i,j1);
                m=m+1;
            end
        end
    elseif grau==3
        for j=1:f
            for j1=1:f
                for j2=1:f
                    R(i,m)=V(i,j)*V(i,j1)*V(i,j2);
                end
            end
        end
    end
end

```

```

        m=m+1;
    end
end
end
else
    for j=1:f
        for j1=1:f
            for j2=1:f
                for j3=1:f
                    R(i,m)=V(i,j)*V(i,j1)*V(i,j2)*V(i,j3);
                    m=m+1;
                end
            end
        end
    end
end

end

R
Reg=unique(R)

Comp=length(R)

```