

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CÂMPUS CURITIBA  
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

HIGOR DE SOUZA SERAFIN

**TOOLBOX DE IDENTIFICAÇÃO DE SISTEMAS  
VIA FUNÇÕES ORTONORMAIS**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2018

HIGOR DE SOUZA SERAFIN

**TOOLBOX DE IDENTIFICAÇÃO DE SISTEMAS  
VIA FUNÇÕES ORTONORMAIS**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do curso de Engenharia de Controle e Automação do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR) como requisito para obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Elder Oroski

CURITIBA  
2018

Higor de Souza Serafin

## Toolbox de Identificação de Sistemas via Funções Ortonormais

Este Trabalho de Conclusão de Curso de Graduação foi julgado e aprovado como requisito parcial para a obtenção do Título de Engenheiro de Controle e Automação, do curso de Engenharia de Controle e Automação do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR).

Curitiba, 19 de novembro de 2018.

---

Prof. Paulo Sérgio Walenia, Dr.  
Coordenador de Curso  
Engenharia de Controle e Automação

---

Profa. Annemarle Gehrke Castagna, Dr.  
Responsável pelos Trabalhos de Conclusão de Curso  
de Engenharia de Controle e Automação do DAELT

### **ORIENTAÇÃO**

---

Elder Oroski, Dr.  
Universidade Tecnológica Federal do Paraná  
Orientador

### **BANCA EXAMINADORA**

---

Elder Oroski, Dr.  
Universidade Tecnológica Federal do Paraná

---

Victor Baptista Frencl, Dr.  
Universidade Tecnológica Federal do Paraná

---

Rafael Fontes Souto, Dr.  
Instituto Federal de Educação, Ciência e Tecnologia de  
Brasília

## **AGRADECIMENTOS**

A Deus por ter me dado as condições de chegar até o final desta etapa da minha vida.

A minha família e amigos por todo apoio e amor que dedicaram-me para que eu conseguisse conquistar mais este sonho.

Ao meu amigo Gabriel Marafon que me acompanhou na primeira fase deste trabalho.

Ao professor Dr. Elder Oroski pela orientação durante estes últimos três semestres.

Ao meu amigo Matheus Kunhavalik que forneceu os dados da planta didática para a validação dos algoritmos.

## RESUMO

SERAFIN, Higor de Souza. Toolbox de Identificação de Sistemas via Funções Ortonormais, 79 p. Trabalho de Conclusão de Curso Engenharia de Controle e Automação, Universidade Tecnológica Federal do Paraná, 2018.

Este trabalho apresenta conceitos de Identificação de Sistemas por meio de funções ortonormais, métodos heurísticos e determinísticos de otimização. Com estas ferramentas, foi desenvolvida uma *toolbox* com interface gráfica para o usuário, a fim de facilitar o aprendizado destes conceitos e para que a mesma sirva como um recurso didático para auxílio ao professor. A *toolbox* foi desenvolvida em um modelo *open source*, fazendo com que a comunidade que utilizará a mesma tenha liberdade para modificá-la e, assim, incentivar a contribuição para sua melhoria. Na validação de alguns conceitos, utilizou-se dos algoritmos desenvolvidos para realizar a identificação de uma planta didática que realiza o controle de fluxo de água e aferição do teor de pH (Potencial Hidrogeniônico) do fluido. Foi realizada a comparação entre os resultados dos algoritmos e encontrada as funções que modelam o sistema. Por fim, é realizado um experimento didático com uma turma de disciplina de Introdução à Identificação de Sistemas para validar o viés didático da *toolbox*.

**Palavras-Chave:** Identificação de sistemas. Funções ortonormais. Identificação caixa preta. Otimização. Software Educacional.

## ABSTRACT

SERAFIN, Higor de Souza. Toolbox for Orthonormal Functions Systems Identification, 79 p. Trabalho de Conclusão de Curso Engenharia de Controle e Automação, Universidade Tecnológica Federal do Paraná, 2018.

This work presents concepts of System Identification using orthonormal function, deterministic and heuristic optimization methods. With these tools, a toolbox with a graphical user interface has been developed, in order to facilitate the learning process of these concepts and to be a didactic resource to help the teacher. The toolbox code will be open source causing the community that will utilize it have the freedom to modify it and encouraging the contribution for its improvement. In the validation of some concepts, optimization algorithms have been used to perform the identification of a water flux control and pH (Potential of Hydrogen) content admeasurement didactic plant. A comparison between the methods results have been made, and the functions that describe the system have been encountered. At last, a didactic experiment is carried out with a discipline class Introduction the Systems Identification to validate the didactic toolbox.

**Keywords:** Systems identification. Orthonormal functions. Black-box identification. Optimization. Educational Software.

## LISTA DE FIGURAS

FIGURA 1	– Filtro FIR. ....	25
FIGURA 2	– Filtro de Laguerre. ....	26
FIGURA 3	– Resposta ao degrau de uma função de Laguerre. ....	27
FIGURA 4	– Filtro de Kautz. ....	29
FIGURA 5	– Resposta ao degrau de um par de funções de Kautz. ....	29
FIGURA 6	– Conceito básico ACO. ....	40
FIGURA 7	– Mecanismo de busca do PSO. ....	43
FIGURA 8	– <i>Toolbox</i> desenvolvida para identificação de sistemas. ....	48
FIGURA 9	– Primeira área da <i>toolbox</i> . ....	49
FIGURA 10	– Segunda área da <i>toolbox</i> . ....	50
FIGURA 11	– Planta didática utilizada para identificação. ....	52
FIGURA 12	– Sinal aplicado e coletado para teste de linearidade. ....	53
FIGURA 13	– Espectro de frequência dos sinais de entrada e saída. ....	54
FIGURA 14	– Sinal PRMLS aplicado ao sistema. ....	54
FIGURA 15	– Sinal de saída do sistema. ....	55
FIGURA 16	– Resposta do sistema à função degrau unitário. ....	55
FIGURA 17	– Resultado da identificação com funções de Laguerre e AG. ....	59
FIGURA 18	– Resultado da identificação com funções de Laguerre e PSO. ....	61
FIGURA 19	– Comportamento do MSE com o aumento das funções da base. ....	62
FIGURA 20	– Resultado da identificação com funções de Laguerre e ACO. ....	63
FIGURA 21	– Resultado da identificação com funções de Laguerre e GD. ....	65
FIGURA 22	– Resultado da identificação com funções de Laguerre e GN. ....	65
FIGURA 23	– Resultado da identificação com funções de Laguerre e LM. ....	66
FIGURA 24	– Resultado da identificação com funções de Laguerre e o algoritmo híbrido. ....	67
FIGURA 25	– Resultado do questionário. ....	69
FIGURA 26	– Resultado para cada critério. ....	70

## LISTA DE TABELAS

TABELA 1	– Parâmetros para modelo AG. ....	58
TABELA 2	– Resultados do Algoritmo Genético. ....	59
TABELA 3	– Parâmetros para modelo PSO. ....	60
TABELA 4	– Resultados com o otimizador PSO. ....	60
TABELA 5	– Parâmetros para modelo ACO. ....	62
TABELA 6	– Resultados com o otimizador ACO. ....	63
TABELA 7	– Melhor resultado de cada método heurístico. ....	64
TABELA 8	– Parâmetros dos algoritmos determinísticos. ....	64
TABELA 9	– Resultados dos algoritmos determinísticos. ....	64
TABELA 10	– Parâmetros do algoritmo híbrido. ....	67
TABELA 11	– Resultados dos algoritmos determinísticos. ....	67
TABELA 12	– Comparativo dos resultados. ....	68

## LISTA DE SIGLAS

ACO	<i>Ant Colony Optimization</i>
AG	Algoritmo Genético
AS	<i>Ant System</i>
BIBO	<i>Bound Input Bound Output</i>
FIR	<i>Finite Impulse Response</i>
GD	Gradiente Descendente
GN	Gauss-Newton
GUI	<i>Graphical User Interface</i>
GUIDE	<i>Graphical User Interface Development Environment</i>
LM	Levenberg-Marquardt
MSE	<i>Mean Square Error</i>
OBF	<i>Orthonormal Basis Function</i>
pH	Potencial Hidrogeniônico
PSO	<i>Particle Swarm Optimization</i>
PRMLS	<i>Pseudo Random Multi Level Sequence</i>
UI	<i>User Interface</i>

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>11</b>
1.1 TEMA	11
1.1.1 Delimitação do Tema	12
1.2 PROBLEMAS E PREMISSAS	13
1.3 OBJETIVOS	14
1.3.1 Objetivo Geral	14
1.3.2 Objetivos Específicos	14
1.4 JUSTIFICATIVA	15
1.5 PROCEDIMENTOS METODOLÓGICOS	15
1.6 ESTRUTURA DO TRABALHO	16
<b>2 FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
2.1 SISTEMAS E MODELOS	17
2.2 IDENTIFICAÇÃO DE SISTEMAS	19
2.2.1 CAIXA BRANCA	19
2.2.2 CAIXA PRETA	20
2.2.3 CAIXA CINZA	20
2.3 ESPAÇO DE HILBERT	21
2.4 FUNÇÕES ORTONORMAIS	23
2.4.1 FUNÇÕES FIR	25
2.4.2 FUNÇÕES DE LAGUERRE	26
2.4.3 FUNÇÕES DE KAUTZ	28
2.5 SÍNTESE	30
<b>3 OTIMIZAÇÃO</b>	<b>31</b>
3.1 MÉTODOS DE OTIMIZAÇÃO	31
3.2 ALGORITMOS DETERMINÍSTICOS	32
3.2.1 Definições	32
3.2.2 GD - GRADIENTE DESCENDENTE	33
3.2.3 GN - GAUSS-NEWTON	34
3.2.4 LM - LEVENBERG-MARQUARDT	35
3.3 ALGORITMOS HEURÍSTICOS	36
3.3.1 AG - ALGORITMO GENÉTICO	36
3.3.1.1 <i>Fitness</i>	37
3.3.1.2 Torneio	37
3.3.1.3 <i>Crossover</i>	37
3.3.1.4 Elitismo	38
3.3.1.5 Pressão seletiva	38
3.3.1.6 Mutação	38
3.3.1.7 Estrutura do Algoritmo	38
3.3.2 ACO - <i>ANT COLONY OPTIMIZATION</i>	39
3.3.3 Definições iniciais	40
3.3.4 <i>Tour</i>	40

3.3.5	Evaporação e atualização de feromônios .....	41
3.3.6	PSO - <i>PARTICLE SWARM OPTIMIZATION</i> .....	42
3.4	ALGORITMO HÍBRIDO .....	44
3.5	SÍNTESE .....	45
<b>4</b>	<b>DESENVOLVIMENTO GRÁFICO DA <i>TOOLBOX</i></b> .....	<b>46</b>
4.1	CRITÉRIOS PEDAGÓGICOS DE DESENVOLVIMENTO .....	46
4.2	INTERFACE GRÁFICA .....	48
4.3	SÍNTESE .....	51
<b>5</b>	<b>METODOLOGIA DOS EXPERIMENTOS DE VALIDAÇÃO</b> .....	<b>52</b>
5.1	METODOLOGIA DE IDENTIFICAÇÃO DE SISTEMAS .....	52
5.2	METODOLOGIA DO EXPERIMENTO PEDAGÓGICO .....	56
5.3	SÍNTESE .....	57
<b>6</b>	<b>ANÁLISES E RESULTADOS</b> .....	<b>58</b>
6.1	RESULTADOS DE IDENTIFICAÇÃO E OTIMIZAÇÃO .....	58
6.2	RESULTADOS PEDAGÓGICOS .....	68
6.3	SÍNTESE .....	70
<b>7</b>	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>72</b>
	<b>Apêndice A – QUESTIONÁRIO APLICADO EM SALA DE AULA</b> .....	<b>74</b>
	<b>REFERÊNCIAS</b> .....	<b>76</b>

# 1 INTRODUÇÃO

## 1.1 TEMA

Em decorrência do avanço industrial nas últimas décadas, os processos de produção se tornaram cada vez mais complexos e, como consequência, as plantas industriais passaram a requerer sistemas de controle mais robustos, restringindo a atuação dos modelos em uso devido à capacidade de representação dos mesmos (CAMPELLO et al., 2007). Desta forma, fez-se necessário o desenvolvimento de técnicas que suprissem as necessidades industriais por meio de estratégias mais sofisticadas, entre elas a identificação de sistemas.

A identificação de sistemas, segundo Ljung (2010), tem como objetivo elaborar modelos matemáticos para sistemas dinâmicos com base em dados de entrada e dados de saída observados. Esta metodologia, baseada em observação de dados, possui suas raízes direcionadas para as áreas da estatística e análise de séries temporais (DEISTLER, 2002), sendo atualmente estendida aos mais diversificados ramos de estudos.

De acordo com Hsia (1997), uma das principais características das técnicas de identificação de sistemas é a modelagem de sistemas dinâmicos, que possuem propósitos bem definidos. Um modelo matemático pode variar sua complexidade dependendo de quantas propriedades do sistema deseja-se representar, ou seja, um mesmo sistema poderá ter diferentes modelagens dependendo de seu propósito.

Segundo Garcia et al. (1989), as técnicas que utilizam modelos matemáticos dinâmicos foram aquelas que apresentaram destaque em relação às demais. Nestas estratégias, o modelo desenvolvido do processo executa uma função fundamental na previsão do comportamento do sistema e, com isso, auxilia na ação do controlador. Para realizar a identificação de sistemas, segundo Brandolt et al. (2002), devem ser executadas as seguintes etapas:

- Determinar e parametrizar os modelos matemáticos que melhor descrevam o sistema em análise;
- Encontrar os parâmetros que levam o modelo a se comportar de forma semelhante ao sistema;
- Validar o modelo e avaliar se o resultado satisfaz às expectativas finais.

### 1.1.1 Delimitação do Tema

Este trabalho discorre sobre técnicas de identificação de sistemas dotadas de estrutura OBF (*Orthonormal Basis Function*), bem como otimização dos modelos obtidos através de estratégias determinísticas e heurísticas. As sequências de funções ortonormais foram inicialmente desenvolvidas na década de 1920 e publicadas nos trabalhos de Takenaka (1925) e Malmquist (1925). Na década de 1950, uma representação destas funções no espaço contínuo foi apresentado em Kautz (1954) com aplicação de síntese de redes. Contudo, aplicações em identificação de sistemas apareceram, com maior relevância, nos trabalhos de Ninness e Gustafsson (1994) e Wahlberg (1991) apenas no final do século XX.

As aplicações das OBFs vão além da área de identificação de sistemas. Segundo Heuberger et al. (2003), diversos problemas que envolvem teoria de circuitos, telecomunicações, estimação, processamento de sinais e teorias de controle podem ser representados ou parametrizados, de forma eficiente, segundo uma classe de sistemas que usam as OBFs. De acordo com estes autores, com o conhecimento prévio de um sistema ou sinal, este pode ser descrito através de sua decomposição em componentes ortonormais, o que leva a um algoritmo de predição ou estimação mais robusto.

Em decorrência das características das OBFs, passou-se a ter um maior desenvolvimento de esquemas de estimação que fazem uso das mesmas. O modelo mais comum a ser empregado é o modelo FIR (*Finite Impulse Response*). De acordo com Wahlberg (1991), os modelos FIR são estruturas que implicam em expandir o sistema em uma função de transferência discreta com os polos localizados na origem. Entretanto, se o sistema for de primeira ordem e apresentar polos próximos do círculo de raio unitário, portanto, polos lentos, é mais indicado que se utilizem as estruturas do modelo de Laguerre.

Em casos que existe um conhecimento prévio do comportamento do sistema e se ele for de segunda ordem com polos complexos, Ninness et al. (1995) indica que o modelo mais apropriado, dentre as OBFs, seja o de Kautz. Para os autores mencionados, o modelo FIR é

um caso específico do modelo de Laguerre quando os polos estão localizados na origem, já as funções de Laguerre são casos particulares dos modelos de Kautz quando os polos são reais.

Aliado às técnicas de identificação, são utilizadas técnicas de otimização, que, segundo Wallin (2004), têm como objetivo resolver o problema obtido da melhor maneira possível. Existem várias maneiras de se classificar problemas de otimização: linear e não linear, determinístico e estocástico, discreto e contínuo entre outras. Nos trabalhos mais recentes, os pesquisadores costumam utilizar-se da classificação de convexo e não convexo, pois, dentre outras características, a mesma garante uma solução ótima ou garante que a mesma não exista (WALLIN, 2004). A garantia fornecida pelas funções convexas é de grande valor, porém as mesmas vêm ao custo de uma elevada complexidade teórica e matemática, fugindo do objetivo deste trabalho.

## 1.2 PROBLEMAS E PREMISSAS

Desenvolver modelos matemáticos de processos, baseados no seu comportamento físico, pode se tornar uma tarefa bastante árdua, à medida em que os sistemas se tornam mais complexos (AGUIRRE, 2007). Para sanar este problema, foram desenvolvidos métodos de identificação de sistemas, os quais podem ser divididos em três grupos: caixa branca, baseados apenas na física do processo; caixa preta, que utilizam apenas os dados coletados do sistema para gerar o modelo; e caixa cinza, que combinam as técnicas dos dois outros métodos.

Com o auxílio do *software* MATLAB<sup>®</sup>, desenvolveu-se uma *toolbox* com uma GUI (*Graphical User Interface*) que permita ao usuário final a identificação de diferentes sistemas apenas com a inserção dos dados coletados de sua planta. Este também poderá escolher com quais técnicas de identificação e otimização trabalhar, sendo possível ainda a manipulação de parâmetros nestas técnicas com o objetivo de observar o comportamento das funções. Esta *toolbox* difere da existente no MATLAB<sup>®</sup> porque esta fará o uso das OBFs.

Neste trabalho, serão apresentados alguns métodos de identificação que fazem uso das técnicas caixa preta. Para este desenvolvimento, será dada exclusividade para as metodologias que fazem a identificação por meio das funções com bases ortonormais. Os modelos que serão abordados são: FIR, Laguerre e Kautz. Os sistemas abordados serão paramétricos, invariantes no tempo e possuem apenas uma entrada e uma saída, podendo ou não serem lineares. As técnicas de otimização neste trabalho serão classificadas como determinísticas e heurísticas, pois esta divisão possui uma base sólida e uma teoria com um nível de complexidade mais adequado ao público alvo, do ponto de vista didático.

## 1.3 OBJETIVOS

### 1.3.1 Objetivo Geral

Desenvolver uma *toolbox* no *software* MATLAB<sup>®</sup> que permita a identificação de sistemas por meio de funções ortonormais, bem como a otimização dos modelos obtidos com finalidade didática.

### 1.3.2 Objetivos Específicos

- Estudar as funções ortonormais:
  - FIR;
  - Laguerre;
  - Kautz;
- Estudar as técnicas de otimização heurísticas:
  - Algoritmo Genético (AG);
  - *Particle Swarm Optimization* (PSO);
  - *Ant Colony Optimization* (ACO);
- Estudar as técnicas de otimização determinísticas:
  - Gauss-Newton (GN);
  - Gradiente Descendente (GD);
  - Levenberg-Marquardt (LM);
- Combinar métodos de otimização heurísticos e determinísticos;
- Implementar algoritmos no MATLAB<sup>®</sup> que correspondam ao comportamento de cada função ortonormal citada;
- Desenvolver uma *toolbox*, em ambiente MATLAB<sup>®</sup>, com interface intuitiva para o usuário final;
- Testar o desempenho da *toolbox* com alunos da disciplina de Introdução à Identificação de Sistemas.

## 1.4 JUSTIFICATIVA

A utilização de técnicas de identificação de sistemas apresentam aplicações em diversos campos, como nas áreas industriais, meteorológicas, econômicas, biológicas, entre outras. O desenvolvimento de uma *toolbox* que permita ao usuário obter um modelo matemático que descreva um fenômeno ou processo que o mesmo estuda, possibilita a minimização do esforço para obtenção de resultados. Esta ferramenta também pode fornecer aos professores de disciplinas, como Introdução à Identificação de Sistemas, um artifício que facilite a compreensão e aplicação das técnicas apresentadas em sala de aula.

Como já exposto, existem diferentes técnicas que conseguem modelar sistemas, contudo, neste trabalho, escolheu-se utilizar métodos que utilizam funções ortonormais, pois, visando também a modelagem de sistemas não lineares, Campello et al. (2007) afirmam que a obtenção de um modelo equivalente, técnicas que utilizam redes neurais, modelos Volterra ou modelos *Fuzzy*, requerem um maior custo computacional.

Segundo Ninness et al. (1995), outras vantagens que podem ser apresentadas no uso de OBFs para identificação de sistemas são as estruturas lineares em relação a seus parâmetros e, no caso em que se conhece previamente o comportamento do modelo, pode se ter uma estimação com poucos parâmetros, além do conhecimento e facilidade de análise teórica que induz a diretrizes práticas de implementações dos modelos.

## 1.5 PROCEDIMENTOS METODOLÓGICOS

O desenvolvimento do trabalho acontece em primeiro momento por meio do estudo das funções ortonormais utilizadas no modelo FIR, Laguerre e Kautz, para que se possa ter uma melhor compreensão do funcionamento desta ferramenta na identificação de sistemas. Em paralelo a isto, é realizado o estudo dos métodos de otimização heurísticos e determinísticos.

Na segunda etapa do trabalho, buscam-se formas de implementar os conceitos estudados anteriormente na plataforma MATLAB<sup>®</sup>. Com os algoritmos finalizados, busca-se associar os códigos gerados com as janelas GUI do mesmo, concluindo a *toolbox* proposta neste trabalho. Nesta etapa, são realizados testes comparativos com algoritmos cuja eficácia é conhecida para a validação dos algoritmos implementados.

Com as duas etapas anteriores concluídas, será testado didaticamente através de um experimento em sala de aula na disciplina de Introdução à Identificação de Sistemas

da Universidade Tecnológica Federal do Paraná, Campus Curitiba, se a ferramenta de fato auxilia no entendimento dos conceitos estudados no curso. Serão utilizados como critérios a intuitividade da interface, o desempenho do suporte presente na *toolbox*, a satisfação do aluno e a eficiência da ferramenta.

## 1.6 ESTRUTURA DO TRABALHO

Este trabalho segue a seguinte estrutura, o Capítulo 2 apresenta uma fundamentação teórica sobre as funções ortonormais e alguns conceitos complementares, para que se possa compreender melhor sobre este assunto. O Capítulo 3 versa sobre as ferramentas de otimização e como elas são estruturadas. Dentro do Capítulo 4, são apresentados os critérios para o desenvolvimento da interface da *toolbox* e suas funcionalidades são expostas. No Capítulo 5, são apresentadas as metodologias utilizadas para a validação dos algoritmos desenvolvidos e o experimento pedagógico realizado com os alunos em sala de aula. No Capítulo 6, são apresentados e discutidos os resultados alcançados. Para finalizar, no Capítulo 7 são feitas algumas considerações finais sobre o trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados alguns conceitos importantes para o desenvolvimento do trabalho. Na Seção 2.1, é discorrido sobre Sistemas e Modelos. A Seção 2.2 apresenta o conceito de Identificação de Sistemas e algumas das metodologias mais comuns empregadas nesta área. Na Seção 2.3, são expostas algumas noções sobre o Espaço de Hilbert e, em seguida, a Seção 2.4 apresenta as Funções Ortonormais, sendo elas FIR, Laguerre e Kautz.

### 2.1 SISTEMAS E MODELOS

A definição de **sistemas** é razoavelmente homogênea entre autores como Ljung (1999), Aguirre (2007), Ogata (1998) e Lathi (2007), dos quais pode se inferir que um sistema é uma estrutura com capacidade de processar um determinado conjunto de dados (entradas) que tenha como resposta um outro conjunto de dados (saídas). Um sistema pode ser estruturado a partir de componentes físicos, elétricos, hidráulicos, mecânicos ou até mesmo por algoritmos computacionais. Ogata (1998) estende o conceito de sistemas para fenômenos abstratos e dinâmicos que atuam em função de um certo objetivo. Como consequência, o termo *sistemas* pode ser atribuído a sistemas físicos, econômicos, biológicos, entre outros.

Os sistemas abordados neste trabalho serão os sistemas dinâmicos, lineares, causais e invariantes no tempo, que apresentam como característica principal o fato de que o sinal de saída, em qualquer instante, depende apenas dos sinais passados e atuais de entrada, ou seja, um sistema assim definido, não apresenta uma resposta antes de ser efetuada uma excitação (LATHI, 2007; LJUNG, 1999).

Para Lathi (2007), o estudo de sistemas constitui-se de três áreas: modelagem matemática, análise e projeto. O desenvolvimento deste trabalho está focado basicamente na primeira área, a modelagem matemática dos sistemas. Um modelo de um sistema dinâmico pode ser definido como sendo uma família de equações que expressam o comportamento

(a dinâmica) do sistema com precisão ou dentro de uma faixa aceitável de erro (OGATA, 1998). Uma observação importante a ser feita neste ponto é que um mesmo sistema pode ser representado por diferentes modelos. Esta característica será mostrada no decorrer do trabalho.

O fato de um sistema possuir diferentes representações não deve ser encarado como um problema, pelo contrário, cada modelo apresenta suas particularidades em representar as características do sistema e o custo de implementação também é variável entre os modelos. Um exemplo a ser mencionado são as aplicações em controle ótimo, nas quais trabalhar com representações em espaço de estado tem um custo computacionalmente menor. Entretanto, se desejar estudar a resposta transitória de um sistema linear e invariante no tempo, a modelagem por meio de funções de transferência é muito mais conveniente (OGATA, 1998).

Outros fatores importantes para serem observados na escolha e elaboração de um modelo é a simplicidade e a precisão do mesmo. Estas duas características implicam, principalmente, na complexidade do modelo (OGATA, 1998). Representar um sistema de forma precisa pode significar ter centenas de equações. Por outro lado, obter um modelo simplificado pode significar uma fraca representação do sistema. Sendo assim, faz-se necessário estabelecer um critério, qualitativo ou quantitativo, entre a precisão e simplicidade.

Ljung (1999) menciona que para desenvolver um modelo existem basicamente dois métodos diferentes, ou uma combinação dos mesmos. A primeira forma de se obter um modelo é dividir o sistema em subsistemas, dos quais se conhecem as propriedades de experimentos passados ou de formulações matemáticas e físicas estabelecidas. Cada subsistema recebe um tratamento matemático adequado e o modelo do sistema completo é formulado com a junção das partes. Este método implica em um conhecimento empírico e não depende obrigatoriamente de uma experimentação do sistema atual. O outro método que o autor menciona é oriundo de técnicas de experimentação. Basicamente, o modelo é inferido através da observação, tratamento e análise dos dados de entrada e saída do sistema em questão. Este caminho pode ser denominado como **Identificação de Sistema**. A Seção 2.2 traz uma abordagem mais detalhada sobre este assunto.

## 2.2 IDENTIFICAÇÃO DE SISTEMAS

A Identificação de Sistemas é definida por Aguirre (2007) como uma área do conhecimento que busca estudar caminhos para modelar e analisar sistemas por meio da observação, ou seja, dos dados extraídos desses sistemas. Esta definição vai de encontro com a apresentada no parágrafo anterior. Segundo este autor, nas últimas décadas, tem-se observado uma tendência que irá tornar o uso das técnicas de identificação e análise de sistemas necessárias nas mais diversas áreas do conhecimento humano.

Um sistema dinâmico pode ser representado por *modelos teóricos* e/ou *modelos experimentais* (ISERMANN; MÜNCHHOF, 2011). Estes modelos podem ser complementares um ao outro e a escolha de qual usar para representar um sistema depende do propósito da aplicação. Os modelos teóricos possuem uma interligação entre as propriedades físicas e os parâmetros do sistema (ISERMANN; MÜNCHHOF, 2011; CORRÊA; AGUIRRE, 2004). Sendo assim, são mais indicados para sistemas que já foram otimizados e seu comportamento dinâmico ou estático já é conhecido. Em contrapartida, os modelos experimentais contêm apenas parâmetros que possuem uma relação desconhecida com as propriedades do sistema (ISERMANN; MÜNCHHOF, 2011).

Dentro desta divisão de modelos, há três possíveis grupos de acordo com a metodologia empregada, sendo elas: *caixa-branca*, *caixa-cinza* e *caixa-preta* (AGUIRRE, 2007; OROSKI, 2015; ISERMANN; MÜNCHHOF, 2011). Na sequência, é realizado uma melhor abordagem de cada metodologia.

### 2.2.1 CAIXA BRANCA

As técnicas que usam a *modelagem caixa branca* são aquelas que Isermann e Münchhof (2011) classificam como modelos *teóricos*, porque se baseiam nas leis físicas que descrevem o sistema. Nestes casos, os parâmetros ou são conhecidos ou são determinados previamente. Esta metodologia requer conhecimento do sistema a ser modelado e tempo para que seja feito um equacionamento adequado. Os modelos obtidos normalmente são deduzidos por meio de equações diferenciais ordinárias, por serem mais simples. Entretanto, isto dificulta a representação de sistemas com parâmetros distribuídos, que requerem uma modelagem com equações diferenciais parciais (ISERMANN; MÜNCHHOF, 2011).

Outro contraponto exposto por Corrêa e Aguirre (2004) é que, em sistemas mais complexos, a obtenção dos parâmetros muitas vezes não são possíveis. Desta forma,

a Identificação de Sistemas busca técnicas que possam substituir a modelagem caixa branca (AGUIRRE, 2010). Uma das propriedades destas técnicas alternativas é requerer pouco ou nenhum conhecimento prévio do sistema estudado. Estes métodos são conhecidos como *modelagem caixa preta* e são melhor discutidos na sequência.

### 2.2.2 CAIXA PRETA

As técnicas de *identificação caixa preta* são aquelas que Isermann e Münchhof (2011) denominam como *experimentais*. Estas técnicas são motivadas pela dificuldade de se encontrar as equações que regem um sistema ou quando são conhecidas, elas são impraticáveis em decorrência de algumas limitações, como recursos e tempo (AGUIRRE, 2010).

Nesta metodologia, são utilizados apenas os dados de entrada e de saída dos sistemas. A escolha de qual estrutura será usada, normalmente é feita de forma empírica, nos casos mais simples. Tais métodos se baseiam em técnicas de álgebra linear e em conceitos de estatística (CORRÊA; AGUIRRE, 2004). Na busca pelos parâmetros, normalmente não existe uma relação clara entre a estrutura e os parâmetros com os aspectos físicos do sistema (CORRÊA; AGUIRRE, 2004).

De acordo com Isermann e Münchhof (2011), uma das vantagens destas técnicas é que uma mesma análise experimental pode ser empregada para diferentes sistemas complexos. Corrêa e Aguirre (2004) ainda reforçam que os modelos caixa preta podem ser obtidos com uma maior facilidade e que o sucesso da modelagem depende fortemente da qualidade dos dados utilizados.

Corrêa e Aguirre (2004) citam que uma desvantagem destas técnicas é que, pelo fato do modelo não possuir um significado físico, pode existir uma dificuldade para selecionar ou obter parâmetros além do necessário. É importante ressaltar que neste trabalho se utilizará técnicas que fazem uso da modelagem caixa preta.

### 2.2.3 CAIXA CINZA

A rigor, qualquer técnica que não esteja enquadrada dentro das duas metodologias apresentadas anteriormente pode ser enquadrada dentro da *modelagem caixa cinza* (CORRÊA; AGUIRRE, 2004). As técnicas que compõem este grupo são caracterizadas por utilizar informações auxiliares que não estão presentes no conjunto de dados dinâmicos utilizados para a identificação (AGUIRRE, 2010).

A modelagem caixa cinza busca mesclar as vantagens das técnicas de identificação caixa preta e caixa branca (CORRÊA; AGUIRRE, 2004). Cabe ressaltar que estes modelos também apresentam desvantagens, por exemplo, em uma situação na qual as informações auxiliares forem conflitantes com os dados amostrados, pode-se resultar em um modelo com baixa qualidade. Desta forma, Aguirre (2010) menciona que os modelos que utilizam técnicas caixa cinza são uns dos grandes desafios atuais da identificação de sistemas.

Com esta contextualização feita sobre Identificação de Sistemas, a próxima seção apresentará alguns conceitos matemáticos que fazem parte do embasamento teórico deste trabalho, para se poder prosseguir às funções ortonormais.

### 2.3 ESPAÇO DE HILBERT

Nesta seção, é apresentado o espaço de Hilbert, algumas definições e propriedades. É importante ressaltar desde já que, por não ser o elemento central deste trabalho, muitas deduções não serão apresentadas, apenas referenciadas. Desta forma, para se chegar a definição de Espaço de Hilbert, é necessário a abordagem de algumas definições preliminares.

**Definição 1.** *Seja  $J$  um  $\mathbb{C}$ -espaço vetorial. A norma em  $J$  é uma aplicação em que*

$$\|\cdot\| : J \longrightarrow \mathbb{C},$$

$$z \longmapsto \|z\|,$$

*conhecido como norma de  $z$ , sendo que  $\forall z, w \in J$  e  $\forall \lambda \in \mathbb{C}$ , as seguintes condições são satisfeitas:*

$$(i) \|z\| \geq 0 \text{ e } \|z\| = 0 \Leftrightarrow z = 0;$$

$$(ii) \|\lambda z\| = |\lambda| \cdot \|z\|;$$

$$(iii) \|z + w\| \leq \|z\| + \|w\|.$$

**Definição 2.** *Seja  $J$  um  $\mathbb{C}$ -espaço vetorial. O produto interno em  $J$  é dado como uma aplicação  $\langle \cdot, \cdot \rangle : J \times J \longrightarrow \mathbb{C}$ , que satisfaz as seguintes condições:*

$$(i) \langle z + w, u \rangle = \langle z, u \rangle + \langle w, u \rangle, \forall u, z, w \in J;$$

$$(ii) \langle \lambda z, w \rangle = \lambda \langle z, w \rangle, \forall \lambda \in \mathbb{C}, \forall z, w \in J;$$

$$(iii) \langle z, z \rangle > 0, \text{ se } z \neq 0.$$

**Proposição 1.** *Se uma sequência converge, ela é chamada de sequência de Cauchy.*

**Definição 3.** *Um espaço vetorial métrico é completo se todas as seqüências de Cauchy, neste espaço, forem convergentes.*

Por sua vez,

**Proposição 2.** *Todo espaço vetorial  $V$ , normado, é métrico.*

A demonstração das Proposições 1 e 2 são desenvolvidas em Santos (2008). Com estas definições é possível definir o que é o espaço de Hilbert.

**Definição 4.** *O espaço de Hilbert,  $\mathcal{H}$ , é definido como um espaço vetorial com norma e completo, no qual a norma é definida por meio de um produto interno neste espaço.*

Outras características importantes do espaço de Hilbert a serem ressaltadas, é a ortogonalidade e ortonormalidade, que são enunciadas a seguir.

**Definição 5.** *Sejam  $v, w \in E$ , no qual  $E$  é definido como um espaço vetorial com produto interno. diz-se que  $v$  e  $w$  são ortogonais se  $\langle v, w \rangle = 0$ .*

**Definição 6.** *Sendo  $E$  definido como um espaço com produto interno, diz-se que um subconjunto  $X \subset E$  é ortonormal se, e somente se, for ortogonal e  $\forall u \in E$ ,  $u$  é unitário.*

**Definição 7.** *Se  $E$  for um espaço com produto interno, dimensão  $n$ , e  $\{u_1, \dots, u_n\}$  forma um conjunto ortonormal, então  $\{u_1, \dots, u_n\}$  é uma base ortonormal de  $E$ .*

A importância destas características é que, segundo Santos (2008), as bases ortonormais podem ser utilizadas para decompor vetores e que, no espaço de Hilbert, os elementos podem ser aproximados por elementos destas bases. Outras propriedades de Hilbert, importantes para este trabalho, são evidenciadas abaixo:

**Teorema 1.** *Para todo espaço de Hilbert existe uma base ortonormal.*

No trabalho desenvolvido por Santos (2008), é feita a demonstração do Teorema 1.

**Definição 8.** *Seja  $\{v_\alpha\}_{\alpha \in J}$  um conjunto ortonormal em  $\mathcal{H}$ , a família  $\langle v_\alpha, v \rangle_{\alpha \in J}$  é denominada coeficientes de Fourier de  $v \in \mathcal{H}$ , e o somatório  $\sum_{\alpha \in J} \langle v_\alpha, v \rangle v_\alpha$  é denominado de série de Fourier de  $v$  em relação à  $\{v_\alpha\}_{\alpha \in J}$ .*

**Teorema 2.** *Se  $\{v_\alpha\}_{\alpha \in J}$  for um conjunto ortonormal em  $\mathcal{H}$ , então as seguintes afirmações podem ser feitas:*

(i)  $\{v_\alpha\}_{\alpha \in J}$  constitui uma base ortonormal do espaço de Hilbert;

(ii) Se  $v$  pertence a Hilbert, a série de Fourier de  $v$ , em relação à  $\{v_\alpha\}_{\alpha \in J}$ , irá convergir em Hilbert para  $v$ .

O Teorema 2 é demonstrado com detalhes em Oliveira (2008). Com estas definições firmadas, torna-se possível prosseguir para a próxima seção, a qual descreve as funções ortonormais, que fazem o uso da base matemática apresentada.

## 2.4 FUNÇÕES ORTONORMAIS

De acordo com Strang (2010), as Funções Ortonormais podem ser entendidas como vetores em um determinado espaço vetorial de funções, que possuem produto interno nulo entre si e norma unitária. Este produto interno é definido em Wahlberg (1999) como sendo:

$$\langle \phi_p(z), \phi_q(z) \rangle = \frac{1}{2\pi i} \oint_C \phi_p(z) \phi_q(z)^* \left( \frac{1}{z^*} \right) \frac{dz}{z}, \quad (1)$$

em que  $\phi_p$  e  $\phi_q$  são funções ortonormais,  $C$  é o círculo de raio unitário,  $i$  é a unidade imaginária,  $p, q \in \mathbb{N}$ ,  $z \in \mathbb{C}$  e  $*$  é o operador conjugado. Em sua forma discreta, este produto interno pode ser representado por meio da equação (2)

$$\langle \phi_p(k), \phi_q(k) \rangle = \sum_{k=0}^{\infty} \phi_p(k) \phi_q(k). \quad (2)$$

A partir disto, pode-se definir a sua norma dada pela equação (3).

$$\|\phi_n(k)\| = \sqrt{\langle \phi_n(k), \phi_n(k) \rangle}. \quad (3)$$

Desta forma, as funções ortonormais pertencem ao espaço de Hilbert relacionado às funções quadraticamente somáveis, também chamado de espaço  $L^2$  (CAMPELLO et al., 2007), ou seja, ao conjunto de funções que respeitam a condição expressa em (4)

$$\sum_{k=0}^{\infty} \phi_k(k)^2 \leq \infty. \quad (4)$$

Em decorrência deste fato, todas as definições, teoremas e proposições enunciadas na Seção 2.3 são aplicáveis às funções ortonormais. Outra implicação bastante importante para identificação de sistemas lineares, enunciada por Campello et al. (2007) e em Lemma et al. (2010), é que, dada uma função  $h(k) : \mathbb{N} \rightarrow \mathbb{R}$  quadraticamente somável em  $[0, \infty)$ , irá existir um número inteiro  $n > 0$  que, para qualquer erro ( $e$ ) maior que zero tem-se

$$\sum_{k=0}^{\infty} \left( h(k) - \sum_{i=1}^n c_i \phi_i(k) \right)^2 < e, \quad (5)$$

no qual  $\phi_1(k), \dots, \phi_n(k)$  são as  $n$  funções ortonormais da base e  $c_1, \dots, c_n$  são ganhos escalares e reais. Desta forma, pode-se mostrar que, para qualquer  $k$ , a equação (6) converge à função original para um  $n$  infinito.

$$\hat{h}(k) = \sum_{i=1}^n c_i \phi_i(k). \quad (6)$$

Com a equação (6), pode-se encontrar os coeficientes da série. O desenvolvimento matemático para esta etapa é encontrado nos trabalhos de Campello et al. (2007) e Oroski (2015). A expressão obtida é a seguinte

$$c_j = \sum_{k=0}^{\infty} h(k) \phi_j(k). \quad (7)$$

Para sistemas dinâmicos lineares, a representação por meio de OBF consiste em desenvolver a resposta ao impulso com uma base de funções ortonormais. Isto é possível apenas para sistemas BIBO (*Bounded Input Bounded Output*) estáveis, pois a sua resposta ao impulso é absolutamente e, por consequência, quadraticamente somável (CAMPELLO et al., 2007). Para sistemas não lineares, a representação com OBF é possível se for feito um mapeamento estático não linear e, assim, o modelo pode ser representado através de uma dinâmica linear, ou seja, um modelo de *Wiener* (CAMPELLO; OLIVEIRA, 2007). Alguns operadores para este mapeamento são OBF-Volterra e OBF-Takagi Sugeno (CAMPELLO et al., 2007).

Existem diversas funções ortonormais, que podem ser definidas de acordo com produto interno utilizado. As mais relevantes são as funções de Kautz, Laguerre, Jacobi e as Funções Ortonormais Generalizadas (BELT, 1997). Nas subseções a seguir, serão apresentadas as três funções ortonormais propostas no início deste trabalho.

### 2.4.1 FUNÇÕES FIR

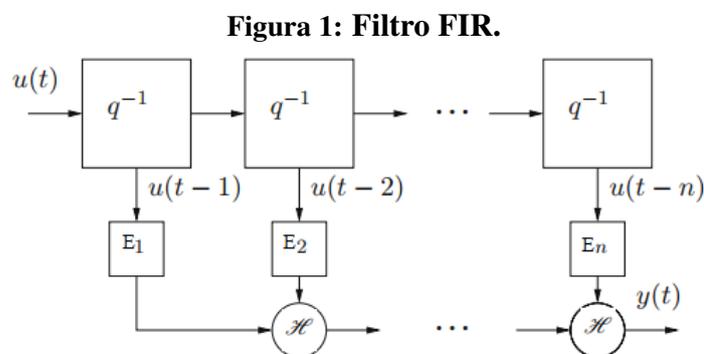
Para sistemas lineares, a função de transferência pode ser parametrizada em função da resposta ao impulso  $h_k$  do sistema (HEUBERGER et al., 2005), como mostrado na equação (8).

$$G(z) = \sum_{k=1}^{\infty} h_k z^{(-k)}. \quad (8)$$

Nota-se que haverá pelo menos um atraso entre a entrada e a saída do sistema. Esta estrutura também é chamada de expansão de Laurent e foi amplamente empregada nas décadas anteriores (OROSKI, 2015).

De acordo com Heuberger et al. (2005) para se obter o modelo linear, é necessário truncar o somatório da equação (8) em um  $k$  finito e o erro da estimação tende a zero à medida que  $k$  tende a infinito. Os autores também mencionam que, para sistemas com  $|a|$  próximo da origem, sendo que  $a$  representa os polos do sistema, um modelo FIR com uma ordem baixa consegue representar o sistema de forma satisfatória. Em contrapartida, em um sistema em que  $|a|$  está próximo do círculo de raio unitário, o somatório expresso em (8) converge lentamente para o sistema, requerendo uma elevada ordem do modelo para se obter uma aproximação adequada.

Segundo Heuberger et al. (2005), o modelo FIR também pode ser encarado como um filtro digital. Para uma melhor compreensão do funcionamento desta estrutura, mostra-se de forma esquemática, na Figura 1, um modelo do filtro FIR, em que  $q^{-1}$  é um operador de atraso,  $E_n$  são os ganhos e  $u(t)$  a entrada.



FONTE: Adaptado de Heuberger et al. (2005).

## 2.4.2 FUNÇÕES DE LAGUERRE

As funções de Laguerre se comportam como uma concatenação de primeira ordem, parametrizadas com um polo real (LEMMA et al., 2010). Estas funções podem ser expressas por meio da equação (9).

$$L_k(z) = \frac{\sqrt{1-p^2}}{z-p} \left( \frac{1-pz}{z-p} \right)^{(n-1)}, \quad (9)$$

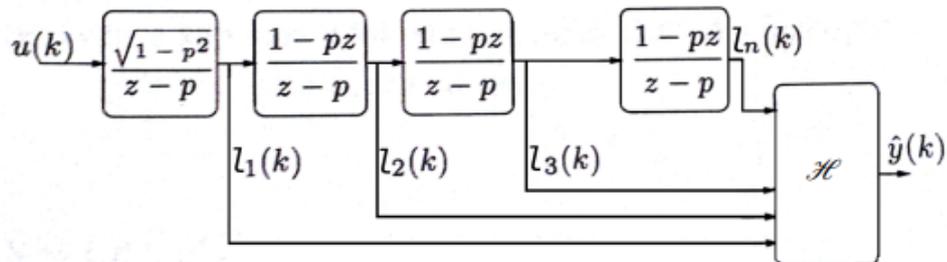
no qual  $z \in \mathbb{C}$  e  $p \in \mathbb{R}$  é o polo, e ele precisa respeitar a seguinte condição  $|p| < 1$  (LEMMA et al., 2010).

Como as funções de Laguerre são funções ortonormais, elas podem ser usadas para modelar sistemas estáveis (LEMMA et al., 2010; WAHLBERG, 1991; HEUBERGER et al., 2005; FU; DUMONT, 1993). Teoricamente, qualquer sistema linear pode ser descrito por meio de uma expansão infinita da série de Laguerre mostrada na equação (10) (FU; DUMONT, 1993). Contudo, na prática, faz-se necessário um truncamento nesta série. A escolha do melhor ponto para fazer este truncamento não será abordado neste trabalho, porém, este assunto é bem desenvolvido em Fu e Dumont (1993).

$$G(z) = \sum_{k=1}^{\infty} g_k L_k(z). \quad (10)$$

Na equação (10),  $G(z)$  representa a função de transferência de um sistema linear, em que  $g_k$  são ganhos escalares e  $L_k(z)$  são as funções de Laguerre expressas na equação (9). As funções de Laguerre também podem ser vistas como um filtro digital. Sua estrutura é exibida na Figura 2.

**Figura 2: Filtro de Laguerre.**



FONTE: Adaptado de Campello et al. (2007).

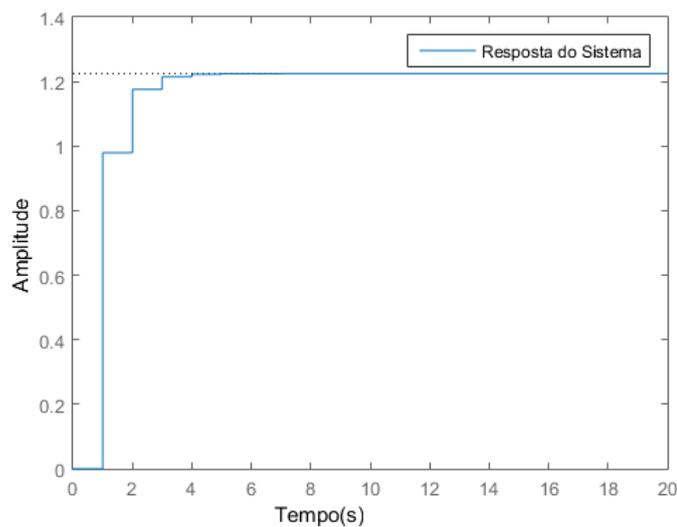
Para sistemas lineares e invariantes no tempo, o operador do espaço de Hilbert pode ser definido como sendo a combinação linear das  $k$  saídas dos filtros de Laguerre, como mostrado na equação (11) (OROSKI, 2015). Entretanto, para sistemas não lineares, é necessário uma série de combinações não lineares entre as saídas do filtro para se conseguir uma representação adequada (OROSKI; BAUCHSPIESS, 2014).

$$\hat{y}(k) = \sum_{i=1}^n c_i l_i(k), \quad (11)$$

em que  $\hat{y}$  representa a saída estimada,  $c_i$  são os ganhos e  $l_i(k)$  são as entradas  $u(k)$  filtradas pelas funções de Laguerre.

Segundo Wahlberg (1991), as OBFs permitem que seja incorporado o conhecimento *a priori* do sistema. Sendo assim, a base de Laguerre é adequada para modelar sistemas que possuem polos puramente reais, ou que possuam uma parte imaginária pouco significativa (MACHADO; AMARAL, 2010). Isto decorre do fato destas funções serem parametrizadas por polos reais e, desta forma, já respondem naturalmente como os sistemas que possuem os tipos de polos mencionados. Na Figura 3, é mostrada a resposta ao degrau de uma função de Laguerre.

**Figura 3: Resposta ao degrau de uma função de Laguerre.**



FONTE: Autoria Própria.

### 2.4.3 FUNÇÕES DE KAUTZ

As funções de Kautz são tratadas por Wahlberg (1994) como uma generalização das funções de Laguerre. A diferença principal entre estas duas funções é que as funções de Kautz são parametrizadas por um par de polos complexos conjugados (LEMMA et al., 2010; WAHLBERG, 1994), e são expressas por meio das equações (12) e (13).

$$K_{2m}(z) = \frac{\sqrt{(1-c^2)(1-b^2)}}{z^2 + b(c-1)z - c} \left( \frac{-cz^2 + b(c-1)z + 1}{z^2 + b(c-1)z - c} \right)^{(m-1)}, \quad (12)$$

$$K_{2m-1}(z) = \frac{(z-b)\sqrt{1-c^2}}{z^2 + b(c-1)z - c} \left( \frac{-cz^2 + b(c-1)z + 1}{z^2 + b(c-1)z - c} \right)^{(m-1)}. \quad (13)$$

Neste contexto,  $-1 < b < 1$  e  $-1 < c < 1$ , e correspondem, respectivamente, aos termos mostrados nas equações (14) e (15). Os fatores  $\bar{p}$  e  $p$  representam os polos conjugados complexos que parametrizam as funções e  $m \in \mathbb{N}$ . Já  $K_{2m}$  e  $K_{2m-1}$ , são as funções de Kautz pares e ímpares, respectivamente.

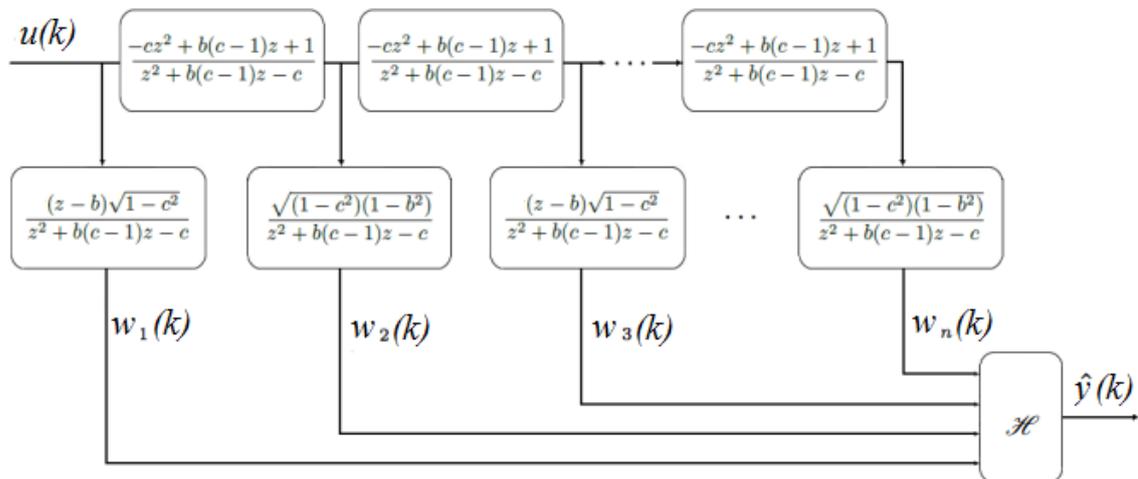
$$b = \frac{(p + \bar{p})}{1 + p\bar{p}}, \quad (14)$$

$$c = -p\bar{p}. \quad (15)$$

As bases de Kautz também compõem o espaço das funções  $L^2$  (OROSKI, 2015). Desta forma, também podem ser expressas na forma de um filtro digital, como mostra a Figura 4. Neste contexto, as funções de Kautz podem representar a dinâmica de sistemas lineares por meio da equação (16), em que  $w_i$  são as entradas  $u(k)$  filtradas pelas funções de Kautz,  $c_i$  representam os ganhos e  $\hat{y}$  é a saída estimada.

$$\hat{y} = \sum_{i=1}^n c_i w_i(k). \quad (16)$$

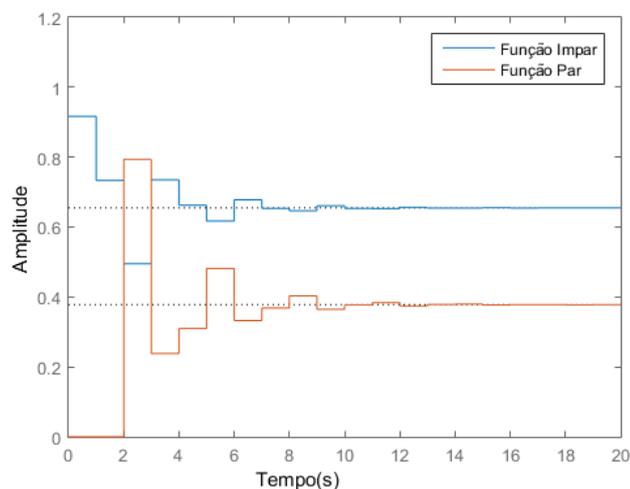
**Figura 4: Filtro de Kautz.**



FONTE: Adaptado de Oroski e Bauchspiess (2014)

Uma das vantagens que as funções de Kautz apresentam, em relação a Laguerre, é que esta última não representa muito bem a dinâmica de sistemas subamortecidos já que são parametrizadas por polos reais (WAHLBERG; MÄKILÄ, 1996). Para conseguir resultados comparáveis aos de Kautz, é necessário um maior número de funções (CAMPELLO et al., 2007). O desempenho das funções de Kautz na modelagem de sistemas subamortecidos é decorrente, principalmente, por estas possuírem em si o comportamento subamortecido devido serem parametrizadas por um par de polos complexo. A Figura 5 mostra a resposta ao degrau de um par de funções de Kautz.

**Figura 5: Resposta ao degrau de um par de funções de Kautz.**



FONTE: Autoria Própria.

Para a modelagem de sistemas não lineares, assim como em Laguerre, é necessário a combinação não linear de parâmetros. Em Oroski e Bauchspiess (2014) é feita a dedução de uma das formas de representar sistemas não lineares, e esta é expressa na equação (17).

$$M_{nk}(z) = \sum_{i=1}^n c_i K_i(z) + \sum_{p=1}^i \sum_{q=p}^n c_{p,q} K_p(z) K_q(z), \quad (17)$$

em que  $i, p, q \in \mathbb{N}$ ,  $K_{i,p,q}$  são as funções que compõe a base de Kautz, e  $M_{nk}$  é a função de transferência estimada. Esta estrutura é uma variação das séries de Voltera, que não será tratado neste trabalho, porém, em Oroski (2015) é feita uma abordagem mais profunda do assunto.

## 2.5 SÍNTESE

Este capítulo teve como objetivo o levantamento teórico sobre as funções ortonormais. Primeiramente, foi necessário estabelecer alguns conceitos, a fim de embasar a teoria utilizada no trabalho. Munido destes conhecimentos, pode-se concluir que as funções ortonormais compõem o então chamado espaço de Hilbert. Como implicação, estas funções são capazes de decompor vetores dentro deste espaço. Assim, estes vetores decompostos, por pertencerem ao espaço de Hilbert, podem ser aproximados por uma base de funções ortonormais. Convenientemente, é isso que se busca em identificação de sistemas.

Para este trabalho, foram escolhidas as funções ortonormais de FIR, Laguerre e Kautz. Estas possuem algumas características bem interessantes, como a inserção do conhecimento *a priori* do sistema. Para sistemas com polos na origem, a abordagem mais recomendada é o uso da base de funções FIR para a identificação. Se o sistema apresentar um comportamento superamortecido, através do que foi exposto no capítulo, indica-se a utilização das funções de Laguerre. Agora, se o sistema é majoritariamente subamortecido, a orientação é o emprego de funções de Kautz, porque apresentam um melhor desempenho para sistemas com polos complexos.

### 3 OTIMIZAÇÃO

Este capítulo é destinado a apresentação dos algoritmos de otimização utilizados como parte do algoritmo completo de identificação. Neste contexto, a Seção 3.1 versa sobre a utilização de técnicas de otimização. A Seção 3.2 explana sobre a escola determinística e sobre os algoritmos Gauss-Newton (GN), Gradiente Descendente (GD) e Levenberg-Marquardt (LM). A Seção 3.3 discorre sobre os algoritmos heurísticos como o algoritmo Algoritmo Genético (AG), *Particle Swarm Optimization* (PSO) e *Ant Colony Optimization* (ACO) e por fim, a Seção 3.4 estuda a combinação destas áreas nos chamados algoritmos híbridos.

#### 3.1 MÉTODOS DE OTIMIZAÇÃO

Segundo Talbi (2009), o principal objetivo em resolver um problema de otimização é encontrar a solução, ou conjunto de soluções, ótima(s). Esse conjunto de soluções pode encontrar o melhor local ou global, dependendo do problema e da técnica utilizada.

O que define numericamente a qualidade de uma resposta é a **Função Objetivo**. A função objetivo escolhida para este trabalho é o MSE (*Mean Square Error*), ou erro médio quadrático, e é definida pela equação (18)

$$MSE = \frac{1}{N} \sum_{k=1}^N (\hat{y}(k) - y(k))^2. \quad (18)$$

Assim, o problema de otimização passa a ter como objetivo a minimização de (18). A adoção desta métrica se deu, pois, segundo Oroski (2015), a mesma apresenta vantagens estatísticas.

Talbi (2009) define mínimo global  $x_g^*$  como a solução  $x^*$  que possui uma função objetivo  $f(x)$  menor que todas as outras soluções para o mesmo espaço  $S \subset \mathbb{R}^n$ . Define-se matematicamente o mínimo global como  $f(x) \geq f(x^*), \forall x \in S$ , ou, incluindo a tolerância,  $f(x) \geq f(x^*) - \varepsilon, \forall x \in S$ , no qual  $\varepsilon$  é uma determinada tolerância maior que zero.

Em contraste, um mínimo local  $x_l^*$  é a solução  $x^* \in S$  que minimiza a função objetivo  $f(x)$  na vizinhança e não é necessariamente um mínimo global (FLOUDAS, 2000). Define-se o mínimo local como  $f(x) \geq f(x^*), \forall x \in S : \|x - x^*\| < \varepsilon$ .

## 3.2 ALGORITMOS DETERMINÍSTICOS

### 3.2.1 Definições

Segundo Floudas (2000) para muitas aplicações práticas a determinação exata do mínimo global pode ser muito custosa computacionalmente e até mesmo desnecessária. Por isto, foi adotada a tolerância  $\varepsilon$  nas definições de mínimo global e local, fazendo com que o custo computacional seja reduzido.

Oroski (2015) explana o teorema 3 para auxiliar na busca dos pontos mínimos.

**Teorema 3** (Teorema de Máximos e Mínimos). *Seja  $x$  um ponto interior de  $S$  e suas derivadas parciais existem. Se  $x$  for um ponto de máximo ou mínimo de  $f$ , então o gradiente  $\nabla f$  será nulo.*

Define-se o gradiente da função objetivo em (19)

$$\nabla f(x) = \frac{\partial f(x)}{\partial x}. \quad (19)$$

No entanto, devido  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , seu gradiente é melhor representado pela Jacobiana  $J_{f(x)}$ ,

$$J_f(x) = \nabla f(x) = \left[ \frac{\partial f(x)}{\partial x_1} \quad \frac{\partial f(x)}{\partial x_2} \quad \dots \quad \frac{\partial f(x)}{\partial x_n} \right]^T. \quad (20)$$

Define-se também sua Hessiana  $H_f(x)$ ,

$$H_f(x) = \nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \dots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ & & \ddots & \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}. \quad (21)$$

A maioria dos métodos de otimização seguem a ideia de dar passos de tamanho  $\alpha_k$  em uma direção  $d_k$ , no qual  $k$  é o número da iteração e  $x_k$  e  $x_{k+1}$  são componentes do vetor de parâmetros  $x$ . A equação (22) representa esta ideia de maneira iterativa.

$$x_{k+1} = x_k + \alpha_k d_k, \quad (22)$$

em que o passo  $\alpha_k$  é escolhido de maneira que

$$\alpha_k = \underset{\alpha}{\operatorname{arg\,min}} f(x_k + \alpha d_k) \quad (23)$$

### 3.2.2 GD - GRADIENTE DESCENDENTE

Proposto originalmente por Cauchy (1847), trata-se de um dos primeiros métodos de otimização desenvolvidos. É pouco utilizado na prática, porém, é utilizado didaticamente como ponto de partida para outros métodos mais sofisticados de otimização. Neste método, é utilizada apenas a direção da descida mais íngreme, ignorando informações de iterações anteriores. A convergência deste método é boa no início, porém, torna-se demasiadamente lenta ao se aproximar do mínimo.

O método deriva da observação que uma função contínua deveria decrescer na direção de seu gradiente negativo (MEZA, 2010). A sua única dificuldade é em definir o tamanho do passo a ser dado. O Algoritmo 1 descreve a implementação do método, em que  $x_0$  são os ganhos iniciais e  $iteração_{max}$  representa o número máximo de iterações.

---

#### **Algoritmo 1:** Gradiente Descendente

---

Definem-se os parâmetros iniciais:  $x_0$ ,  $\varepsilon$  e  $iteração_{max}$ ;

**enquanto**  $tolerância > \varepsilon$  **and**  $iteração_{num} \leq iteração_{max}$  **faça**

    Calcula-se  $\alpha_k$  ;

    Calcula-se  $x_{k+1}$ ;

    Calcula-se  $\nabla f(x_{k+1})$ ;

    Calcula-se  $\varepsilon = \|\nabla f(x_{k+1})\|_2$ ;

**fim**

---

### 3.2.3 GN - GAUSS-NEWTON

Utilizado para resolver problemas de regressão não lineares, a ideia do método de Gauss-Newton (GN) é a de transformar o problema de minimização não linear em um problema de programação quadrática, evitando o cálculo da derivada de segunda ordem, que pode ser computacionalmente custoso (WANG, 2012).

O algoritmo será apresentado em sua versão clássica, proposta por Gauss (1809) e embasado em Wang (2012). Existem várias modificações que melhoram determinado aspecto deste método, porém estas fogem do escopo deste trabalho. O GN pode ser visto como uma modificação do método clássico de Newton, que possui a seguinte fórmula iterativa

$$x_{k+1} = x_k - [\nabla^2 f(x)]^{-1} \nabla f(x), \quad (24)$$

sendo  $\nabla f(x)$  expresso pela equação (25), em que  $\xi(x)$  é o vetor de resíduos e  $J_\xi$  a Jacobiana deste vetor.

$$\nabla f(x) = J_\xi(x) \xi(x), \quad (25)$$

$$\nabla^2 f(x) = \sum_{k=1}^n \nabla \xi(x) \nabla \xi(x)^\top - \sum_{k=1}^n \xi(x) \nabla^2 \xi(x). \quad (26)$$

O GN difere do método clássico de Newton negligenciando a segunda parte da equação (26). Isso é possível pois, segundo Wright (1999), a primeira parcela da equação (26) é geralmente mais importante que a segunda, fazendo com que o algoritmo se torne computacionalmente menos custoso e convergindo no mesmo número de iterações que o método clássico (WANG, 2012). No entanto, para problemas em que o segundo termo é significativo e não pode ser ignorado, a convergência do algoritmo se torna mais lenta.

Pode-se simplificar a equação (26) considerando a notação de  $J_f(x)$  e descartando a segunda parcela da mesma, tornando-a:

$$B(x) = \nabla f(x) \approx J_f(x)^\top J_f(x). \quad (27)$$

Os procedimentos do método são estruturados no Algoritmo 2.

---

**Algoritmo 2:** Gauss-Newton
 

---

Definem-se os parâmetros iniciais:  $x_0$ ,  $\varepsilon$  e  $iteração_{max}$ ;

**enquanto**  $tolerância > \varepsilon$  and  $iteração_{num} \leq iteração_{max}$  **faça**

    Calcula-se  $d_k$ ;

    Calcula-se  $B(x_0)d_0 = -\nabla f(x_0)$ ;

    Calcula-se  $x_{k+1} = x_k + d_k$ ;

    Calcula-se  $\varepsilon = \|\nabla f(x_{k+1})\|_2$ ;

**fim**

---

### 3.2.4 LM - LEVENBERG-MARQUARDT

De acordo com Ranganathan (2004), o método de LM foi apresentado originalmente por Levenberg (1944), a partir da observação das vantagens dos algoritmos GD e GN. Aproveitando-se dos passos largos, quando o gradiente é pequeno, proporcionado pelo GD, e da utilização da informação da segunda derivada do método GN, Levenberg propôs um método que combina os dois métodos anteriores, cuja regra de atualização é dada pela equação (28)

$$x_{k+1} = x_k - (H_f - \lambda I)^{-1} \nabla f(x_k), \quad (28)$$

na qual  $H_f$  é a matriz Hessiana no ponto  $x_k$ ,  $I$  é a matriz identidade e  $\lambda$  é o peso do gradiente.

A desvantagem do método acima é que se o valor de  $\lambda$  for grande, a Hessiana calculada torna-se menos significativa, aproximando o método de um simples GD. Para resolver isto, Marquardt (1963) propôs alterar a matriz identidade na equação (28) pela diagonal da Hessiana, resultando na regra de atualização de Levenberg-Marquardt, exposta na equação (29)

$$x_{k+1} = x_k - (H - \lambda \text{diag}[H])^{-1} \nabla f(x_k). \quad (29)$$

Devido à Hessiana ser proporcional à curvatura, a equação (29) implica em passos largos em direções de curvatura baixa e pequenos passos em direções de curvatura acentuada (RANGANATHAN, 2004).

A estrutura do método é descrita no Algoritmo 3. Apesar deste método funcionar muito bem na prática, o custo computacional de calcular a matriz inversa torna-se proibitivo para casos que apresentam alguns milhares de parâmetros (RANGANATHAN, 2004).

---

**Algoritmo 3: Levenberg-Marquardt**


---

Definem-se os parâmetros iniciais:  $x_0$ ,  $\varepsilon$  e  $iteração_{max}$ ;

**enquanto**  $tolerância > \varepsilon$  **and**  $iteração_{num} \leq iteração_{max}$  **faça**

Calcula-se a regra de atualização  $x_{k+1} = x_k - (H - \lambda \text{diag}[H])^{-1} \nabla f(x_k)$ ;

Calcula-se o erro  $\varepsilon$  para o novo vetor de parâmetros;

**se**  $\varepsilon$  **aumentou** **então**

Refaz-se o passo, restaurando os pesos para seus valores anteriores;

Multiplica-se  $\lambda$  por 10 ou algum outro fator significativo;

**senão**

Divide-se  $\lambda$  por 10 ou algum outro fator significativo;

**fim**

**fim**

---

### 3.3 ALGORITMOS HEURÍSTICOS

De acordo com Kokash (2018), algoritmos que fornecem um resultado aproximado ou não fornecem solução para todas as instâncias do problema são denominados heurísticos. Segundo Talbi (2009), heurísticas encontram boas soluções em uma grande variedade de problemas à um custo computacional aceitável e podem ser classificadas em duas categorias: heurísticas específicas e meta heurísticas. As heurísticas específicas são desenvolvidas para a resolução personalizada de um tipo de problema e as meta heurísticas servindo para várias classes de problemas de um “nível superior”.

Por se tratar de otimizações de técnicas de identificação caixa preta, que serão utilizadas em uma *toolbox* com a finalidade de atender aos mais diversos comportamentos de sistemas, existe um benefício claro na escolha de algoritmos meta heurísticos.

#### 3.3.1 AG - ALGORITMO GENÉTICO

Desenvolvido em 1962 (TALBI, 2009), baseia-se no mecanismo da seleção natural de Darwin. O AG combina a sobrevivência do mais apto com troca aleatória de informação genética para compor seu algoritmo (GOLDBERG, 1989). Faz-se analogia ao mecanismo de reprodução sexuado de organismos, em que cada geração cria descendentes cuja informação genética é uma combinação do material genético de seus progenitores. Baseado nesse mecanismo, o indivíduo que está mais adaptado ao seu ambiente, neste caso, o que possui

melhor ajuste, se reproduzirá e uma quantidade maior de proles terá suas informações genéticas. É este comportamento que leva o algoritmo à convergência.

Devido à facilidade de implementação, intuitividade e capacidade de resolver problemas altamente não lineares, o AG se tornou muito popular na academia e gerou inúmeras variações (HASSAN; WECK, 2005). Neste documento, será utilizada uma versão que contará com seleção via torneio, elitismo, taxa de *Crossover* decrescente e taxa de mutação crescente ao longo das iterações. A escolha destes elementos se deu pelo fato à proporcionarem maior liberdade no ajuste da pressão seletiva e não apresentarem grau elevado de complexidade.

É discorrido sobre cada função utilizada na composição do algoritmo e em seguida sua estrutura é apresentada no Algoritmo 4. No AG, cada indivíduo é modelado com um conjunto de constantes (genes), cuja quantidade varia conforme a dimensionalidade do problema a ser resolvido, e o vetor que engloba estas constantes é conhecido como cromossomo.

#### 3.3.1.1 *Fitness*

A função *Fitness* serve para avaliar o quanto cada indivíduo da população está apto ao seu ambiente. Na prática, isso significa calcular o erro entre o resultado obtido e o resultado esperado.

#### 3.3.1.2 Torneio

Consiste em sortear aleatoriamente um certo número de indivíduos e selecionar o que possuir melhor *Fitness*. Este indivíduo é denominado vencedor. O indivíduo selecionado será utilizado na função de *Crossover*. Quanto maior o número de indivíduos selecionados pelo torneio, maior a pressão seletiva (MILLER; GOLDBERG, 1995).

#### 3.3.1.3 *Crossover*

Quando um indivíduo é sorteado, parte de seu cromossomo é misturado com parte do cromossomo do vencedor do torneio, criando assim um novo indivíduo denominado prole. Esta operação possui o objetivo de realizar a busca global do espaço (OROSKI, 2015) e possui uma probabilidade de ocorrer para cada indivíduo, denominada *taxa de Crossover*.

#### 3.3.1.4 Elitismo

O elitismo possui a função de preservar os melhores indivíduos da população com base no *Fitness*. Seu uso é recomendado para que não se perca o melhor resultado global. É recomendado a preservação de poucos indivíduos devido ao aumento drástico da pressão seletiva.

#### 3.3.1.5 Pressão seletiva

De maneira simplificada, pressão seletiva é um termo utilizado na literatura de otimização genética para expressar a razão entre busca local e busca global do algoritmo. Uma pressão elevada resultará em uma convergência precipitada para um mínimo local, e uma pressão demasiadamente baixa poderá resultar na convergência lenta do algoritmo (PACHECO, 1999). Uma pressão seletiva equilibrada permite que o algoritmo busque o mínimo global sem ficar preso em mínimos locais.

#### 3.3.1.6 Mutação

Um indivíduo sorteado terá um de seus genes alterado por um valor aleatório. Esta operação possui o objetivo de busca local do espaço (OROSKI, 2015) e, assim como o *Crossover*, esta função possui uma probabilidade de ocorrer para cada indivíduo, denominada *taxa de mutação*.

#### 3.3.1.7 Estrutura do Algoritmo

O Algoritmo 4 apresenta a estrutura de implementação do método, utilizando as funções apresentadas nesta seção e parâmetros iniciais como a tolerância  $\epsilon$  e o número máximo de iterações  $iteração_{max}$ .

---

**Algoritmo 4:** Algoritmo Genético
 

---

Definem-se os parâmetros iniciais: taxa de *crossover*, taxa de mutação, dimensão da população,  $\epsilon$  e  $iteração_{max}$ ;

Gera-se a população aleatória de soluções;

Simula-se o modelo para cada cromossomo;

Avalia-se o *Fitness* de a cada cromossomo;

**enquanto**  $Fitness_{mínimo} > \epsilon$  and  $iteração_{num} \leq iteração_{max}$  **faça**

    Aplica-se o *Crossover*;

    Aplica-se a *Mutação*;

    Simula-se o modelo para cada cromossomo;

    Avalia-se o *Fitness* de acada cromossomo;

    Aplica-se o Elitismo;

**fim**

---

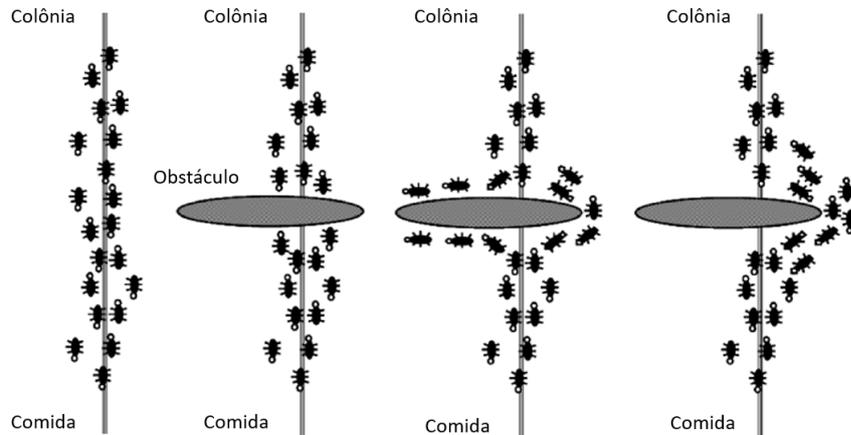
### 3.3.2 ACO - ANT COLONY OPTIMIZATION

Derivado do estudo do comportamento de formigas, o algoritmo utiliza-se do fato que formigas comunicam-se via *estigmergia*, uma forma de comunicação indireta através de modificações no ambiente, para coordenar uma população de agentes artificiais (DORIGO, 2004).

Existem inúmeras variações de algoritmos baseados nesse conceito. A versão clássica proposta por Dorigo (2004) aproveita-se do fato de que formigas deixam um rastro de feromônios por onde passam e as mesmas tendem a seguir o caminho que possui maior quantidade desta substância.

A Figura 6 exemplifica este comportamento. Quando um obstáculo é inserido entre o ninho e o alimento da colônia, existe a mesma probabilidade de o contornarem por ambos os lados, porém o caminho do lado direito é menor. Isso acarretará que para um mesmo intervalo de tempo, mais formigas passarão pelo lado mais curto (TALBI, 2009). Pela quantidade de feromônio ser proporcional à quantidade de formigas, o menor caminho possuirá uma quantidade maior dessa substância, guiando as outras formigas por este mesmo caminho e criando um *feedback* positivo conhecido como *autocatálise* (DORIGO, 2006).

**Figura 6: Conceito básico ACO.**



FONTE: (TALBI, 2009).

Dorigo (2004) define uma formiga artificial como um procedimento estocástico construtivo que, incrementalmente, cria uma solução adicionando componentes de solução à uma solução parcial em construção. Isto significa que o método pode ser utilizado em qualquer problema de otimização combinatorial. A principal dificuldade, segundo Dorigo (2004), está em definir o “mapa” que as formigas artificiais usarão para construir a solução.

### 3.3.3 Definições iniciais

De acordo com Dorigo (2004), uma boa heurística é inicializar os traços de feromônios  $\tau_0$  em um valor levemente acima do valor esperado depositado em uma iteração. Se o valor inicial for muito pequeno, o algoritmo possuirá uma tendência de convergir para os primeiros caminhos percorridos.

### 3.3.4 Tour

Em toda iteração, é aplicado a cada formiga o chamado *random proportional rule*, para decidir qual a próxima possível solução a se visitar. A probabilidade da formiga  $k$ , que se encontra na solução  $i$ , de ir para a solução  $j$  é definida na equação (30)

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \quad \text{se } j \in N_i^k, \quad (30)$$

em que  $\eta_{ij} = 1/d_{ij}$  e significa que a probabilidade heurística de ir da solução  $i$  à solução  $j$  é inversamente proporcional à sua distância,  $\alpha$  é o peso exponencial do feromônio,  $\beta$  é o peso

exponencial heurístico,  $N$  é a vizinhança possível (soluções não visitadas) da formiga,  $\tau_{ij}$  é o valor do traço do feromônio e  $\eta_{ij}$  é o valor da informação heurística.

O parâmetro  $\alpha$  influencia de tal maneira que, para  $\alpha = 0$ , as soluções mais próximas tem chance maior de serem selecionadas. Se  $\beta = 0$ , são utilizados apenas os feromônios. Isto normalmente culmina em resultados pobres e, combinado com valores de  $\alpha > 1$ , leva a estagnação rápida do algoritmo, em que todas as formigas seguem o mesmo caminho e constroem o mesmo *tour*.

### 3.3.5 Evaporação e atualização de feromônios

Após a construção do *tour*, os feromônios são evaporados em todos os caminhos por um fator constante  $\rho$ ,  $0 < \rho < 1$ , por meio da expressão (31)

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij}, \quad \forall (i, j) \in L. \quad (31)$$

Após a evaporação, são depositados feromônios onde as formigas passaram através da expressão (32)

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k, \quad \forall (i, j) \in L, \quad (32)$$

em que  $\Delta\tau_{ij}^k$  é a quantidade de feromônio que a formiga  $k$  deposita e  $L$  são os possíveis caminhos.  $\Delta\tau_{ij}^k$  e é definido pela equação (33):

$$\Delta\tau_{ij}^k = \begin{cases} 1/C^k, & \text{se o traço } (i, j) \text{ pertence a } T^k, \\ 0, & \text{caso contrário.} \end{cases} \quad (33)$$

em que  $C^k$  é o comprimento do *tour*  $T^k$ , que é calculado como a soma dos comprimentos dos caminhos pertencentes à  $T^k$ .

Dorigo (2004) menciona que o desempenho deste algoritmo, comparado com outros meta heurísticos, tende a decrescer dramaticamente conforme as iterações aumentam. A estrutura básica do método conhecida como AS (*Ant System*), baseada em (DORIGO, 2006), é apresentada no Algoritmo 5.

---

**Algoritmo 5: Ant Colony Optimization**


---

Definem-se os parâmetros iniciais  $\tau_0$ ,  $k$ ,  $\alpha$ ,  $\beta$ ,  $\varepsilon$  e  $iteração_{max}$ ;

Gera-se o mapa de possíveis soluções ;

**enquanto**  $Fitness_{mínimo} > \varepsilon$  **and**  $iteração_{num} \leq iteração_{max}$  **faça**

Realiza-se o tour;

Avaliam-se as soluções encontradas através do *Fitness*;

Evaporam-se os feromônios;

Atualiza-se o valor dos feromônios;

**fim**

---

### 3.3.6 PSO - PARTICLE SWARM OPTIMIZATION

Desenvolvido por Eberhart e Kennedy (1995), o PSO foi inspirado pelo comportamento cooperativo de vários bandos de animais para varrer seu espaço de busca, a fim de suprir suas necessidades. No PSO, uma população inicial aleatória é gerada e propagada ao espaço de possíveis soluções em busca da melhor solução global através de movimentos (iterações) e de informações que são compartilhadas por todos os membros da população (HASSAN; WECK, 2005). Utiliza-se da experiência de cada partícula (*Pbest*), da experiência da população (*Gbest*) e da velocidade individual para definir a próxima posição no espaço de busca (ALAM, 2016). A Figura 7 exprime graficamente esta relação.

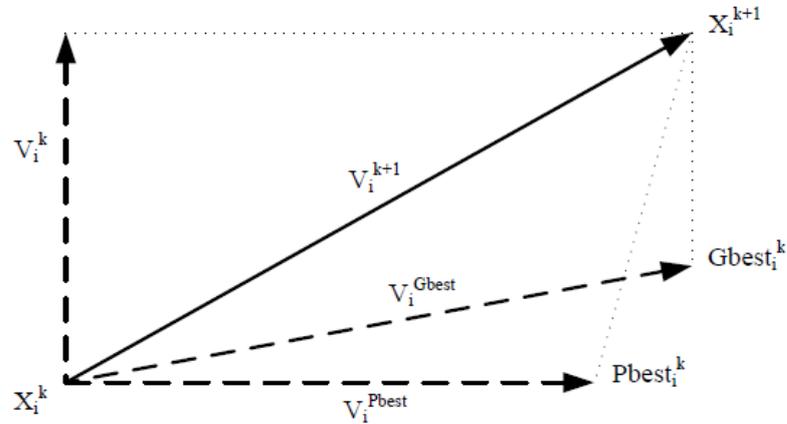
As equações (34) e (35) são fundamentais para o método, pois representam matematica e iterativamente os conceitos da Figura 7, sendo que em (34) é atualizada a velocidade de cada partícula e, em (35), a sua posição.

$$V_{i,j}^{k+1} = \omega V_{i,j}^k + c_1 r_1 (Pbest_{i,j}^k - X_{i,j}^k) + c_2 r_2 (Gbest_{i,j}^k - X_{i,j}^k), \quad (34)$$

$$X_{i,j}^{k+1} = X_{i,j}^k + V_{i,j}^{k+1}, \quad (35)$$

em que  $i$  é o número da partícula,  $j$  é o número da componente dimensional da partícula,  $k$  é o número da iteração,  $X$  é a posição da partícula,  $V$  é a velocidade da partícula,  $\omega$  é o coeficiente de inércia,  $c_1$  é o coeficiente pessoal da partícula,  $c_2$  é o coeficiente global da população,  $r_1$  e  $r_2$  são fatores aleatórios com distribuição uniforme entre 0 e 1.

**Figura 7: Mecanismo de busca do PSO.**



FONTE: (ALAM, 2016).

A metodologia base do PSO é descrita no Algoritmo 6, sendo que  $b$  é o número da melhor partícula,  $Pbest$  é o melhor resultado da partícula e  $Gbest$  é o melhor resultado da população.

---

**Algoritmo 6: Particle Swarm Optimization**

---

Definem-se os valores dos parâmetros iniciais  $c_1$ ,  $c_2$ ,  $\omega$ ,  $\varepsilon$  e  $iteração_{max}$ ;

Inicializa-se uma população com partículas de posição aleatórias e velocidade nula (ou um valor escolhido);

Calcula-se o *Fitness* das partículas  $F_i^k = f(X_i^k), \forall i$  e armazena o índice  $b$  da melhor partícula;

Faz-se  $Pbest_i^k = X_i^k, \forall i$  e  $Gbest^k = X_b^k$ ;

**enquanto**  $Fitness_{mínimo} > \varepsilon$  **and**  $iteração_{num} \leq iteração_{max}$  **faça**

- Atualiza-se a velocidade e a posição das partículas de acordo com as equações (34) e (35), respectivamente;
- Calcula-se novamente o *Fitness* e verifica se o novo valor ( $F_i^{k+1}$ ) é menor que o  $Pbest$ , se positivo, atualiza-se o valor de  $Pbest$ ;
- Avalia se o novo  $Gbest$  é menor que o anterior, caso verdadeiro, atualiza-se o valor de  $Gbest$ ;

**fim**

---

### 3.4 ALGORITMO HÍBRIDO

Existem na literatura diversos trabalhos que buscam desenvolver modelos de algoritmos híbridos para otimização (BOECHEL, 2003; LIMA, 2008; FRITSCHE et al., 2014; LUCHI, 2016). Neste trabalho, é proposto um modelo de algoritmo que combina o algoritmo PSO e o LM.

O método consiste em utilizar-se, em um primeiro momento, do algoritmo PSO, que possui uma dependência menor do ponto inicial, para encontrar uma solução (ou conjunto de soluções) adequada ao problema e, em seguida, aplicá-la como ponto de partida para que o algoritmo LM realize uma busca mais precisa e encontre o ponto ótimo. Esta combinação se aproveita do ponto forte dos métodos heurísticos, que é encontrar um conjunto de soluções adequadas ao problema com menor custo computacional, com a vantagem dos métodos determinísticos em garantirem o mínimo global.

O Algoritmo Híbrido é descrito no Algoritmo 7. Procurou-se manter a mesma simbologia para as variáveis presentes nos algoritmos PSO e LM apresentados anteriormente.

---

#### **Algoritmo 7:** *Algoritmo Híbrido*

---

Definem-se os valores dos parâmetros iniciais  $c_1$ ,  $c_2$ ,  $\omega$ ,  $\varepsilon$  e  $iteração_{max}$ ;

Inicializa-se uma população com partículas de posição aleatórias e velocidade nula (ou um valor escolhido);

Calcula-se o *Fitness* das partículas  $F_i^k = f(X_i^k)$ ,  $\forall i$  e armazena o índice  $b$  da melhor partícula;

Faz-se  $Pbest_i^k = X_i^k$ ,  $\forall i$  e  $Gbest^k = X_b^k$ ;

**enquanto**  $Fitness_{mínimo} > \varepsilon$  **and**  $iteração_{num} \leq iteração_{max}$  **faça**

- Atualiza-se a velocidade e a posição das partículas de acordo com as equações (34) e (35), respectivamente;
- Calcula-se novamente o *Fitness* e verifica se o novo valor ( $F_i^{k+1}$ ) é menor que o *Pbest*, se positivo, atualiza-se o valor de *Pbest*;
- Avalia se o novo *Gbest* é menor que o anterior, caso verdadeiro, atualiza-se o valor de *Gbest*;
- Gera uma variável  $n$  aleatória com distribuição uniforme e valor entre 0 e 1;

**se**  $n > 0,5$  **então**

Define-se  $b$  como parâmetros iniciais,  $x_0$ , do LM;

Executa o LM descrito no Algoritmo 3;

**fim**

---

### 3.5 SÍNTESE

Neste capítulo apresentaram-se os principais métodos de otimização determinística e meta heurísticos e uma comparação entre seus usos, vantagens e desvantagens. Apresentaram-se os métodos e algoritmos determinísticos de GN, GD e LM e os meta heurísticos AG, PSO e ACO. Também foram expostas as bases matemáticas necessárias.

Algoritmos meta heurísticos são utilizados para uma grande variedade de problemas e possuem um custo computacional menor se comparado com os métodos determinísticos. No entanto, não garantem uma solução ótima. Em contraste, algoritmos determinísticos são mais sensíveis ao comportamento do sistema e dos parâmetros iniciais, entretanto, conseguem garantir uma solução ótima ao custo de maior processamento.

## 4 DESENVOLVIMENTO GRÁFICO DA *TOOLBOX*

Neste capítulo, são apresentados a metodologia e o embasamento pedagógico para o desenvolvimento da *toolbox*. O mesmo está dividido em duas seções. Na Seção 4.1, são expostos sobre os critérios de avaliação de ferramentas educacionais e alguns parâmetros encontrados na literatura que devem ser observados no momento do desenvolvimento de uma aplicação como a proposta neste trabalho. Na Seção 4.2, são apresentados os resultados da interface gráfica desenvolvida e suas funcionalidades.

### 4.1 CRITÉRIOS PEDAGÓGICOS DE DESENVOLVIMENTO

Para Giraffa (1999) qualquer *software* pode ser definido como um Software Educacional (SE), desde que o mesmo seja aplicado no processo de ensino e aprendizagem por meio de uma metodologia definida. Autores como Silva (2012), Oliveira et al. (2001) apresentam outras definições para SE que, apesar das diferenças, ambas convergem para a conclusão de que um SE precisa facilitar o processo de aprendizagem e possuir uma metodologia pedagógica elaborada pelo profissional que o utilize. Outro ponto importante para ser ressaltado é que a ferramenta não deve substituir o professor, e sim, ser um meio de auxílio ao profissional.

Neste cenário, buscou-se a elaboração de um SE com auxílio do GUIDE (*Graphical User Interface Development Environment*), um ambiente nativo do MATLAB<sup>®</sup> que fornece ferramentas para criar aplicativos customizados para o mesmo. O GUIDE gerará automaticamente o código da UI (*User Interface*) que será modificado para construir um *layout* intuitivo ao usuário. O objetivo desta etapa consistiu em atrelar os algoritmos desenvolvidos com uma interface para o usuário.

O desenvolvimento de uma *toolbox* para ser utilizada como um recurso didático requer alguns aspectos de qualidade. O primeiro a ser observado é o exposto na NBR 924-11 de 2002. Esta norma trata sobre os requisitos ergonômicos para trabalho de escritórios com

computadores, com um enfoque na usabilidade. A orientação presente nesta norma é que, para medir a usabilidade que o usuário terá ao entrar em contato com o *software*, é necessário identificar os objetivos do contexto de uso e decompor em conjuntos mensuráveis. Em Nielsen (1994) foi realizado algo semelhante ao que a norma apresentou anos mais tarde, neste trabalho o autor destaca a importância da usabilidade quando enumera cinco critérios que um *software* deve alcançar, sendo eles:

- Intuitividade - o programa precisa ser de fácil operação;
- Eficiência - o programa deve desempenhar um alto nível de produtividade;
- Memorização - as telas precisam ser fáceis de serem memorizadas;
- Erro - o programa deve apresentar o mínimo possível de erros;
- Satisfação - o programa deve satisfazer o usuário.

Segundo Koscianski e Soares (2007) outros aspectos que devem ser avaliados além da usabilidade é a funcionalidade, manutenibilidade e portabilidade. Dentro da literatura sobre SE, encontram-se diversos trabalhos que tratam sobre como avaliar a qualidade do mesmo (PEREIRA et al., 2016; OLIVEIRA et al., 2011; RAMOS; MENDONÇA, 1991). Assim, utilizou-se dos principais critérios de avaliação para desenvolver uma interface para o usuário. Os parâmetros mais importantes são elencados abaixo.

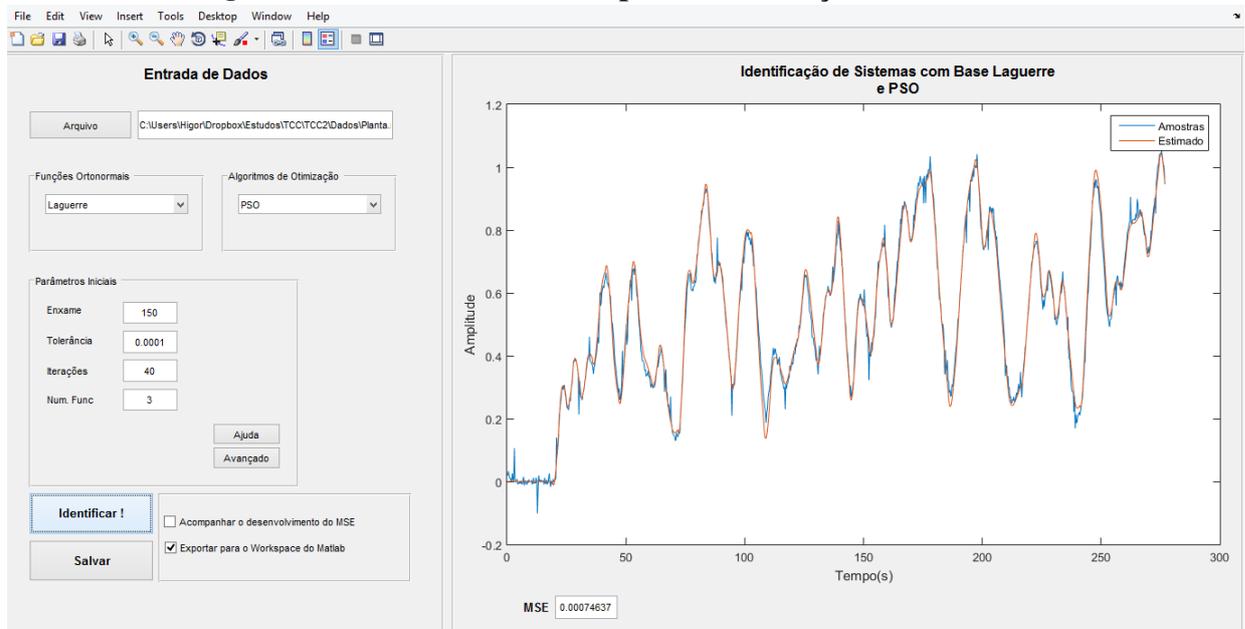
- Sucessão de Operações - este parâmetro trata sobre um preenchimento das informações nos campos da interface de forma lógica ao usuário. Também ele trata da possibilidade do usuário ter acesso sobre todas as informações a todo o momento e poder transferir informações de uma aplicação para outra;
- Linguagem de Iteração - este critério é referente ao vocabulário utilizado na interface. Ele remete ao uso da terminologia que o aluno esteja ambientalizado, neste caso específico, às terminologias utilizadas em identificação de sistemas;
- Tempo de Resposta - este parâmetro aborda o tempo em que se leva para alcançar o resultado. Não existe consenso entre os especialistas sobre um tempo ideal, mas deve-se levar em consideração o que o usuário acredita ter um tempo aceitável para alcançar o resultado da aplicação;
- Tratamento de Erro - o erro que o usuário cometer deve ser rapidamente assinalado, dando preferência para modelos que indiquem possíveis diagnósticos dos erros;

- Ajuda - é necessário o desenvolvimento de um sistema de ajuda ao usuário que seja de fácil interpretação. O sistema deve prever ajuda sobre a funcionalidade da ferramenta e ajuda referente aos conceitos teóricos utilizados pelo *software*. Quando possível, é interessante a utilização de ajudas *online* e disponibilização de referências bibliográficas caso o usuário queira ir além do que lhe foi fornecido.

## 4.2 INTERFACE GRÁFICA

Buscando seguir os parâmetros elencados na Seção 4.1, foi desenvolvida a *toolbox* para identificação de sistemas. A interface com o usuário é mostrada na Figura 8.

**Figura 8: Toolbox desenvolvida para identificação de sistemas.**



FONTE: Autoria Própria.

Foi desenvolvido um ambiente em que as informações gerais da ferramenta são acessadas diretamente na tela inicial. O *software* está dividido em duas áreas. A primeira delas apresenta todas as informações que o usuário deve inserir para realizar a identificação, começando pelo arquivo com os dados amostrados, passando pela escolha da função ortonormal, a seleção do otimizador e, por último, a inserção dos parâmetros que o usuário deseja para os algoritmos. A Figura 9 exibe melhor esta área.

Os parâmetros iniciais para os otimizadores não são iguais para todos os algoritmos. Sendo assim, cada vez que for selecionado um tipo de otimizador os parâmetros iniciais serão

modificados de forma a não apresentar informações desnecessárias ao usuário. Neste campo, também há um botão para acionar uma guia com parâmetros avançados. Este botão não se habilita para os algoritmos de otimização determinística, pois na forma como estes foram construídos, não se deixou flexibilidade para uma sintonização avançada dos mesmos.

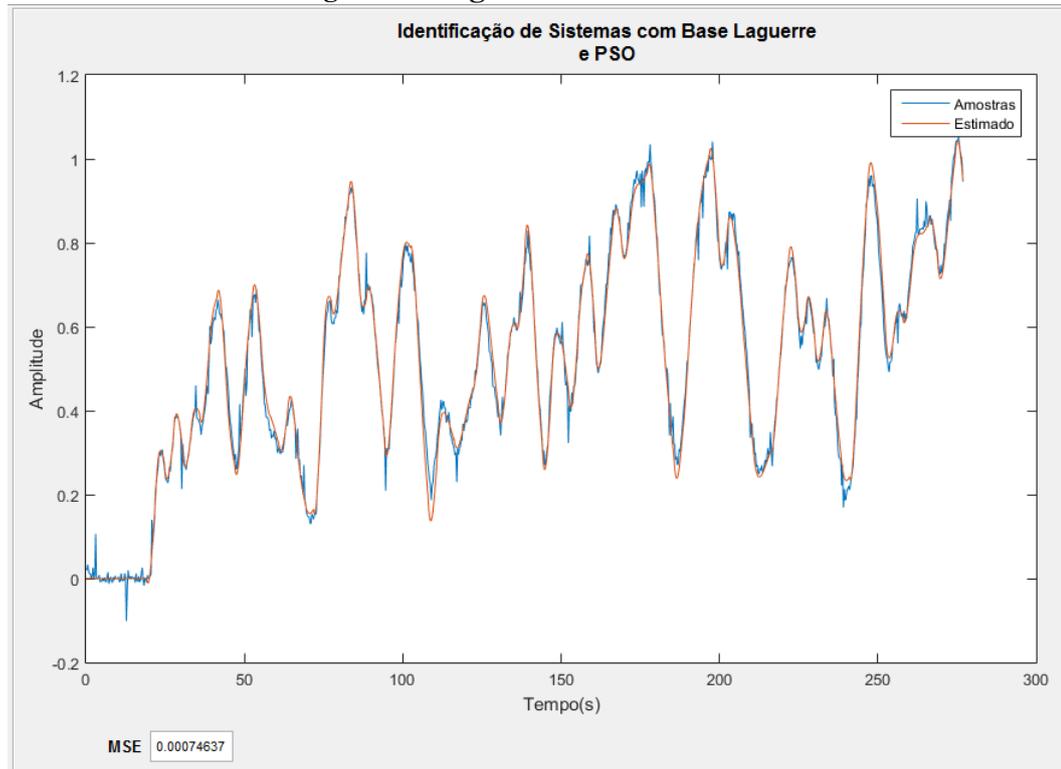
**Figura 9: Primeira área da *toolbox*.**

FONTE: Autoria Própria.

Ainda nesta área, encontra-se um botão que ativa o menu de ajuda. Este botão aciona uma janela à parte em que o usuário pode encontrar informações a respeito dos parâmetros iniciais e como eles podem influenciar o algoritmo, informações sobre as funções ortonormais e os algoritmos de otimização, além de uma gama de referências para um aprofundamento do estudo.

A segunda área da *toolbox* é destinada para os resultados atingidos pelos algoritmos, apresentando um resultado visual por meio de um gráfico com o sinal amostrado da planta e o sinal gerado pelo modelo desenvolvido. Na parte inferior desta área, apresenta-se o MSE que o modelo atingiu, sendo este o elemento métrico utilizado para classificar o desempenho da modelagem. A Figura 10 exibe esta área com mais detalhes.

**Figura 10: Segunda área da *toolbox*.**



FONTE: Autoria Própria.

Ao encontrar um modelo que esteja dentro da faixa de precisão que o usuário queira, existe a opção de salvar as informações do modelo. Para isto, na primeira área da ferramenta há um botão para salvar. Este permite que os dados sejam salvos no *workspace* do MATLAB<sup>®</sup> ou como um arquivo de dados em qualquer local do computador. Optou-se por deixar estas duas opções para que se possa usar estes dados não somente no MATLAB<sup>®</sup>, buscando a portabilidade dos resultados. Ao salvar os resultados uma série de informações são fornecidas ao usuário, sendo elas: o número de funções utilizadas, o ganho de cada função, o(s) polo(s) que parametrizou (aram) as funções, a base de funções ortonormais, a função de transferência que modela o sistema, o MSE, um vetor com as saídas estimadas, um vetor com os dados que excitaram o sistema, o histórico do desenvolvimento do MSE, as amostras da planta e os parâmetros de sintonia que são diferentes para cada algoritmo de otimização. Estas informações são fornecidas a fim de permitir a repetibilidade do experimento e para que possam ser realizados diferentes tratamentos com o modelo, por exemplo, projetar um controlador.

### 4.3 SÍNTESE

Neste capítulo foram apresentados quais os critérios utilizados para o desenvolvimento da *toolbox* de identificação de sistemas. Como pode ser observado, a ferramenta apresenta uma interface compacta, sem exibir excesso de informações para o usuário. Com isto, pretende-se facilitar o manuseio da ferramenta por usuários iniciantes. Também foi exposto o resultado do desenvolvimento gráfico e descrita as funcionalidades da ferramenta.

## 5 METODOLOGIA DOS EXPERIMENTOS DE VALIDAÇÃO

Com o estudo das funções ortonormais e os algoritmos de otimização, expostos nos capítulos anteriores, tornou-se possível a implementação de alguns algoritmos na plataforma do MATLAB<sup>®</sup>. O *script* implementado permite a geração de bases utilizando as funções ortonormais e o uso dos algoritmos de otimização estudados. Neste capítulo, é discorrido sobre a metodologia de validação dos algoritmos e sobre a metodologia do experimento realizado em sala de aula para a validação da *toolbox* desenvolvida.

### 5.1 METODOLOGIA DE IDENTIFICAÇÃO DE SISTEMAS

Para validação dos algoritmos implementados, foi realizada a identificação de uma planta didática, mostrada na Figura 11. Esta é uma planta desenvolvida pela Smar<sup>®</sup> para controle de fluxo de água e aferição do teor de pH do fluido. Ela possui um sistema de atuação composto por uma bomba de água que é controlada por um inversor. Esta planta se encontra no Laboratório de Instrumentação Industrial da UTFPR, localizado na sala B-004.

**Figura 11: Planta didática utilizada para identificação.**

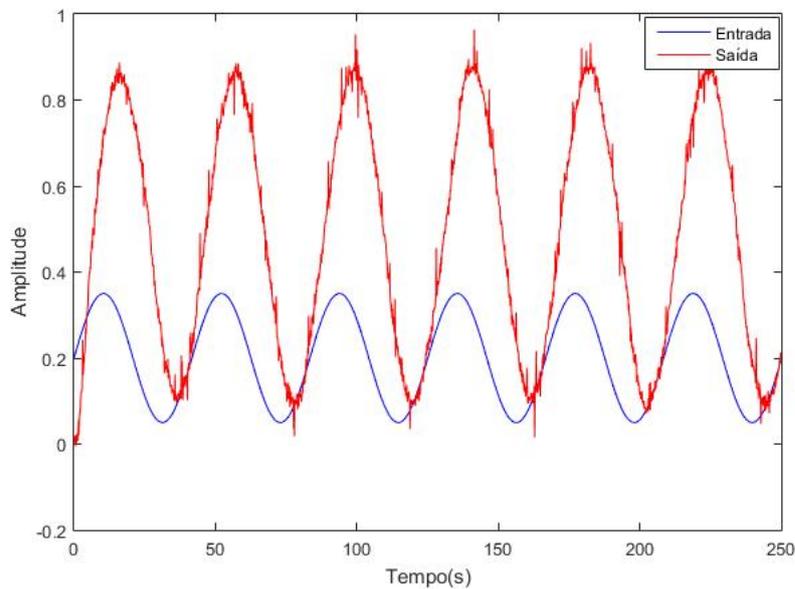


FONTE: Aatoria Própria.

Inicialmente, foi aplicado à planta um sinal senoidal, mostado na equação (36), para testar sua linearidade. Em que  $x$  é o sinal e  $t$  é o tempo medido em segundos. A Figura 12 mostra o resultado deste teste. Em azul, é visto o sinal de entrada e, em vermelho, o sinal da saída. Como pode-se observar, existem alguns ruídos no sinal de saída, a amplitude é diferente, o sinal apresenta um atraso na resposta e um *offset* em relação ao sinal que excitou a planta.

$$x = 0,20 + 0,15 \times \text{sen}(0,13 \times t) \quad (36)$$

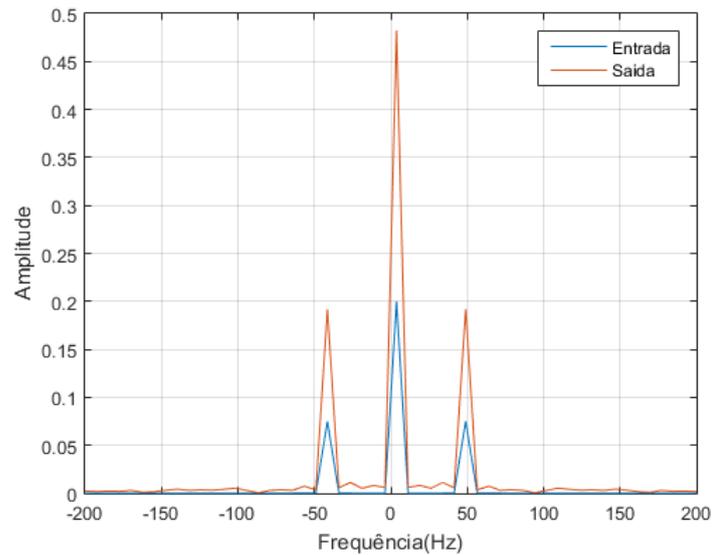
**Figura 12: Sinal aplicado e coletado para teste de linearidade.**



FONTE: Autoria Própria.

Para ter certeza sobre a linearidade do sistema, foi feita uma análise no espectro da frequência, mostrado na Figura 13, a fim de saber se as mesmas componentes de frequência aparecem em ambos os sinais.

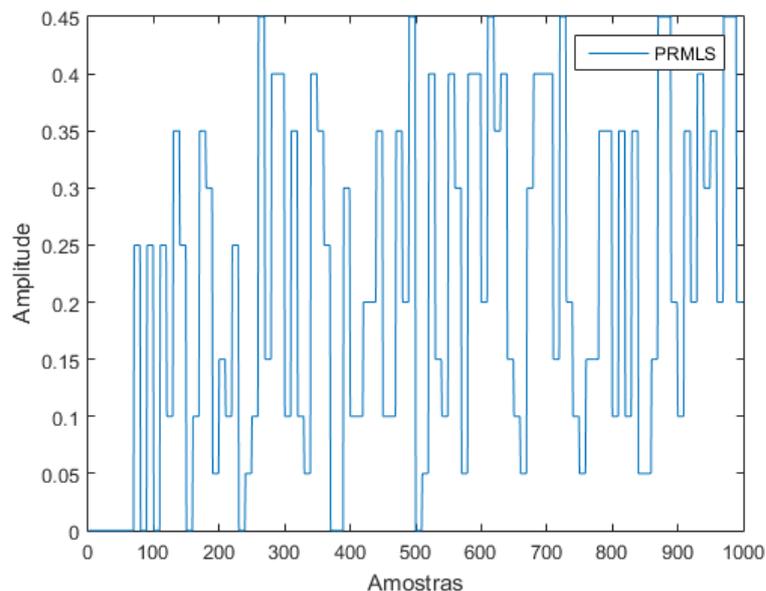
**Figura 13: Espectro de frequência dos sinais de entrada e saída.**



FONTE: Autoria Própria.

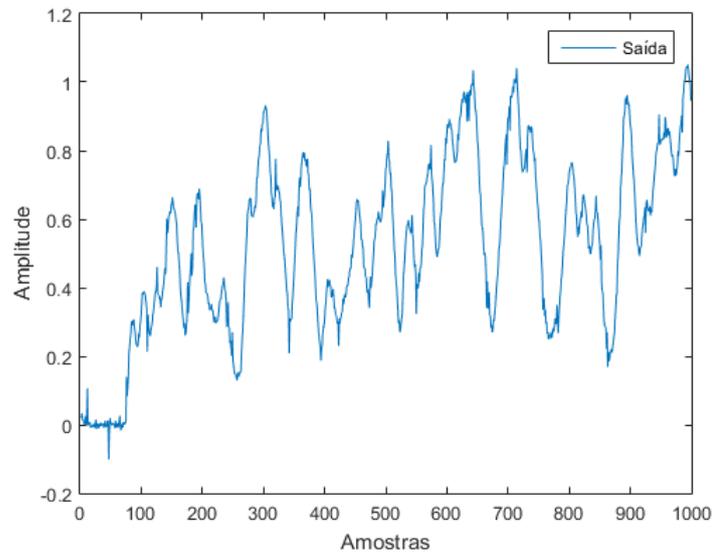
Como pode ser notado, o sinal de saída, em vermelho, apresenta algumas componentes de frequências diferentes aos do sinal de entrada, em azul, e existe uma frequência central em 0 Hz que representa o *offset* já mencionado. Entretanto, esta planta será considerada linear, pois as frequências mais relevantes estão dentro da mesma faixa do espectro. Após esta definição, o sistema foi excitado com um sinal PRMLS (*Pseudo Random Multi Level Sequence*), mostrado na Figura 14. A resposta do sistema esta entrada é exibida na Figura 15.

**Figura 14: Sinal PRMLS aplicado ao sistema.**



FONTE: Autoria Própria.

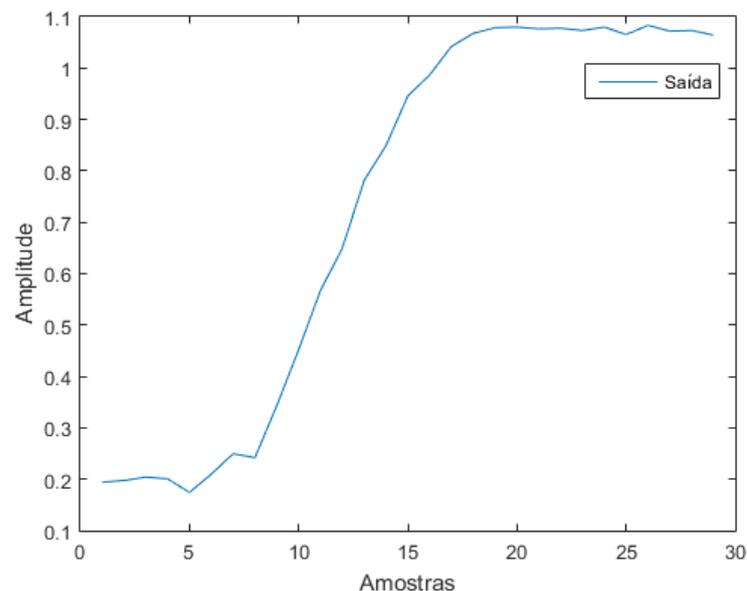
**Figura 15: Sinal de saída do sistema.**



FONTE: Aatoria Própria.

Para finalizar a coleta dos dados da planta, excitou-se o sistema com uma função degrau unitário. Com isto, pretende-se saber como é o comportamento da planta, uma vez que as funções ortonormais permitem o uso do conhecimento *a priori* para escolher qual função terá o melhor desempenho na representação do sistema. A Figura 16 fornece a resposta do sistema para este tipo de excitação e, como pode-se observar, a planta estudada tem o comportamento de sistemas superamortecidos.

**Figura 16: Resposta do sistema à função degrau unitário.**



FONTE: Aatoria Própria.

## 5.2 METODOLOGIA DO EXPERIMENTO PEDAGÓGICO

Munido pontos norteadores apresentados na Seção 4.1, foi elaborada uma interface gráfica para realizar a identificação de sistemas. Com o *software* pronto, foi realizado um experimento em sala de aula na disciplina de Introdução à Identificação de Sistemas com 21 (vinte e um) alunos. Este grupo era composto por alunos dos cursos de Engenharia de Controle e Automação e Engenharia Elétrica. Estes estudantes foram considerados usuários com conhecimento básico de identificação de sistemas, pois os mesmos já haviam acompanhado diversas aulas da disciplina.

O experimento consistiu em quatro fases. Na primeira etapa, foi fornecido um conjunto de dados aos alunos e pedido para que realizassem a identificação do sistema com as diferentes ferramentas disponíveis na *toolbox*. Esta etapa teve duração de 20 minutos e nela não ocorreu interferência do professor. O objetivo inicial foi observar como os alunos reagiriam a *toolbox* sem que houvesse um conhecimento da ferramenta, criando uma situação de primeiro contato do usuário com a aplicação.

Na segunda etapa, foi realizada uma exposição teórica sobre as OBF's e os algoritmos de otimização. Neste momento, também foi aberto para perguntas e teve uma duração aproximada de 40 minutos. Esta etapa foi necessária, pois, apesar dos alunos estarem cursando a disciplina de identificação de sistemas, o tópico de identificação com funções ortonormais e alguns algoritmos de otimização não são abordados na disciplina. Na terceira fase do experimento, foi disponibilizado um segundo conjunto de dados e requereu o mesmo desafio inicial aos estudantes, desta vez por 15 minutos. Com as três primeiras fases concluídas, foi entregue um formulário com nove questões para os estudantes responderem individualmente. O formulário consiste em oito questões objetivas e um questão dissertativa, podendo ser visto no Apêndice A. Esta última fase não teve um tempo definido, à medida que os indivíduos terminavam seu questionário, o mesmo era recolhido.

As oito questões objetivas, buscou-se enquadrar três dos cinco parâmetros expostos por Nielsen (1994). Devido a aplicação possuir apenas uma tela principal, em que se encontra as principais informações para realizar a identificação, julgou-se não ser necessária a avaliação do critério de Memorização. O parâmetro Erro também não foi considerado, porque o mesmo pôde ser diagnosticado na fase de desenvolvimento da *toolbox*, sendo que este quesito foi substituído pela análise do sistema do menu de ajuda desenvolvido, pois o critério Suporte (ou Ajuda) é de grande relevância nos trabalhos de avaliação de SE já citados. A questão nove teve como objetivo abrir a discussão para os alunos darem sua opinião sobre o que poderia ser melhorado.

### 5.3 SÍNTESE

Este capítulo abordou as metodologias empregadas nos experimentos de validação. Na primeira seção, descreveu-se a coleta dos dados de uma planta didática para validar os algoritmos de otimização e identificação. Também foi estudado o comportamento desta planta nos quesitos de linearidade e de resposta ao degrau unitário. Com isto, foi definido que para este trabalho a planta será considerada um sistema linear e superamortecido. Na Seção 5.2, foi descrito os procedimentos do experimento realizado com os alunos em sala de aula. Esta segunda parte tem como objetivo validar a interface desenvolvida para a *toolbox*.

## 6 ANÁLISES E RESULTADOS

Com a aplicação das metodologias descritas no Capítulo 5, chegou-se à um conjunto de resultados que são expostos e analisados neste capítulo. A Seção 6.1 aborda os resultados dos testes com os algoritmos de identificação e otimização e, por sua vez, na Seção 6.2, é discorrido sobre os resultados do experimento pedagógico realizado com os alunos.

### 6.1 RESULTADOS DE IDENTIFICAÇÃO E OTIMIZAÇÃO

Após a validação da linearidade da planta e a coleta dos dados, foi feito o tratamento dos sinais na *toolbox*. Primeiramente, foram realizados alguns testes com o algoritmo genético. Como exposto no Capítulo 2, devido ao comportamento superamortecido da planta, as funções mais indicadas para se trabalhar, neste caso, são as de Laguerre. No entanto, para fins de validação, foi realizada a identificação com as três funções estudadas. Os parâmetros do algoritmo são mostrados na Tabela 1. Estes foram ajustados de forma empírica, assim como os parâmetros dos otimizadores que virão na sequência.

**Tabela 1: Parâmetros para modelo AG.**

Parâmetros	Valor
População	200
Nº de Funções	Aleatório
Taxa de Crossover	80-60%
Taxa de Mutação	20-80%

FONTE: Aatoria Própria.

Como o número ideal de funções para formar a base é desconhecido, as bases foram formadas com dimensões aleatórias, na busca de encontrar um valor que diminuísse o erro a um valor satisfatório. Em relação ao erro, o critério de avaliação foi o erro médio quadrado (MSE), expresso pela equação (18).

É importante ressaltar que a taxa de *crossover* varia em forma decrescente e a taxa

mutação de modo crescente, a fim de dificultar a convergência para mínimos locais. Com isto, os resultados obtidos são mostrados na Tabela 2. É interessante destacar que, para as bases de Kautz, o algoritmo retorna os polos parametrizados, ou seja,  $b$  e  $c$  descritos nas equações (12) e (13).

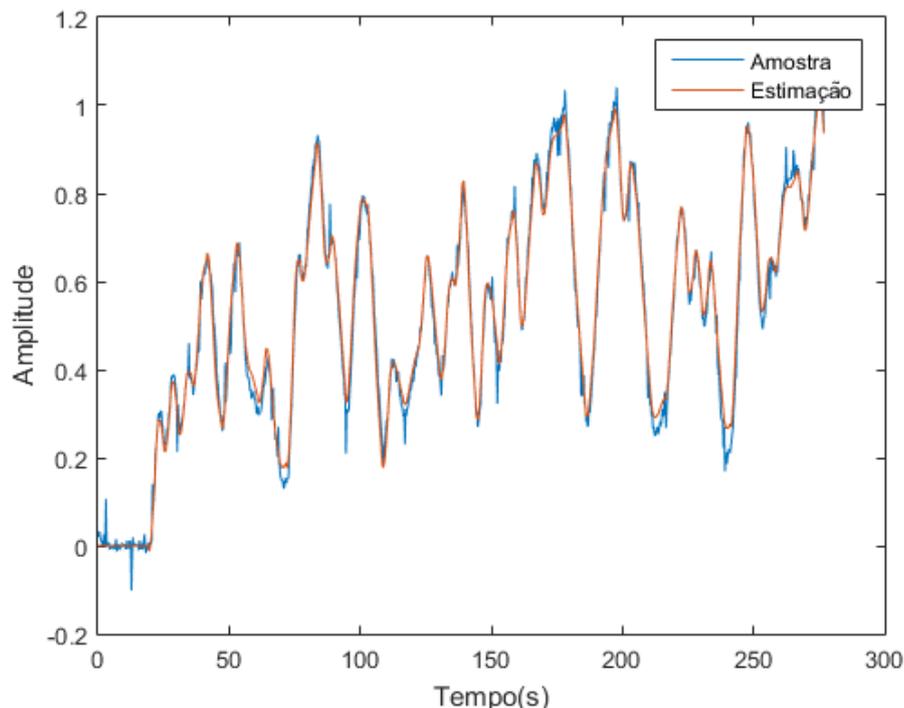
**Tabela 2: Resultados do Algoritmo Genético.**

Função	Nº Funções	Polos	MSE
FIR	22	0	0,0185
Laguerre	2	0,881	$7,97 \times 10^{-4}$
Kautz	10	$-0,414 \pm 0,798i$	$9,16 \times 10^{-4}$

FONTE: Autoria Própria.

De antemão, pode-se comprovar que o sistema não possui polos próximos da origem, pois foi necessário um grande número de funções para formar a base FIR e, mesmo assim, não atingiu um bom resultado. Para os resultados das identificações com as bases formadas pelas funções de Kautz e Laguerre, não houve uma grande diferença entre os valores de MSE. Entretanto, o uso das funções de Kautz não se justificam neste caso devido ao maior esforço computacional para gerar 10 funções necessárias para a modelagem, sendo que o modelo de Laguerre já atende o sistema com 2 funções. A Figura 17 mostra a comparação entre a resposta do modelo identificado com funções de Laguerre e a resposta do sistema real.

**Figura 17: Resultado da identificação com funções de Laguerre e AG.**



FONTE: Autoria Própria.

A função de transferência gerada que descreve este sistema é mostrada na equação (37).

$$\hat{G}(z) = \frac{-0,0449z^2 + 0,1210z - 0,0717}{z^3 - 2,643z^2 + 2,328z - 0,6838}. \quad (37)$$

Em prosseguimento das experimentações, foi realizada a modelagem usando como otimizador o algoritmo PSO. Os parâmetros para esta ferramenta são mostrados na Tabela 3.

**Tabela 3: Parâmetros para modelo PSO.**

Parâmetros	Valor
População	150
Nº Funções	Aleatório
$\omega$	0,8
$c_1$	1
$c_2$	1

FONTE: Autoria Própria.

Os melhores resultados da implementação podem ser vistos na Tabela 4. A Figura 18 mostra a resposta do sistema identificado com funções de Laguerre e a resposta do sistema real amostrado.

**Tabela 4: Resultados com o otimizador PSO.**

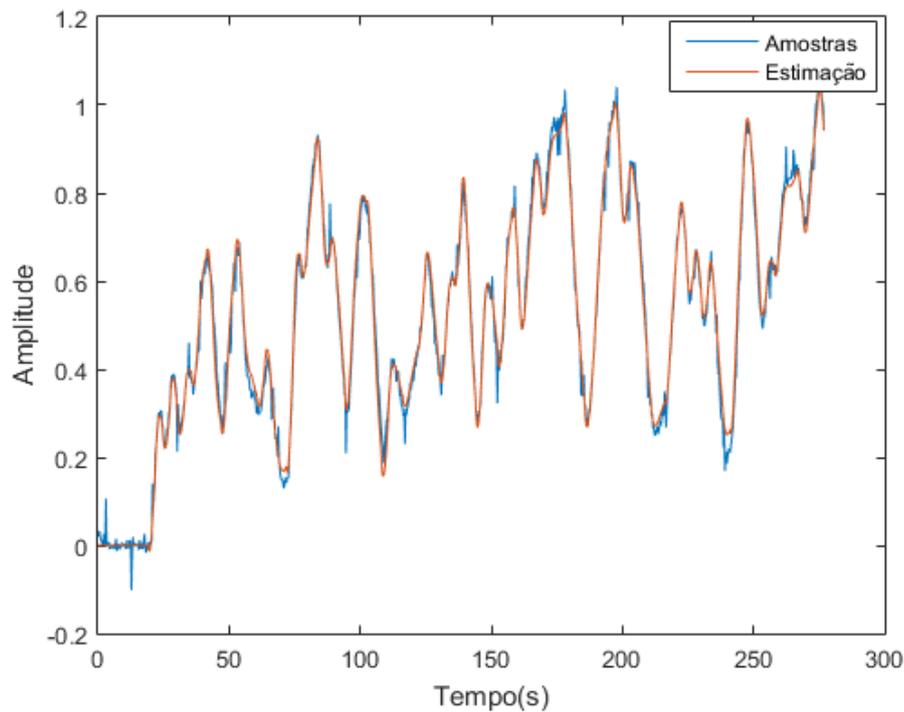
Função	Nº Funções	Polos	MSE
Laguerre	3	0,899	$6,98 \times 10^{-4}$
Kautz	6	$-0,353 \pm 0,683i$	$1,40 \times 10^{-3}$

FONTE: Autoria Própria.

A função de transferência gerada que descreve este sistema é mostrada na equação (38).

$$\hat{G}(z) = \frac{-0,0517z^5 + 0,2670z^4 - 0,5761z^3 + 0,5910z^2 - 0,2990z + 0,0599}{z^6 - 5,399z^5 + 12,15z^4 - 14,57z^3 + 9,837z^2 - 3,541z + 0,5310}. \quad (38)$$

**Figura 18: Resultado da identificação com funções de Laguerre e PSO.**

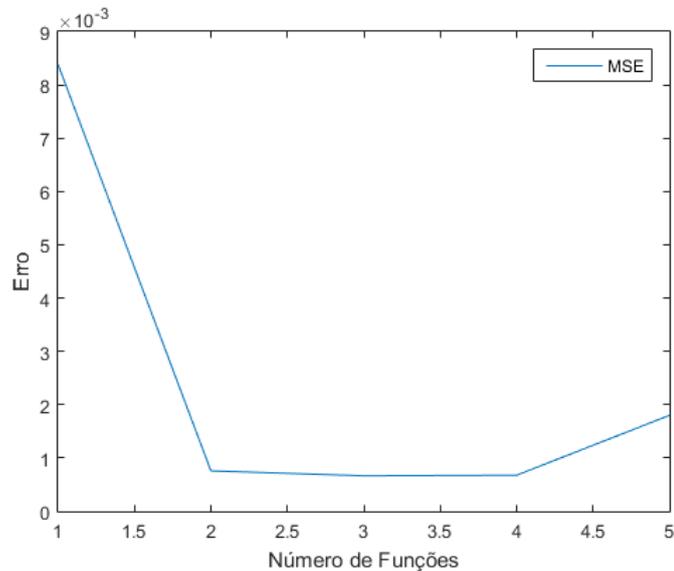


FONTE: Autoria Própria.

Por meio do conjunto de resultados que se obteve neste último teste, foi possível concluir que os polos deste sistema se encontram entre 0,85 e 0,95. A partir de então, os testes foram feitos com a geração de polos dentro deste raio e com as funções de Laguerre. Com isto, levantou-se um gráfico para especular se o aumento do número das funções que formam a base iriam ter fortes implicações nos resultados. O resultado deste procedimento é mostrado na Figura 19.

É notório que não houve grande melhora nos resultados a partir de duas funções. Segundo Campello et al. (2007), isto ocorre devido ao polo já estar bem posicionado em relação ao polo real do sistema e em decorrência do modelo escolhido não conseguir representar a não linearidade do sistema que, apesar de ser pouca, limitou os resultados a uma ordem de grandeza de  $10^{-4}$ .

**Figura 19: Comportamento do MSE com o aumento das funções da base.**



FONTE: Autoria Própria.

Na sequência, foi realizada a validação do otimizador ACO. Os parâmetros para este algoritmo são mostrados na Tabela 5.

**Tabela 5: Parâmetros para modelo ACO.**

Parâmetros	Valor
Nº de Funções	Aleatório
$\rho$	0,05
$\alpha$	1
Nº de Formigas	25

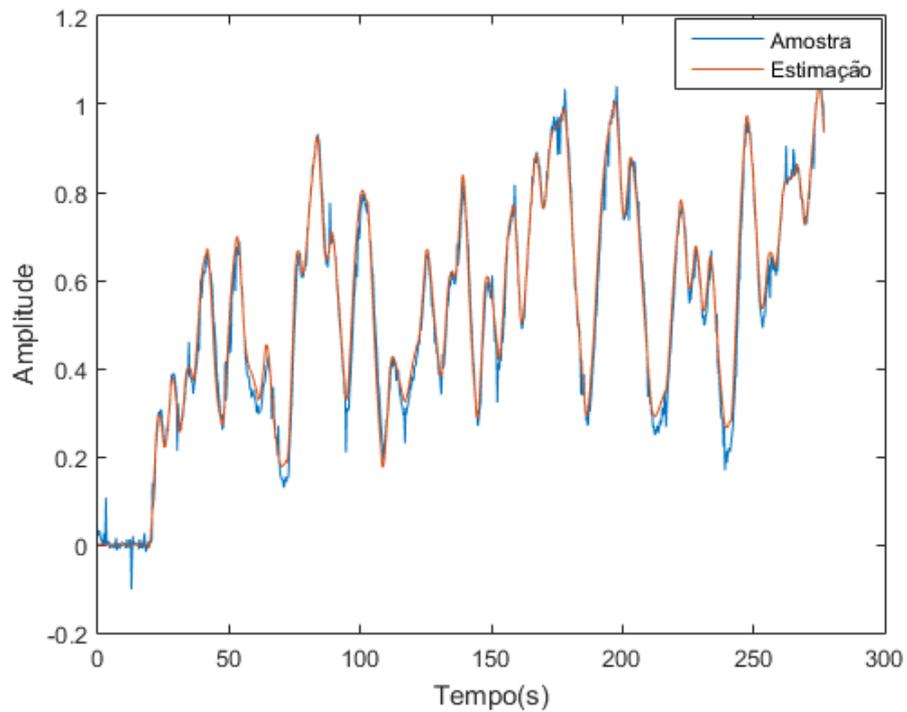
FONTE: Autoria Própria.

Os resultados desta experimentação são vistos na Tabela 6. Assim como no otimizador PSO, não foram realizados experimentos com as funções FIR, porque foi considerado desnecessário, uma vez que no AG já havia ficado claro que, para este sistema, os resultados não são satisfatórios. É possível observar que o módulo dos pólos utilizados para parametrizar as funções de Kautz e Laguerre encontram-se na região estimada anteriormente. O resultado gráfico deste teste é exibido na Figura 23, em que são vistas a resposta do sistema real amostrado e a resposta do sistema modelado com as funções de Laguerre.

**Tabela 6: Resultados com o otimizador ACO.**

Função	Nº de Funções	Polos	MSE
Laguerre	2	0,878	$1,1 \times 10^{-3}$
Kautz	6	$0,897 \pm 0,395i$	$8 \times 10^{-3}$

FONTE: Aatoria Própria.

**Figura 20: Resultado da identificação com funções de Laguerre e ACO.**

FONTE: Aatoria Própria.

A função de transferência gerada que descreve este sistema é mostrada na equação (39). Também pode ser visto na Tabela 7 o comparativo entre os melhores resultados dos três algoritmos de otimização heurística.

$$\hat{G}(z) = \frac{-0,0293z^2 + 0,0939z - 0,0599}{z^3 - 2,634z^2 + 2,312z - 0,6767}. \quad (39)$$

**Tabela 7: Melhor resultado de cada método heurístico.**

Função	Nº de Funções	Polos	Ganhos	MSE
AG - Laguerre	2	0,881	[0,253 0,395]	$7,97 \times 10^{-4}$
PSO - Laguerre	3	0,899	[0,288 0,380 -0,080]	$6,98 \times 10^{-4}$
ACO - Laguerre	2	0,878	[0,278 0,387]	$1,1 \times 10^{-3}$

FONTE: Aatoria Própria.

Para a validação dos algoritmos determinísticos utilizou-se como base os resultados alcançados por meio dos testes com os algoritmos heurísticos. Foram utilizados os mesmos dados amostrados da planta e, na modelagem, foram usadas três funções de Laguerre para formar a base. Os valores iniciais de ganhos e de polo para teste foram baseados nos resultados alcançados pelo algoritmo PSO. A Tabela 8 mostra os parâmetros utilizados nos algoritmos.

**Tabela 8: Parâmetros dos algoritmos determinísticos.**

Algoritmo	Ganhos	Polo	Iterações	Passo
GD				0,2
GN	[0,3 0,4 -0,001]	0,85	300	-
LM				-

FONTE: Aatoria Própria.

Com estes parâmetros inseridos nos respectivos algoritmos, alcançou-se os resultados mostrados na Tabela 9.

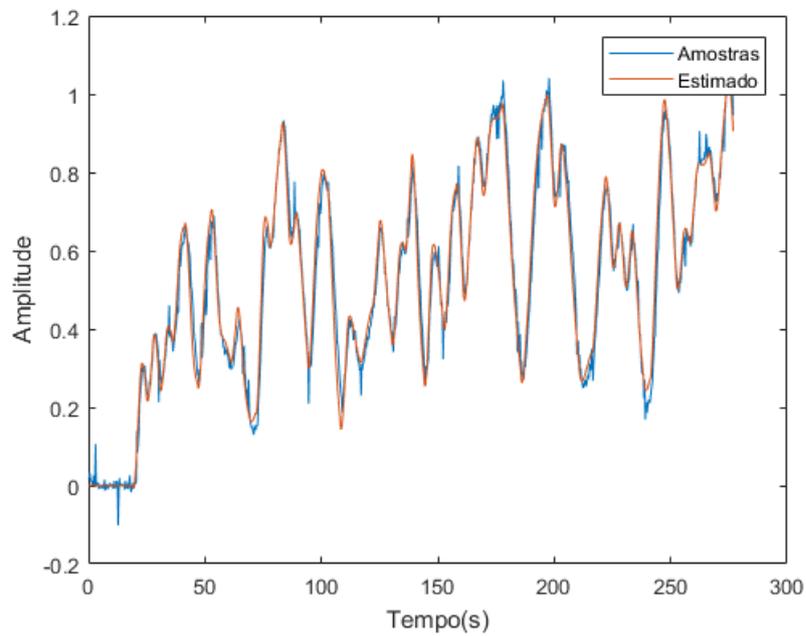
**Tabela 9: Resultados dos algoritmos determinísticos.**

Algoritmo	Ganhos	Polo	MSE
GD	[0,292 0,392 -0,009]	0,870	$1,80 \times 10^{-3}$
GN	[0,299 0,399 -0,001]	0,869	$2,10 \times 10^{-3}$
LM	[0,256 0,397 -0,021]	0,886	$7,63 \times 10^{-4}$

FONTE: Aatoria Própria.

Pode-se observar que para um mesmo conjunto de parâmetros iniciais, o LM foi o que apresentou o melhor desempenho. Este resultado é decorrente do mesmo combinar estratégias de melhoramento nos pontos em que o GD e o GN apresentam um menor desempenho, como já exposto no Capítulo 3. As Figuras 21, 22 e 23 mostram os resultados de cada otimizador e, na sequência, as equações (40), (41) e (42) apresentam as funções de transferência que podem representar a planta. Como pode-se observar, no decorrer desta seção foram mostradas diferentes funções que modelam o mesmo sistema, fortalecendo a afirmação feita na Seção 2.1 de que um mesmo sistema pode ser representado por diferentes modelos.

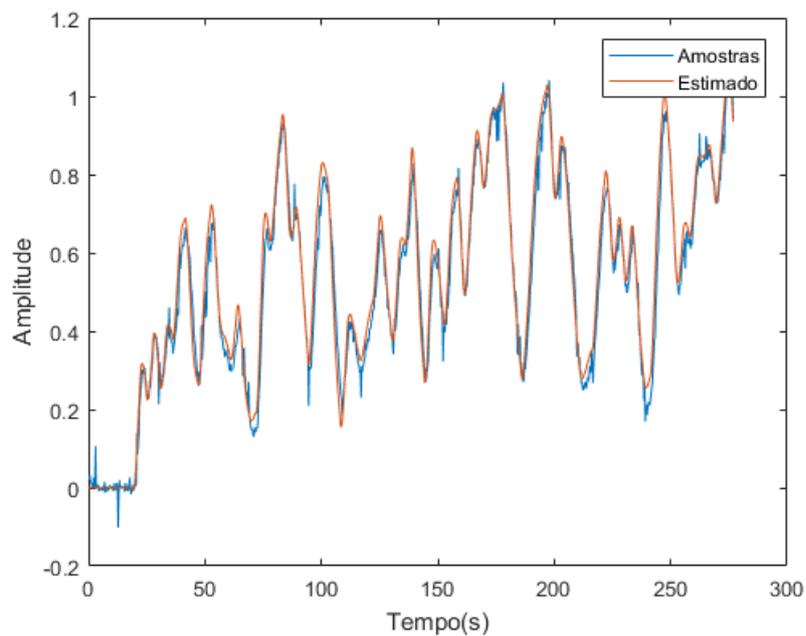
**Figura 21: Resultado da identificação com funções de Laguerre e GD.**



FONTE: Autoria Própria.

$$\hat{G}(z) = \frac{-0,028z^5 + 0,1690z^4 - 0,378z^3 + 0,404z^2 - 0,208z + 0,0419}{z^6 - 5,22z^5 + 11,4z^4 - 13,2z^3 + 8,59z^2 - 2,99z + 0,434}. \quad (40)$$

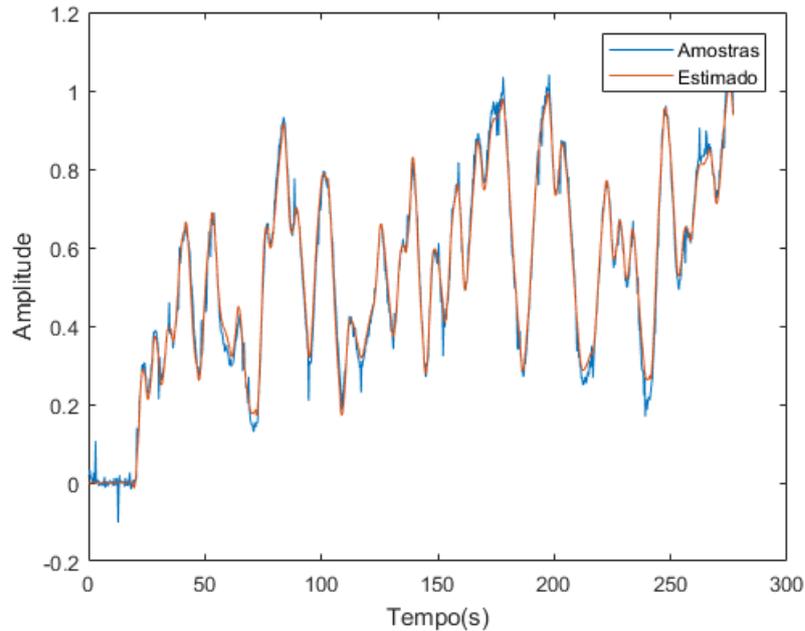
**Figura 22: Resultado da identificação com funções de Laguerre e GN.**



FONTE: Autoria Própria.

$$\hat{G}(z) = \frac{-0,0241z^5 + 0,153z^4 - 0,351z^3 + 0,378z^2 - 0,196z + 0,039}{z^6 - 5,22z^5 + 11,3z^4 - 13,2z^3 + 8,58z^2 - 2,98z + 0,432}. \quad (41)$$

**Figura 23: Resultado da identificação com funções de Laguerre e LM.**



FONTE: Autoria Própria.

$$\hat{G}(z) = \frac{-0,052z^5 + 0,274z^4 - 0,562z^3 + 0,567z^2 - 0,282z + 0,055}{z^6 - 5,31z^5 + 11,8z^4 - 13,8z^3 + 9,23z^2 - 3,27z + 0,483}. \quad (42)$$

O último otimizador a ser testado foi o algoritmo híbrido. Com já descrito, este otimizador é uma combinação entre o PSO e o LM, sendo assim, ele não pode ser classificado com um algoritmo determinístico. Este fato implica em que pode ser alcançado resultados diferentes para os mesmos parâmetros de entrada. Nestes testes, também foram utilizados os dados coletados da planta e as funções de Laguerre. Os parâmetros iniciais são mostrados na Tabela 10.

**Tabela 10: Parâmetros do algoritmo híbrido.**

Parâmetros	Valores
Nº de Funções	3
$\omega$	0,8
$c_1$	1
$c_2$	1

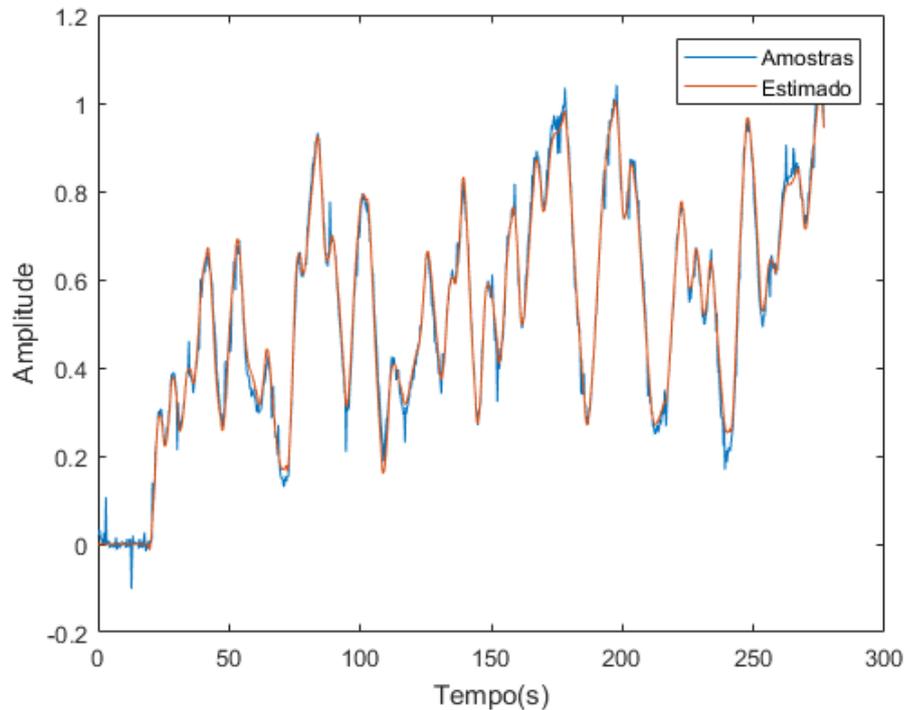
FONTE: Autoria Própria.

Com estes parâmetros inseridos no algoritmo, o melhor resultado obtido é mostrado na Tabela 11. A Figura 24 ilustra o resultado visual da identificação e a função de transferência é expressa na equação (43).

**Tabela 11: Resultados dos algoritmos determinísticos.**

Algoritmo	Ganhos	Polo	MSE
Híbrido	[0,294 0,370 -0,088]	0,904	$6,82 \times 10^{-4}$

FONTE: Autoria Própria.

**Figura 24: Resultado da identificação com funções de Laguerre e o algoritmo híbrido.**

FONTE: Autoria Própria.

$$\hat{G}(z) = \frac{-0,048z^5 + 0,259z^4 - 0,546z^3 + 0,564z^2 - 0,286z + 0,058}{z^6 - 5,42z^5 + 12,3z^4 - 14,8z^3 + 10z^2 - 3,62z + 0,544}. \quad (43)$$

Para finalizar os testes, foi realizado um comparativo entre os algoritmos PSO, LM e híbrido. O experimento consistiu em gerar um conjunto de dez soluções com o algoritmo híbrido e um segundo conjunto de dez soluções com o PSO. Para encontrar estes resultados os algoritmos foram parametrizados com os valores da Tabela 3, mas foram usadas três funções de Laguerre ao invés de valores aleatórios. Com o grupo de soluções, foi calculada a média aritmética dos valores dos MSE para usá-las como parâmetros métricos. Já para fazer um comparativo com o LM, utilizou-se do melhor resultado encontrado, expresso na Tabela 9. A Tabela 12 mostra os resultados alcançados.

Apesar dos algoritmos terem alcançados resultados que estão na mesmo ordem de grandeza,  $10^{-4}$ , o algoritmo híbrido foi aquele, que na média, atingiu o melhor desempenho. Isto decorre do mesmo combinar técnicas presentes nos outros dois algoritmos.

**Tabela 12: Comparativo dos resultados.**

Algoritmo	MSE
Híbrido	$7,05 \times 10^{-4}$
PSO	$7,42 \times 10^{-4}$
LM	$7,63 \times 10^{-4}$

FONTE: Aatoria Própria.

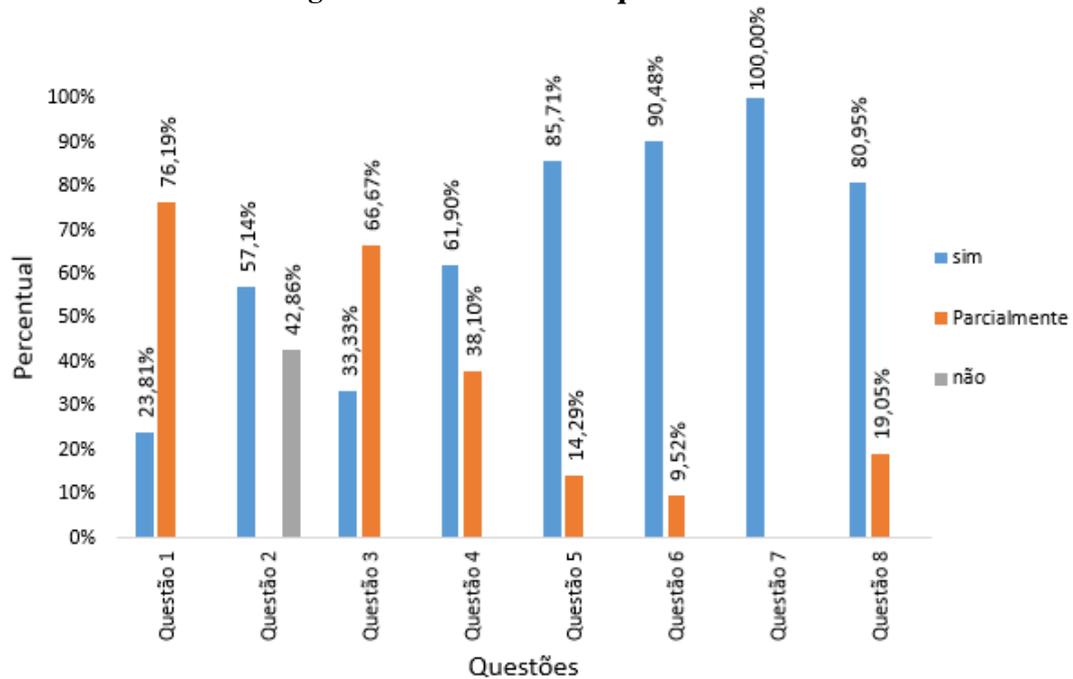
## 6.2 RESULTADOS PEDAGÓGICOS

Com a *toolbox* implementada foi realizado um experimento em sala de aula como descrito no Capítulo 5. No Apêndice A, encontra-se o formulário que os vinte e um alunos responderam. Na Figura 25 é exibido um gráfico com os resultados das respostas para cada uma das oito questões objetivas. As duas primeiras questões são relativas ao primeiro momento do experimento, em que os alunos não tinham nenhum contato prévio com a *toolbox*. Como é mostrado no gráfico, 76,19% dos alunos demoraram para entender o que a *toolbox* estava fazendo e 42,86% responderam que não conseguiriam operar o software sem o auxílio de um professor. Estes dois resultados sinalizam uma certa necessidade de aprimoramento no *software*, principalmente se agrupado com os 66,67% das respostas da questão 3 que indicam que a interface é razoavelmente intuitiva.

Na questão 4, os alunos responderam que compreenderam as informações que os menus de ajuda forneceram em 61,90% dos casos, entretanto, ainda existe espaço para o melhoramento destes recursos. A questão 5 apresentou 85,71% de resposta positiva em relação ao aumento do interesse do aluno pela disciplina devido a existência da *toolbox*. Este resultado

ganha força se cruzado com os 90,48% dos alunos que responderam a questão 6 indicando que a ferramenta deixou mais clara a aplicação da identificação de sistemas e a unanimidade nas respostas da questão 7 em dizer que o *software* ajudaria na compreensão da disciplina. A questão 8 apresentou 80,95% de respostas positivas para a compatibilidade da estrutura do laboratório com a *toolbox*. Isto é importante já que o desempenho dos algoritmos está ligado com a capacidade do *hardware* em que se encontra instalado.

**Figura 25: Resultado do questionário.**



FONTE: Autoria Própria.

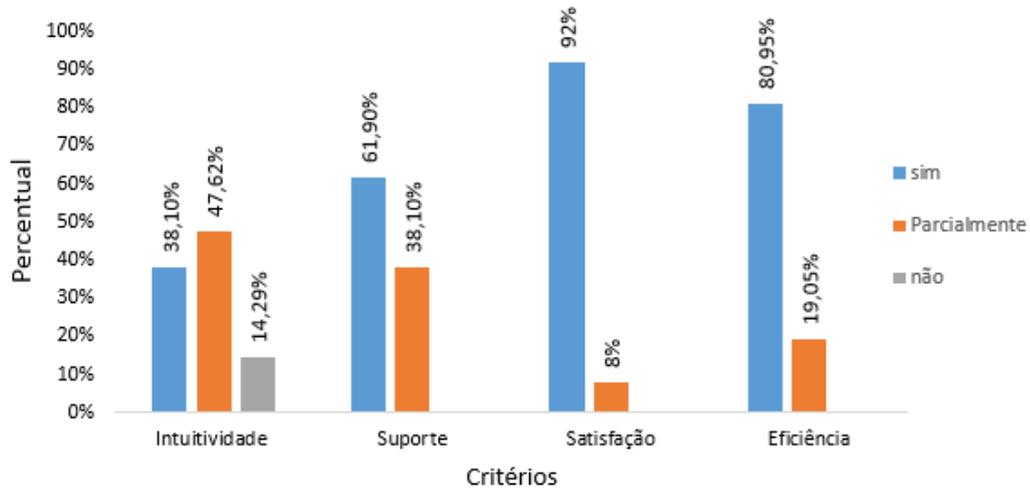
As questões ficaram divididas nos seguintes critérios:

- Intuitividade - questões de 1 a 3;
- Suporte - questão 4;
- Satisfação - questões de 5 a 7;
- Eficiência - questão 8.

Para realizar a análise do desempenho de cada critério, foi elaborado um segundo gráfico ilustrado na Figura 26. Para chegar nestes resultados, foram utilizadas as respostas de cada questão e colocadas as contribuições de cada resposta dentro de seu respectivo critério,

ou seja, para os critérios de intuitividade e satisfação, que foram feitas mais de uma pergunta, utilizou-se de média aritmética para chegar ao percentual de cada resposta.

**Figura 26: Resultado para cada critério.**



FONTE: Autoria Própria.

Como pode ser analisado na Figura 26, os critérios de eficiência e satisfação apresentaram desempenho positivo em 80,95% e 92% casos, respectivamente. O critério suporte apresentou um resultado mais positivo, entretanto, para 38,10% o parâmetro foi atingido parcialmente, indicando que este critério pode ser aperfeiçoado. Isto torna-se mais evidente quando cruzado com as respostas da questão nove, na qual os alunos dissertaram criticando principalmente o menu de ajuda. Os principais pontos citados pelos mesmos são: a necessidade de exemplos, a necessidade do modelo dos algoritmos e uma melhor clareza nas explicações. No critério de intuitividade teve-se um desempenho inferior aos demais. Contudo, quando cruzado com respostas da questão nove, diversos alunos citam que a partir da exposição oral feita sobre o conteúdo ficou mais clara a operação da ferramenta. O principal ponto levantado foi que após a explicação pôde-se realizar uma melhor parametrização dos algoritmos.

### 6.3 SÍNTESE

Neste capítulo, buscou-se validar a teoria apresentada no trabalho. Para a planta utilizada na validação, as funções de Laguerre foram as que melhor se adequaram ao sistema. Este resultado já era esperado, pois, para sistemas superamortecidos, estas funções apresentam uma melhor capacidade de representação. Com relação aos algoritmos de otimização heurísticos, o que apresentou o melhor desempenho, no quesito custo computacional

e resultado obtido, foi o algoritmo PSO. Acredita-se que isto tenha ocorrido devido à forma de concepção dos algoritmos. Os algoritmos AG e ACO são definidos para sistemas discretos e o PSO é definido para sistemas contínuos e, como o problema de otimização apresentado é um problema contínuo, o melhor desempenho do PSO é compreensível.

Entre os algoritmos determinísticos, o LM foi o que apresentou o melhor desempenho. Isto já era esperado, pois este algoritmo combina o que há de melhor dos dois outros algoritmos (GD e GN). É importante complementar que os algoritmos determinísticos possuem o risco de convergirem para mínimos locais e o resultado alcançado depende muito dos valores iniciais. Tendo isto como contraponto, o desenvolvimento do algoritmo híbrido é vantajoso por utilizar os resultados obtidos pela busca feita com o PSO como valores iniciais e, na sequência, realizar um refinamento do resultado com o LM. Como pode ser visto na Tabela 12, o algoritmo híbrido foi aquele que conseguiu atingir o melhor desempenho nos teste. Entretanto, é válido ressaltar que os resultados da otimização com este algoritmo estão ligados aos resultados da busca realizada pelo PSO, podendo nem sempre gerar um resultado desejável.

No desenvolvimento da *toolbox*, buscou-se implementar uma ferramenta que auxilie o professor no desenvolvimento das atividades em sala de aula. Desta forma, o fato do quesito intuitividade não ter apresentado um resultado muito positivo não deve ser encarado com muita preocupação, já que as atividades normalmente serão desenvolvidas sob orientação e supervisão do professor. Apesar da aplicação possuir alguns pontos para serem melhorados, acredita-se que a ferramenta atingiu o que foi proposto. Esta conclusão baseia-se no resultado da satisfação dos alunos, que alcançou resultado positivo de 92%.

## 7 CONSIDERAÇÕES FINAIS

Neste trabalho, um dos principais desenvolvimentos foi a fundamentação teórica referente às funções ortonormais e aos algoritmos de otimização. Estes dois assuntos foram estudados, a fim de estruturar a viabilidade destas ferramentas para serem utilizadas na identificação de sistemas. As funções ortonormais permitem um mapeamento estático de vetores no espaço de Hilbert, o que possibilita a aproximação de modelos reais por intermédio de bases ortonormais e combinações lineares ou não lineares. A utilização destas bases se justifica devido ao custo computacional empregado em seu desenvolvimento. Outra vantagem é a inserção do conhecimento *a priori* para a modelagem, pois, para sistemas subamortecidos, superamortecidos ou com polos próximos à origem, pode-se fazer a escolha de qual será a função ortonormal (FIR, Kautz e Laguerre) que apresentará o melhor custo benefício para a representação.

As equações utilizadas para as aproximações dos sistemas apresentam ganhos escalares à serem definidos. Desta forma, esta tarefa é realizada por meio dos algoritmos de otimização. Estes foram divididos em duas classes: determinísticos e heurísticos e uma combinação de ambos foi desenvolvida. Cada classe apresenta suas vantagens e desvantagens. Contudo, o objetivo neste trabalho foi gerar uma gama de opções ao usuário, para que este escolha qual algoritmo é mais adequado à sua aplicação.

Na sequência, foi realizada a validação do uso das funções ortonormais para identificação de um sistema real que, por meio do critério qualitativo adotado, o MSE, alcançaram erros na ordem de  $10^{-4}$ . Como isso, considera-se que foi obtida uma modelagem satisfatória, apesar de ainda haver espaços para melhoramentos. Com estes resultados, pôde ser feito o avanço para uma etapa que se constituiu da implementação de uma *toolbox* para identificação de sistemas que utilize as OBFs em uma plataforma GUI.

O desenvolvimento da *toolbox* foi baseado em critérios de avaliação de SE. Para validar a implementação da ferramenta, realizou-se um experimento com uma turma de Introdução à Identificação de Sistemas, no qual analisou-se quatro critérios: intuitividade,

suporte, satisfação e eficiência. Apesar de haver espaço para melhoramentos do *software*, com os resultados atingidos (intuitividade de 38,10%, suporte de 61,90%, satisfação de 92% e eficiência de 80,95%) acredita-se que a ferramenta apresenta potencial para ser utilizada como uma ferramenta de auxílio ao professor nas atividades realizadas na disciplina.

## APÊNDICE A – QUESTIONÁRIO APLICADO EM SALA DE AULA

Este questionário é referente a atividade desenvolvida em sala na disciplina de Identificação de Sistemas. As respostas aqui preenchidas serão utilizadas para o aprimoramento da *toolbox* e, farão parte da análise dos resultados no Trabalho de Conclusão de Curso do aluno Higor Serafin.

Por favor preencher os campos com um “X” nas alternativas que melhor se enquadrem com a sua opinião. Na questão 9 responda com um pequeno parágrafo redigido à mão. Não será necessário que o indivíduo que está preenchendo o formulário identifique-se.

**1** – No primeiro momento da atividade como você reagiu?

- Compreendi logo o que a *toolbox* estava fazendo;
- Demorei para entender o que a *toolbox* estava fazendo;
- Não consegui compreender o que estava acontecendo.

**2** – Na sua opinião, você conseguiria operar a *toolbox* sem o auxílio de um professor?

- Sim;
- Não.

**3** – A interface da *toolbox* é intuitiva e clara?

- Bem intuitiva, sem dificuldades para operar;
- Razoavelmente intuitiva;
- Nada intuitiva.

**4** – Os menus de Ajuda são claros e ajudaram no uso do software?

- Sim, compreendi as informações que os menus me deram;
- Sim, mas nem todas as informações estavam claras;
- Não consegui compreender as instruções.

**5** – Na sua opinião, a toolbox aumentou seu interesse pela disciplina?

- Sim;
- Manteve-se igual;
- Não.

**6** – Com as atividades realizadas, na sua opinião, está mais claro a aplicação da Identificação de sistemas?

- Sim;
- Manteve-se igual;
- Não.

**7** – Na sua opinião, a toolbox ajudaria na compreensão da disciplina?

- Sim;
- Talvez;
- Não.

**8** – A toolbox é compatível com a estrutura computacional presente no laboratório?

- Sim, e apresenta boa performance;
- Sim, mas apresenta uma baixa performance;
- Não.

**9** – Na sua opinião, o que poderia ser melhorado na Toolbox?

## REFERÊNCIAS

- AGUIRRE, L. A. **Introdução à Identificação de Sistemas**. 3. ed. UFMG, 2007.
- AGUIRRE, L. A. **Introdução à Identificação de Sistemas**. 4. ed. UFMG, 2010.
- ALAM, M. N. Particle swarm optimization: Algorithm and its codes in matlab. **Department of Electrical Engineering, Indian Institute of Technology, Roorkee**, 2016.
- BELT, H. J. **Orthonormal Bases for Adaptive Filtering**. 1997.
- BOECHEL, T. **Algoritmo de otimização: uma abordagem híbrida utilizando o algoritmo das formigas e genético**. 2003. Tese (Doutorado) — Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação., 2003.
- BRANDOLT, H. G. et al. Simulação de escoamento em dutos por caracterização de eventos. Florianópolis, SC, 2002.
- CAMPELLO, R. J.; OLIVEIRA, G. H.; AMARAL, W. C. Identificação e controle de processos via desenvolvimentos em séries ortonormais. parte a: Identificação. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, SciELO Brasil, v. 18, n. 3, p. 301–321, 2007.
- CAMPELLO, R. J. G. B.; OLIVEIRA, G. H. C. **Modelos não lineares**. Edgard Blücher, 2007.
- CAUCHY, A. Méthodes générales pour la résolution des systèmes d'équations simultanées. **C.R. Acad Sci Par**, 1847.
- CORRÊA, M. V.; AGUIRRE, L. A. Identificação não-linear caixa-cinza: Uma revisão e novos resultados. **Controle Automação**, 2004.
- DEISTLER, M. **Stochastic Theory and Control**. Springer, 2002.
- DORIGO, K. S. . M. **An Introduction to Ant Colony Optimization**. IRIDIA - Technical Report Series, 2006.
- DORIGO, T. S. . M. **Ant Colony Optimization**. The MIT Press, 2004.
- EBERHART, R.; KENNEDY, J. A new optimizer using particle swarm theory. **Micro Machine and Human Science**, 1995.
- FLOUDAS, C. A. **Deterministic Global Optimization: Theory, Methods and Applications**. Springer Science+Business Media Dordrecht, 2000.
- FRITSCHÉ, G.; OLIVEIRA, P. M.; PIRES, E. E. S.; FILHO, P. L. de P. Algoritmo híbrido de otimização glowworm-boids. **Revista Eletrônica Científica Inovação e Tecnologia**, v. 2, n. 2, p. 2–6, 2014.

- FU, Y.; DUMONT, G. An optimum time scale for discrete laguerre network. In: **IEEE TRANSACTIONS AUTOMATIC CONTROL**. 1993.
- GARCIA, C. E.; PRETT, D. M.; MORARI, M. Model predictive control: theory and practice—a survey. **Automatica**, Elsevier, v. 25, n. 3, p. 335–348, 1989.
- GAUSS, C. Theory of the motion of the heavenly bodies moving about the sun in conic sections (translated by c. h. davis, 1963). **New York: Dover**, 1809.
- GIRAFFA, L. M. M. **Uma arquitetura de tutor utilizando estados mentais**. 1999. Tese (Doutorado) — UFRGS, 1999.
- GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Addison-Wesley Publishing Company, INC, 1989.
- HASSAN, B. C. R.; WECK, O. de. A comparison of particle swarm optimization and the genetic algorithm. **46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference**, 2005.
- HEUBERGER, P. S.; HOF, P. M. van den; WAHLBERG, B. **Modelling and identification with rational orthogonal basis functions**. Springer Science & Business Media, 2005.
- HEUBERGER, P. S.; HOOG, T. J. de; HOF, P. M. Van den; WAHLBERG, B. Orthonormal basis functions in time and frequency domain: Hambo transform theory. **SIAM Journal on Control and Optimization**, SIAM, v. 42, n. 4, p. 1347–1373, 2003.
- HSIA, T. C. **System Identification**. 1997.
- ISERMANN, R.; MÜNCHHOF, M. **Identification of Dynamic Systems**. Springer, 2011.
- KAUTZ, W. Transient synthesis in the time domain. **IRE Transactions on Circuit Theory**, 1954.
- KOKASH, N. An introduction to heuristic algorithms. **Department of Informatics and Telecommunications University of Trento, Italy**, 2018.
- KOSCIANSKI, A.; SOARES, M. d. S. Qualidade de software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software.[sl]: Novatec, 2007. **Citado na**, p. 26, 2007.
- LATHI, B. **Sinais e Sistemas Lineares**. Bookman, 2007.
- LEMMA, D.; RAMASAMY, M.; SHUHAIMI, M. System identification using orthonormal basis filters. **Journal of Applied Sciences(Faisalabad)**, Asian Network for Scientific Information, 308-Lasani Town Sargodha Rd Faisalabad 38090 Pakistan, v. 10, n. 21, p. 2516–2522, 2010.
- LEVENBERG, K. A method for the solution of certain problems in least squares. **Quart. Appl. Math. Vol 2**, 1944.
- LIMA, E. O. Algoritmo genético híbrido aplicado à otimização de funções. **Trabalho de conclusão de curso, UFES**, 2008.
- LJUNG, L. **System Identification: Theory for the User**. Prentice Hall, 1999.

- LJUNG, L. Perspectives on system identification. **Annual Reviews in Control**, 2010.
- LUCHI, F. **Um Algoritmo Híbrido entre Evolução Diferencial e Nelder-Mead Usando Entropia para Problemas de Otimização Não-Linear Inteiro Misto**. 2016. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2016.
- MACHADO, R. C. J.; AMARAL, W. Modelos lineares gobf com funções internas. In: **VIII Congresso Brasileiro de Automática**. 2010.
- MALMQUIST, F. Sur la determination d'une classe de fonctions analytiques par leurs valeurs dans un ensemble donné de points. **Comptes Rendus du Sixième Congrès des Mathématiciens Scandinaves**, 1925.
- MARQUARDT, D. An algorithm for least-squares estimation of nonlinear parameters. **SIAM J. Appl. Math Vol 1**, 1963.
- MEZA, J. C. Steepest descent. **WIREs Comp Stat**, 2010.
- MILLER, B. L.; GOLDBERG, D. E. Genetic algorithms, tournament selection, and the effects of noise. **Department of General Engineering, University of Illunois at Urbana-Champaign, USA**, 1995.
- NIELSEN, J. **Usability engineering**. Elsevier, 1994.
- NINNESS, B.; GOMEZ, J.-C.; WELLER, S. MIMO system identification using orthonormal basis functions. In: IEEE. **Decision and Control, 1995., Proceedings of the 34th IEEE Conference on**. 1995. v. 1, p. 703–708.
- NINNESS, B.; GUSTAFSSON, F. A unifying construction of orthonormal bases for system identification. **IEEE Trans. Automat. Control**, 1994.
- OGATA, K. **Engenharia de Controle Moderno**. LTC, 1998.
- OLIVEIRA, C. C. de; COSTA, J. W. da; MOREIRA, M. **Ambientes informatizados de aprendizagem: produção e avaliação de software educativo**. Papirus, 2001.
- OLIVEIRA, C. R. de. **Introdução à Análise Funcional**. IMPA, 2008.
- OLIVEIRA, K. A.; AMARAL, M. A.; DOMINGOS, G. R. A avaliação do uso de objetos de aprendizagem na educação de jovens e adultos. **Revista Brasileira de Informática na Educação**, v. 19, n. 03, p. 53, 2011.
- OROSKI, E. Identificação de sistemas não lineares utilizando modelos narx, funções ortonormais e otimização heurística. 2015.
- OROSKI, J. R. D. M. E.; BAUCHSPIESS, A. Identificação de um levitador magnético não linear a partir de funções de kautz e algoritmos genéticos. In: **Congresso Nacional de Matemática Aplicada à Indústria**. 2014.
- PACHECO, M. A. C. Algoritmos genéticos: Princípios e aplicações. **ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro.**, 1999.

- PEREIRA, W. S.; FILHO, R. J. C.; SILVA, W.; SILVA, R. S. T. D.; DANTAS, V. F.; AGUIAR, Y. Validação de uma abordagem combinada para avaliação de software educativo: Avanços e desafios. **Revista Tecnologias na Educação—Ano**, 2016.
- RAMOS, E. M.; MENDONÇA, I. J. O fundamental na avaliação da qualidade do software educacional. **Edla Ramos: Publicações**, 1991.
- RANGANATHAN, A. The levenberg-marquardt algorithm. 2004.
- SANTOS, G. S. **ESPAÇOS DE HILBERT**. 2008.
- SILVA, A. C. B. da. **Softwares Educativos: Critérios de Avaliação a partir dos Discursos da Interface, da Esfera Comunicativa e do Objeto de Ensino**. 2012. Tese (Doutorado) — UFPE, 2012.
- STRANG, G. **Álgebra Linear e suas Aplicações**. Cengage Learning, 2010.
- TAKENAKA, S. On the orthogonal functions and a new formula of interpolation,. **J. Math.**, 1925.
- TALBI, E.-G. **Metaheuristics**. Wiley, 2009.
- WAHLBERG, B. System identification using laguerre models. **IEEE TRANSACTIONS ON AUTOMATIC CONTROL**, 1991.
- WAHLBERG, B. System identification using kautz model. In: **IEEE TRANSACTIONS AUTOMATIC CONTROL**. 1994.
- WAHLBERG, B. Orthonormal basis function models: A transformation analysis. In: **14th Triennial World Congress**. 1999.
- WAHLBERG, B.; MÄKILÄ, P. On approximation of stable linear dynamical systems using laguerre and kautz functions. In: **Automatica**. 1996.
- WALLIN, R. **Optimization Algorithms for System Analysis and Identification**. 2004. Dissertação (Mestrado) — Link öping University, 2004.
- WANG, Y. Gauss-newton method. **Wires Computational Statistics**, 2012.
- WRIGHT, J. N. S. **Numerical Optimization**. New York:Springer, 1999.