

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALGORITMO DIFERENCIAL EVOLUTIVO APLICADO À IDENTIFICAÇÃO DE SISTEMAS DINAMICOS NÃO-LINEARES

JOSUÉ SILVA DE MORAIS

JANEIRO
2013

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

M827 Morais, Josué Silva de, 1981-
2013 Algoritmo diferencial evolutivo aplicado à identificação de sistemas dinâmicos nãolineares / Josué Silva de Morais. - 2013.
128 f. : il.

Orientador: Fábio Vincenzi Romualdo da Silva.
Tese (doutorado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.
Inclui bibliografia.

1. Engenharia elétrica - Teses. 2. Sistemas não lineares - Teses. 3. Identificação de sistemas - Teses. 4. Métodos de simulação - Teses. I. Silva, Fábio Vincenzi Romualdo da, 1974- II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 621.3

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALGORITMO DIFERENCIAL EVOLUTIVO APLICADO À IDENTIFICAÇÃO DE SISTEMAS DINAMICOS NÃO-LINEARES

Tese apresentada por *Josué Silva de Morais* à Universidade Federal de Uberlândia para obtenção do título de Doutor em Engenharia Elétrica, aprovada em __/__/_____, pela seguinte banca examinadora:

Prof. Fábio Vincenzi R. da Silva, Dr. (UFU-FEELT)
Orientador
Prof. Luís Cláudio de Oliveira Lopes, Dr. (UFU-FEQ)
Prof. Ernane Alves Coelho, Dr. (UFU-FEELT)
Prof. Kleber Lopes Fontoura, Dr. (CEFET-ARAXA)
Prof. Edilberto Pereira Teixeira, Dr. (UNIUBE)

Uberlândia, ___ de _____ de 2013.

ALGORITMO DIFERENCIAL EVOLUTIVO APLICADO À IDENTIFICAÇÃO DE SISTEMAS DINAMICOS NÃO-LINEARES

JOSUÉ SILVA DE MORAIS

Tese apresentada por Josué Silva de Morais à Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica.

Prof. Fábio Vincenzi Romualdo da Silva, Dr.	**Prof. Alexandre Cardoso, Dr.**
(Orientador)	Coordenador do Curso de Pós-Graduação

A ciência sem a religião é manca.

A religião sem a ciência é cega.

Albert Einstein

DEDICATÓRIA

A Deus

dou graças.

Um pouco de ciência nos afasta de Deus. Muito, nos aproxima.

Louis Pasteur

Aos meus amados pais

Ilton Dias de Morais & Maria das Graças Silva de Morais

pelos incentivos, ensinamentos, amor, exemplos de vida e dedicação,

em todas as fases de minha vida.

Aos meus queridos irmãos

Aniel Silva de Morais

Antoniel Silva de Morais

Résia Silva de Morais

pelo companheirismo, amor e amizade.

À minha esposa querida

Tatiane Pereira Santos Morais

pelo amor, dedicação, companheirismo

e por estar sempre presente em minha vida.

Aos meus filhos maravilhosos

José Carlos Santos Morais, Marcelino Santos Morais e Levi Santos Morais

pela alegria que vocês proporcionam na minha vida.

AGRADECIMENTOS

A Deus, pela força e sabedoria.

À UFU, pela oportunidade de conclusão do doutorado, pelo apoio e infra-estrutura necessários para a conclusão deste trabalho.

Aos Professores Fábio Vicenzi Romualdo da Silva e João Batista Vieira Júnior, pela orientação, apoio, amizade, ensinamentos e confiança.

Ao Professor Luís Cláudio de Oliveira Lopes a quem considero meu segundo orientador.

Ao Professor Aniel Silva de Morais, meu irmão e professor, pelo estímulo e auxílio para a realização deste trabalho.

Aos Professores Carlos Bissochi Jr e Márcio Cunha pelo apoio e auxílios prestados.

Aos professores Fábio Vincenzi Romualdo da Silva, Luís Cláudio de Oliveira Lopes, Ernane Alves Coelho, Kleber Lopes Fontoura, e Edilberto Pereira Teixeira por contribuírem como membros de minha banca examinadora.

A meus pais e irmãos, pelo apoio e incentivo em todos os momentos de minha vida.

À Tatiane, minha esposa, pelos incentivos, compreensão e cumplicidade.

A todos aqueles que, cada qual à sua maneira e importância, contribuíram de forma direta ou indireta para que este trabalho se tornasse realidade.

RESUMO

O objetivo principal deste trabalho é contribuir na etapa de seleção da estrutura nos processos de identificação de sistemas utilizando técnicas de inteligência artificial, principalmente o algoritmo Diferencial Evolutivo (DE).

Para atingir este objetivo uma nova representação NARMAX é proposta onde expoentes com valores reais são utilizados. A avaliação do método proposto foi efetuada através de um aplicativo desenvolvido especificamente para este trabalho com o pacote QT utilizando a linguagem C++ e técnicas de programação paralela.

Diferentes tipos de sistemas artificiais foram testados para avaliar a metodologia proposta. Por fim esta metodologia é aplicada em um conversor cc-cc Boost^2-Buck e um reator bioquímico de polímero e os resultados são apresentados.

Palavras chaves: Diferencial Evolutivo, Identificação de Sistemas, NARMAX Expoente Reais.

ABSTRACT

The main objective of this work is to contribute in the structure's selection stage in system identification process using artificial intelligence techniques, especially the Differential Evolutionary (DE) algorithm.

To reach this goal a new NARMAX representation is proposed where exponents with real values are used. The evaluation of the proposed method was performed through an application developed specifically for this work with a QT package using C + + and parallel programming techniques.

Different kinds of artificial systems were tested to evaluate the proposed methodology. Finally the methodology is applied to a dc-dc converter Boost^2-Buck and a biochemical polymer reactor and the results are presented.

Keywords: Differential Evolution, Identification Systems, NARMAX real Exponent.

SUMÁRIO

CAPÍTULO 1 INTRODUÇÃO 1

1.1 - INTRODUÇÃO GERAL 1

1.1 - RELEVÂNCIA 5

1.2 - MOTIVAÇÃO 6

1.3 - APRESENTAÇÃO DO TRABALHO 8

CAPÍTULO 2 REVISÃO BIBLIOGRÁFICA 10

2.1 - TÉCNICAS DE IDENTIFICAÇÃO DE SISTEMAS 10

2.2 - SELEÇÃO DO MODELO 11

2.2.1 - MODELOS NÃO-LINEARES 12

2.2.1.1 - MODELO NCARMA (NONLINEAR CONTROLLED AUTO-REGRESSIVE MOVING AVERAGE) 13

2.3 - SELEÇÃO DE ESTRUTURA 15

2.3.1 - COMPUTAÇÃO EVOLUTIVA 16

2.3.1.1 - ALGORITMO DIFERENCIAL EVOLUTIVO (DE) 19

2.4 - ESTIMAÇÃO DE PARÂMETROS 21

2.5 - ÍNDICES DE AVALIAÇÃO 23

2.5.1 - SOMATÓRIO DO ERRO QUADRÁTICO – SSE (SUM OF SQUARED ERROR) 23

2.5.2 - PRINCÍPIO DA PARCIMÔNIA 23

2.5.3 - CRITÉRIO DE INFORMAÇÃO DE AKAIKE (AKAIKE INFORMATION CRITERION) 24

2.5.4 - CRITÉRIO DE INFORMAÇÃO BAYESIANA (BAYESIAN INFORMATION CRITERION) 25

2.5.5 - ERRO DE PREDIÇÃO 25

2.6 - CONSIDERAÇÕES FINAIS 26

CAPÍTULO 3 TÉCNICAS UTILIZADAS NA METODOLOGIA PROPOSTA DE IDENTIFICAÇÃO DE SISTEMAS 28

3.1 - VISÃO GERAL DA METODOLOGIA PROPOSTA 28

3.2 - DETALHAMENTO DAS ETAPAS 31

3.2.1 - ALOCAÇÃO DOS DADOS NO APLICATIVO 31

3.2.2 - GERAÇÃO DA POPULAÇÃO INICIAL 31

3.2.3 - GERAÇÃO DA POPULAÇÃO MUTANTE 34

3.2.4 - CRUZAMENTO 37

3.2.5 - SELEÇÃO DOS REGRESSORES 37

3.2.6 - CÁLCULO DOS COEFICIENTES 38

3.2.7 - SELECAO DOS CROMOSSOMOS 38

3.2.8 - CRITÉRIO DE PARADA 39

3.2.9 - VALIDAÇÃO DO MODELO 40

3.3 - CONCLUSÃO 41

CAPÍTULO 4 AVALIAÇÃO DO ALGORITMO PROPOSTO 43

4.1 - CONSIDERAÇÕES INICIAIS 43

4.2 - MODELAGEM DE UM SISTEMA LINEAR POLINOMIAL 43

4.3 - MODELAGEM DE UM SISTEMA NÃO LINEAR POLINOMIAL 46

4.4 - MODELAGEM DE UM SISTEMA NÃO LINEAR RACIONAL SISO 47

4.5 - MODELAGEM DE UM SISTEMA NÃO LINEAR RACIONAL MISO _48

4.6 - MODELAGEM DE UM SISTEMA NÃO LINEAR RACIONAL MIMO _52

4.7 - COMENTÁRIOS FINAIS 58

CAPÍTULO 5 RESULTADOS DE SIMULAÇÕES 59

5.1 - CONSIDERAÇÕES INICIAIS 59

5.2 - CONVERSOR BOOST QUADRÁTICO-BUCK 59

5.3 - REATOR BIOQUIMICO 64

5.4 - CONSIDERAÇÕES FINAIS 68

CAPÍTULO 6 CONCLUSÃO GERAL 69

REFERÊNCIAS BIBLIOGRÁFICAS 71

APÊNDICE A DESCRITIVO DO SOFTWARE 79

A.1 - CARREGANDO UM PACOTE DE DADOS: 79

A.2 - INICANDO OS CÁLCULOS 82

A.3 - SELEÇÃO DE CURVA 83

A.4 - PARAR OS CÁLCULOS 85

A.5 - SELEÇÃO DE SAÍDA VISÍVEL 86

A.6 - FUNÇÃO ZOOM 87

A.7 - ESTRUTURA NARMAX POLINOMIAL OU RACIONAL 88

A.8 - TIPO DO EXPOENTE 89

A.9 - TAXA DO ERRO QUADRÁTICO MÉDIO (RJN) 90

A.10 - NÚMERO DE CICLOS SEM ALTERAÇÃO CONSIDERAVEL (NCY) 90

A.11 - SOMATÓRIO DE ERR DESEJADO (SERR) 91

A.12 - CONTINUANDO AS OPERAÇÕES 92

A.13 - FINALIZANDO OS CÁLCULOS 92

A.14 - TAMANHO DA POPULAÇÃO (N) 93

A.15 - ELITISMO (E) 94

A.16 - INICIANDO NOVAMENTE OS CÁLCULOS 94

A.17 - SAINDO DO SOFTWARE 95

A.18 - MENU DE CONFIGURAÇÃO 96

APÊNDICE B TÉCNICAS DE PROCESSAMENTO PARALELO UTILIZADAS 100

B.1 - ESCOLHA DA LINGUAGEM DE PROGRAMAÇÃO 101

B.2 - A CLASSE MULTI-TRHEAD DESTRUCT DO QT 102

B.3 - O MÉTODO RUN. 104

B.4 - O MÉTODO DES_ALGDIFFEVOL. 106

B.5 - RESULTADOS OBITIDOS 108

B.6 - CONCLUSÃO 111

ANEXO A ALGORITMO DA TAXA DE REDUÇÃO DO ERRO (ERR) 113

SELEÇÃO DA ESTRUTURA 113

ALGORITMO ORTOGONAL COM REGRESSÃO-DIRETA 114

ANEXO B MODELO RACIONAL LINEARIZADO NOS PARÂMETROS 117

DESCRIÇÃO ENTRADA-SAÍDA 117

EXPRESSÃO LINEAR NOS PARÂMETROS 118

ANEXO C ESTIMADOR DE MÍNIMOS QUADRADOS PARA O MODELO RACIONAL 120

ESTIMAÇÃO DE PARÂMETROS 120

ESTIMADOR USANDO MÉTODO DOS MÍNIMOS QUADRADOS 120

POLARIZAÇÃO DE $\hat{\Theta}$ 121

ESTIMADOR DE MÍNIMOS QUADRADOS PARA O MODELO RACIONAL – RME 124

ANEXO D ESTIMADOR ESTENDIDO DOS MÍNIMOS QUADRADOS 126

LISTA DE FIGURAS

Figura 2.1 – Diagrama do Protocolo de Identificação (Santos, 2003). 11

Figura 2.2 – Fluxograma de uma estrutura básica para algoritmos genéticos (Coelho & Coelho, 1999) 17

Figura 3.1 – Fluxograma da metodologia proposta de identificação de sistemas. 30

Figura 3.2 – Representação cromossômica por matriz tri-dimensional 32

Figura 3.3 – Estrutura das populações. 33

Figura 3.4 – Fluxograma do Algoritmo Diferencial Evolutivo Implementado. 35

Figura 3.5 – Diagrama de comparação dos modelos. 36

Figura 4.1 – Gráfico Medido X Estimado para o modelo da equação (4.1). 45

Figura 4.2 – Gráfico Medido X Estimado para o modelo da equação (4.2). 45

Figura 4.3 – Gráfico Medido X Estimado para o modelo da equação (4.3). 47

Figura 4.4 – Gráfico Medido X Estimado para o modelo da equação (4.4). 48

Figura 4.5 – Gráfico Medido X Estimado para o modelo com 500 pontos. 49

Figura 4.6 – Gráfico do melhor resultado Medido X Estimado com 1000 pontos. 50

Figura 4.7 – Histograma do melhor resultado com 1000 pontos. 50

Figura 4.8 – Testes estocásticos. 52

Figura 4.9 – Resultado para a saída 1. 53

Figura 4.10 – Resultado para a saída 2. 53

Figura 4.11 – Histograma dos resíduos das saídas 1 e 2 54

Figura 4.12 – Testes estocásticos do modelo obtido 57

Figura 5.1 – Esquema do conversor Boost2-Buck 60

Figura 5.2 – Histograma dos resíduos do modelo do conversor Boost2-Buck 62

Figura 5.3 – Formas de onda da corrente de saída do medida (azul) e estimada (vermelha). 62
Figura 5.4 – Formas de onda da corrente de saída do medida (azul) e estimada (vermelha). 64
Figura 5.5 – Formas de onda do valor de saída medida (azul) e estimada (vermelha) do reator bioquímico. 65
Figura 5.6 – Histograma dos resíduos obtidos na identificação do reator bioquímico. 66
Figura 5.7 – Curvas de correlação para validação estatística do modelo obtido do reator bioquímico 67
Figura A.1 – Função "Carregar" 79
Figura A.2 – Adicionando um dado 80
Figura A.3 – Determinando entradas e saídas 81
Figura A.4 – Propriedades da variável de entrada 81
Figura A.5 – Propriedades da variável de saída 82
Figura A.6 – Inicialização da identificação do sistema 82
Figura A.7 – Apresentação do gráfico e equação calculados 83
Figura A.8 – Gráfico de distribuição de resíduos 84
Figura A.9 – Curva esperada 84
Figura A.10 – Curva medida 85
Figura A.11 – Parando os cálculos 86
Figura A.12 – Seleção de saída visível 86
Figura A.13 – Aplicando Zoom 87
Figura A.14 – Minimizando o Zoom 88
Figura A.15 – Determinando a forma do resultado 89

Figura A.16 – Modificando o tipo do expoente 89

Figura A.17 – Modificando o valor do RJn 90

Figura A.18 – Modificando o valor de NCy 91

Figura A.19 – Determinando um novo valor para SERR 91

Figura A.20 – Continuando os cálculos 92

Figura A.21 – Finalizando os cálculos e tarefas 93

Figura A.22 – Modificando o N 93

Figura A.23 – trocando o valor de E 94

Figura A.24 – Reiniciando o sistema 95

Figura A.25 – Saindo do Software 95

Figura A.26 – Menu de configurações 96

Figura A.27 – Salvar dados carregados 97

Figura A.28 – Cromossomos salvos 98

Figura A.29 – Concatenando arquivos 99

Figura B.1 – Pseudocódigo dos atributos da classe DEStruct. 104

Figura B.2 – Pseudocódigo do método run ( ). 106

Figura B.3 – Diagrama esquemático dos pipelines criados para o Gerenciamento de Tarefas para a Execução de Tarefas em Modo Contínuo. 108

Figura B.4 –Software rodando em plataforma de 2 núcleos reais. 109

Figura B.5 – Software rodando em plataforma de 2 núcleos reais e 2 simulados. 110

Figura B.6 – Software rodando em plataforma de 4 núcleos reais. 110

Figura B.7 – Software rodando em plataforma de 8 núcleos reais. 111

LISTA DE TABELAS

Tabela 3.1: Representação dos regressores no aplicativo. _____32

Tabela 5.1: Parâmetros de simulação do conversor Boost2-Buck. _____60

LISTA DE ABREVIATURAS E SÍMBOLOS

CONJUNTOS NUMÉRICOS

Z	Conjunto dos números inteiros
N	Conjunto dos números naturais
R	Conjunto dos números reais

SIGLAS

AG	Algoritmo Genético
AIC	Akaike's Information Criterion
BIC	Critério de Informação Bayesiana.
DE	Diferencial Evolutivo
Duty	Razão Cíclica de operação do conversor
E	Elitismo
ERR	Error Reduction Ratio
Ex	Experimento.
FPE	Final Prediction Criterion
ICEC	International Contest on Evolutionary Computation
LED	Diodo Emissor de Luz
LMT	Levenberg-Marquardt
MA	Maior Aptidão (melhor individuo da população).
Md	Avaliação média da população.
MIMO	Múltiplas entradas e múltiplas saídas
MISO	Múltiplas entradas e única saída
MQ	Algoritmo dos mínimos quadrados
NARMAX	Nonlinear Autoregressive Moving Average Model with Exogenous Variables.
NARX	Nonlinear Autoregressive Model with Exogenous Variables.
NCARMA	Nonlinear Controlled Auto-Regressive Moving Average
NL	Não Linearidade
OED	Operador de Evolução Dirigido
PID	Proporcional Integral Derivativo
PLC	Controlador Logico Programável
PRBS	Pseudo-Random Binary Signal
RMSE	root mean square error
RSM	Método da Superfície de Resposta
SDK	Software Development kit
SSE	Sum of Squared Error
USB	Universal Serial Bus

VARIÁVEIS

Jn(.)	Índice de desempenho ou função custo

I_{LED}	Corrente nos led's
$Npop$	Tamanho da População
$U_{i,g}$	I-ésima vetor de avaliação da g-ésima população
$V_{i,g}$	I-ésima vetor mutante da g-ésima população
V_{in}	Tensão de entrada do conversor
$X_{best,g}$	Cromossomo com melhor aptidão da g-ésima população
$X_{i,g}$	I-ésima cromossomo da g-ésima população
$X_{new,g}$	Cromossomo gerado aleatoriamente na g-ésima população

CAPÍTULO 1

INTRODUÇÃO

1.1 - INTRODUÇÃO GERAL

Atualmente, com o desenvolvimento da ciência nas mais diversas áreas, os modelos matemáticos capazes de reproduzir, de forma aproximada, o comportamento dinâmico de sistemas reais são de elevada importância, pois segundo (IWASE, IIKUBO, HATAKEYAMA, & FURUTA, 2002), as técnicas de modelagem de sistemas fornecem o modelo matemático que descreve a dinâmica de um determinado sistema que pode ser utilizado no desenvolvimento de sistemas de controle.

Os modelos matemáticos, segundo (SANTOS, 2000), são obtidos através de dois modos: o primeiro modo, o analítico, conhecido também como modelagem teórica, é constituído por um modelo matemático formado por uma ou mais equações diferenciais e (ou) equações algébricas. O segundo modo, o experimental baseia-se em um modelo matemático obtido através de experimentos com informações de entrada e saída de um determinado processo dinâmico real, e tal modelo é representado por meio de equações diferenças.

Um modelo analítico é composto por descrições físicas do sistema, isto é, o modelo é descrito com base nas leis teóricas (leis da mecânica, leis da física, leis da termodinâmica, etc) e leis empíricas necessárias para descrever o comportamento dinâmico do sistema, de modo que tal abordagem permite derivar modelos que descrevem a dinâmica interna do sistema, além da relação entrada-saída (RODRIGUES, 2000). Esse tipo de modelagem também é conhecido como modelagem caixa-branca (BOSSLEY, 1997), pois descreve o comportamento dinâmico de um fenômeno ou de um sistema por aproximações analíticas, através de

equações constitutivas. Sua principal vantagem está no fato do modelo possuir uma interpretação física clara.

Em contra partida, como ressalta (SANTOS, 2000), este tipo de modelo possui algumas desvantagens em relação à complexidade de sua estrutura, ou seja, esse modelo pode envolver dezenas ou até centenas de equações constitutivas e, geralmente, esses modelos são contínuos no tempo, ao passo que as variáveis de saída dos sistemas e ações de controle são discretas no tempo. Para (PAIVA, 1999), esse tipo de modelagem, por ser a mais tradicional, apresenta diversas dificuldades em relação ao tratamento de sistemas mais complexos, não-lineares, estocásticos ou variantes no tempo. Assim, a construção de modelos analíticos para sistemas dessa natureza é geralmente uma tarefa árdua, devido ao seu custo tanto para desenvolvimento quanto computacional.

De acordo com (RODRIGUES, 2000), a modelagem pelas leis físicas de um sistema pode ser um processo complicado; quando o sistema torna-se grande, complexo ou quando aumenta a complexidade nas interações das variáveis do sistema, dificultando assim a sua descrição matemática. Consequentemente, conforme a complexidade do sistema aumenta, a precisão de um modelo utilizado para descrever o sistema com rigor fica restrita, diminuindo sua precisão no decorrer do processo.

Paralelamente a essas dificuldades citadas, segundo (PAIVA, 1999), a indústria tem demonstrado interesse em adquirir autonomia nos processos de produção. Porém, existem algumas questões relativas à complexidade crescente dos processos de produção que contradizem tal autonomia, como apresentado anteriormente.

No modo experimental de modelagem, também conhecido como modelagem experimental ou como Identificação de Sistemas, é necessária a realização de

experimentos e um modelo matemático é obtido através das informações de entrada e saída coletadas, em contraste com a modelagem analítica, onde esta é feita baseada em relações matemáticas e o conhecimento teórico do sistema ou do fenômeno a ser identificado (LJUNG, System Identification- Theory for the User, 1999).

Através da técnica de Identificação de Sistemas, não se faz necessário o conhecimento físico e ou teórico do sistema a ser identificado, pois o modelo matemático é obtido através de dados experimentais de entrada e saída do sistema a ser identificado, por isso o modelo pode ser classificado como um modelo do tipo caixa-preta. Em modelos do tipo caixa-preta os estados interiores são ignorados (SJOBERG, HJALMARSSON, & LJUNG, 1994), (LJUNG, System Identification- Theory for the User, 1999), (SIMANI, FANTUZZI, & BEGHELLI, 2000).

Para (SANTOS, 2000), uma das grandes vantagens desta técnica é a facilidade em se ajustar o modelo matemático, facilitando a formulação e a resolução de problemas de controle de processos. Porém, os parâmetros processados na Identificação de Sistemas não possuem significados físicos, caracterizando-se assim uma desvantagem. Outra desvantagem da modelagem experimental é a do grande esforço computacional que alguns sistemas exigem para selecionar a estrutura e estimar os parâmetros.

Existe outra técnica para a obtenção de modelos denominada de caixa-cinza. Esta técnica utiliza informação a *priori* do sistema em conjunto com os dados de saída e de entrada medidos no mesmo. Para (Jorgensen, 1995) a identificação caixa-cinza é determinada como: "a ciência de construção de modelos que incorporam conhecimento a *priori* do sistema, com um certo grau de incerteza na seleção da estrutura da representação".

Não importa se a modelagem é caixa-branca, caixa-cinza ou caixa-preta, se os modelos são lineares, eles são simples e de fácil obtenção. Por isso, o modelo linear é importante no desenvolvimento das técnicas de identificação, contudo, ele somente é aplicado em regiões restritas de operação. (Billings S. A., 1980) justifica a utilização de modelos lineares em função da complexidade potencial dos não-lineares. No entanto, a necessidade de representar de forma mais precisa o comportamento não-linear de sistemas reais levou à busca de representações não-lineares.

Foi a partir da década de oitenta que apareceram novos métodos para identificar sistemas não-lineares. Dentre os vários métodos propostos, destaca-se a representação NARMAX polinomial proposta em (Leontaritis, 1985a) e NARMAX racional por (Billings S. A., 1989a).

Somente na década de noventa, aplicações e técnicas de identificação de sistemas não-lineares emergiram em grande número. Entre diversos trabalhos, destacam-se: modelagem do controle neural de um sistema cardiovascular usando modelos NARMAX (Vallverdu, 1992); identificação de um sistema de controle de emissão de CO_2 na respiração de um homem (Noshiro, 1993); modelagem de um sistema de neutralização de pH com objetivo de desenvolvimento de controle preditivo (Pröll, 1994); identificação de não-linearidades em alto-falantes usando modelos NARMAX polinomiais (Jang, 1994); recuperação de mapas não-lineares estáticos a partir de dados caóticos usando modelos dinâmicos (Aguirre L. A., 1997).

A maior parte dos trabalhos que usam modelos NARMAX, utiliza a representação polinomial, porque ela possui várias vantagens como a facilidade de simulação, a linearidade nos parâmetros, a possibilidade de obter modelos estáticos e a facilidade de se combinar a seleção da estrutura com a estimação dos parâmetros.

No entanto, (Billings S. A., 1991) destacam que a representação NARMAX racional descreve eficientemente várias classes de não-linearidades. Além disso, os modelos racionais possuem características mais gerais e são capazes de representar com exatidão certos tipos de singularidades. Eles têm boa propriedade de extrapolação, são fáceis para avaliar e incluem funções polinomiais como caso especial (Zhu, 1993). Todavia, uma desvantagem do modelo racional é a sua não-linearidade nos parâmetros, o que torna a estimação deles mais complexa.

1.1 - RELEVÂNCIA

Segundo (PAIVA, 1999), a indústria tem demonstrado interesse em adquirir autonomia nos processos de produção. No entanto, existem algumas questões relativas à complexidade crescente dos processos de produção que contradizem tal autonomia.

Os modelos matemáticos analíticos são obtidos com base nas leis da mecânica, da física, da química ou da termodinâmica, isto é, com base naquilo que se designa por primeiros princípios. Este tipo de modelagem apresenta dificuldades a nível de sistemas de análise fenomenológica complexa ou com fatores de incerteza associados.

Realmente, a modelização rigorosa e precisa de sistemas fundamentados nos primeiros princípios diminui com o aumento da complexidade. Este problema é sintetizado por (Zadeh L. A., 1973) como o "princípio da incompatibilidade":

> *"À medida que a complexidade de um sistema aumenta, a nossa capacidade de descrever o seu comportamento de forma precisa e, além disso, significativa vai diminuindo até que seja atingido um limiar para além do qual a precisão e a relevância se tornam características quase mutuamente exclusivas."*

Neste sentido, a modelagem caixa-preta se torna atrativa, pois não se faz necessário o conhecimento físico e ou teórico do sistema a ser identificado. Por isso, o desenvolvimento de ferramentas para a obtenção de modelos tipo caixa-preta é de grande interesse no campo da pesquisa e desenvolvimento em diversas áreas do conhecimento.

Considerando que o sistema pode ser linear ou não, a ferramenta ideal deveria ser capaz de identificar o tipo de modelo e gerar o modelo apropriado para cada caso. Considerando ainda que os sistemas estão se tornando cada vez mais complexos, a representação matemática a ser adotada deve ser capaz de representar com exatidão certos tipos de singularidades e ser a mais geral possível.

Os trabalhos publicados até o momento apontam que a representação que mais atende a esta tendência é a NARMAX racional (Billings S. A., 1991), (Zhu, 1993), (Braess, 1986), dentre outros autores. Lembrando que, uma desvantagem desta representação é a não-linearidade nos parâmetros, o que torna a estimação destes mais complexa.

1.2 - MOTIVAÇÃO

A representação NARMAX polinomial é largamente usada na identificação de sistemas dinâmicos não-lineares. Por outro lado, a representação NARMAX racional, apesar de ser um caso mais geral que a polinomial, tem sido pouco empregada na modelagem de dinâmica de sistemas reais. Isto ocorre porque o modelo racional não é linear nos parâmetros, o que torna a estimação dos parâmetros dele mais complexa.

Por isso, alguns sistemas exigem grande esforço computacional para selecionar a estrutura e estimar os parâmetros.

Várias soluções foram propostas na literatura para minimizar o esforço computacional que a modelagem experimental necessita em determinadas aplicações, dentre as quais podemos citar:

1. Utilizar uma estrutura AR (Auto-Regressive) para modelar um servidor Web Apache, com um numero de MaxClients reduzido (Thiago W. M. Abreu, 2010). A solução empregada apresentou uma resposta satisfatória para o problema estudado, no entanto, por ser uma estrutura linear nos parâmetros, ela pôde ser resolvida com o método dos mínimos quadrados, ou seja, o sistema pôde ser avaliado por um método direto para obtenção dos parâmetros. Todavia, em muitos casos, a descrição do sistema por modelos com esta caracteristica não são adequados.

   Nesses casos, é necessário introduzir termos não-lineares que podem levar o modelo a ser não linear nos parâmetros. E, consequentemente, a não linearidade nos parâmetros leva o problema à uma solução analítica e exige o emprego de métodos numéricos e testes cíclicos. Quando isso ocorre, a identificação da equação-parâmetro não é um problema trivial, principalmente por causa do número elevado de diferenciação dos sinais gravados.

2. Algoritmo DE (Diferencial Evolutivo) na estimação dos parâmetros num processo de identificação das propriedades plasticas dos materiais, utilizando o modelo de Friderick-Armstrong (ŁUKASZ MACIEJEWSKI, 2007). Neste caso, a indentificação da estrutura não se fez necessária. Contudo, existem casos mais complexos onde o sistema não possui uma estrutura conhecida,

exigindo que a seleção da estrutura necessite de algoritmos análiticos (recursivos).

3. O pesquisador (ZORLU, 2011) apresentou um trabalho que comparou o algoritmo DE com o AG e concluiu que o algoritmo DE apresentou uma performace maior do que o AG, quando aplicado ao problema de estimação de parâmetros nos modelos que ele estudou.
4. Novo algoritmo estimador de parâmetros, denominado COUPLED ESTIMATION ALGORITHM (C-LS algorithms), criado como objetivo de reduzir o esforço computacional na estimação de parâmetros (Yanjun Liu, 2011). Este algoritmo não otimiza a estrutura.

Neste sentido, este trabalho apresenta uma nova representação matemática do tipo NARMAX que pode ser tanto racional quanto polinomial, mas com expoentes reais. E para identificar modelos com esta nova representação também é proposto uma nova metodologia que envolve algoritmos estocásticos e determinísticos. Esta metodologia de identificação de sistemas utiliza o algoritmo Diferencial Evolutivo, com um novo operador mutação, capaz de gerar um domínio infinito de regressores para serem testados assim como refinar os expoentes destes regressores, que são números reais. Ademais, na metodologia proposta não necessita informar se a estrutura vai ser racional ou polinomial, se os expoentes serão reais ou inteiros, pois todas estas combinações são testadas e ele é capaz de convergir para a mais simples e mais representativa conforme função de aptidão que é utilizada.

1.3 - APRESENTAÇÃO DO TRABALHO

As principais contribuições do trabalho proposto são:

1. Apresentação de um novo tipo de representação matemática para identificação de sistemas utilizando de expoentes reais;
2. Desenvolvimento de uma metodologia (algoritmo) para identificar sistemas dinâmicos não-lineares que abrange representações do tipo NARMAX racional e com expoentes reais;

A seguir serão apresentados os capítulos que compõem este trabalho:

O Capítulo 2 contém uma revisão bibliográfica do contexto em que este trabalho está inserido. Ao longo deste capítulo são mostradas várias estratégias de identificação de sistemas. Dentre elas, o algoritmo DE.

O Capítulo 3 apresenta as técnicas utilizadas na metodologia proposta de identificação de sistemas e como ela é aplicada na representação NARMAX racional.

O Capítulo 4 utiliza de modelos artificiais, sendo alguns retirados da literatura, com características singulares que os algoritmos comumente empregados apresentam dificuldade e que a metodologia proposta resolve.

O Capítulo 5 apresenta dois estudos de caso, a identificação de um conversor CC-CC denominado BOOST QUADRÁTICO – BUCK e a identificação de um Reator de Polímero utilizando a metodologia e a representação proposta neste trabalho.

O Capítulo 6 contém a conclusão geral, com as análises dos principais resultados obtidos. (Billings S. A., 1980)

CAPÍTULO 2

REVISÃO BIBLIOGRÁFICA

Na literatura atual, encontram-se vários métodos, com características e técnicas diversas, para a obtenção de modelos paramétricos empregados em sistemas dinâmicos. Com o objetivo de otimizar o processo de identificação de um determinado sistema, isto é, de reduzir o esforço computacional, tanto no quesito de acelerar a convergência para o modelo desejado, quanto para diminuir a quantidade de regressores, fez-se um estudo de identificação de sistemas, algoritmos evolutivos e técnicas de otimização.

Neste sentido, este capítulo apresenta o resultado deste estudo com o propósito de auxiliar a compreensão das técnicas que foram selecionadas e das que foram definidas no trabalho proposto.

2.1 - TÉCNICAS DE IDENTIFICAÇÃO DE SISTEMAS

Identificação de sistemas consiste na representação do comportamento de um processo por meio de um modelo matemático. Alguns autores chamam de identificação caixa preta aquela que independe do conhecimento prévio a respeito do sistema a ser modelado, e caixa cinza quando os modelos obtidos provêm com o auxílio de um pré-conhecimento do sistema a ser modelado (Aguirre L. A., 2007).

O procedimento de identificação de sistemas pode ser dividido nas seguintes etapas: tratamento das medidas, seleção do modelo, determinação da estrutura, estimação dos parâmetros e validação do modelo, conforme fluxograma na Figura 3.2 (Santos, 2003).

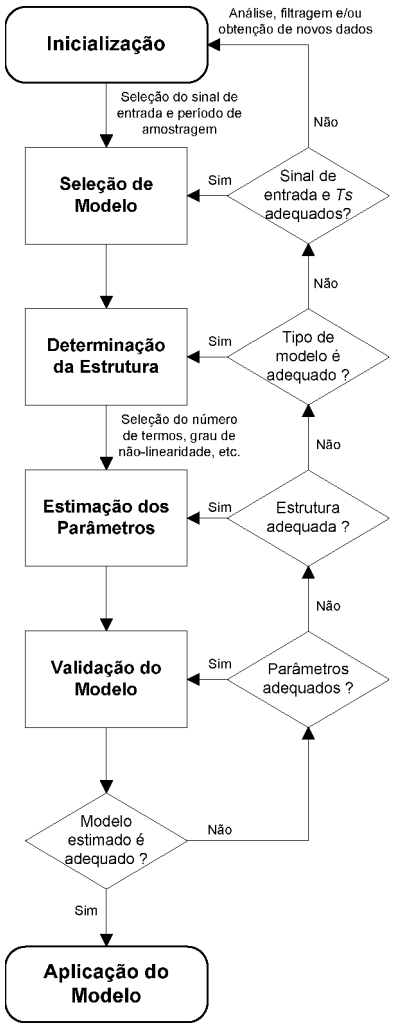


Figura 2.1 – Diagrama do Protocolo de Identificação (Santos, 2003).

Cada uma destas etapas, contidas no fluxograma, serão explicadas na sequência do texto.

2.2 - SELEÇÃO DO MODELO

Existem diversos modelos matemáticos para a representação de um processo dinâmico e cada um possui características distintas que devem ser levadas em conta na seleção. Para escolher o modelo mais adequado para uma aplicação deve-se considerar sua capacidade de representar as características da planta. Contudo, a

simplicidade do modelo está diretamente relacionada ao esforço computacional envolvido, sendo este um fator fundamental para uma implementação em tempo real. Na prática, o modelo escolhido é em geral o mais simples possível, capaz de atender aos requisitos operacionais estabelecendo um compromisso entre capacidade de aproximação versus simplicidade de representação (Santos, 2003).

2.2.1 - MODELOS NÃO-LINEARES

Um sistema não-linear é aquele que não atende ao princípio da sobreposição de efeitos e com isso, sistemas não-lineares criam novas frequências em regime permanente, ou seja, o sinal de saída pode apresentar frequências que não estão presentes no sinal de entrada. Do ponto de vista matemático um sistema não-linear é aquele que não pode ser modelado por meio de equações lineares: algébricas, diferenciais ou a diferenças. Portanto para estes casos soluções numéricas produzidas com o auxílio do computador, são sempre possíveis de serem obtidas, apesar dos limites de precisão (quantização numérica) (PEARSON, 2003).

Conforme o conhecimento prévio das características do processo é possível determinar, a partir de um teste de não linearidade, a possibilidade de representação linear ou a necessidade de emprego de um modelo não-linear. No caso de um modelo linear, informações em relação à dinâmica, complexidade, estabilidade em malha aberta e conhecimento prévio da estrutura são utilizadas para a definição de um modelo paramétrico ou não-paramétrico. Quando o modelo escolhido é não-linear, o conhecimento do tipo de não linearidade (NL) possibilita a seleção de modelos de *Wiener* ou *Hammerstein* (NL estática) ou Bilinear e Volterra (NL dinâmica). É possível ainda decidir a respeito da forma de representação da não linearidade estática, conforme o conhecimento da sua estrutura, representação via expansão

polinomial ou racional, ou mesmo através de modelos semi-paramétricos (neural e/ou nebuloso) (SANTOS, 2003).

2.2.1.1 - MODELO NCARMA (NONLINEAR CONTROLLED AUTO-REGRESSIVE MOVING AVERAGE)

Para uma melhor compreensão é importante que se faça claro as seguintes denominações:

- Coeficientes são as constantes C_i e d_i da Equação (2.2);
- Termo é a variável com suas variações de linearidade e ordem. Exemplo: $y(k-j)^{n_y}$;
- Expoentes são as constantes n_y, n_u, n_e, d_y, d_u e d_e que são os graus de não linearidade de cada termo da Equação (2.2);
- Regressores é o conjunto de termos multiplicados por um coeficiente. Exemplo: $C_i * y(k-j)^{n_y} * u(k-r)^{n_u} * e(k-q)^{n_e}$;

Os modelos NARX (*Nonlinear Autoregressive Model with Exogenous Variables*) são modelos discretos no tempo que explicam o valor da saída *y(k)* em função de valores prévios dos sinais de saída e de entrada, ou seja, $y(k) = F[y(k-1), \dots, y(k-n_y), u(k-T_d), \dots, u(k-n_u)]$, sendo n_y, n_u, T_d são os maiores atrasos em *y*, em *u* e o tempo morto respectivamente. Mas a fim de evitar polarização de parâmetros, ou seja, o erro da equação correlacionado com alguns regressores é comum incluir termos de ruído no modelo. Quando isso é feito o modelo passa a ser um modelo NCARMA (Nonlinear Controlled Auto-Regressive Moving Average), que na literatura aparece comumente como NARMAX *(Nonlinear Auto-Regressive Moving Average Model with Exogenous Variables)*. Este representa o sistema através de uma função polinomial ou racional com grau de não linearidade *l*

cuja parcela determinística é apresentada como o somatório de termos com graus de não linearidade m ($1 \leq m \leq l$), conforme (2.1) (Aguirre L. A., 2007):

$$y(k) = \sum_{i} C_i \prod_{j=1}^{n_y} y(k-j) \prod_{r=1}^{n_u} y(k-r) \prod_{q=0}^{n_e} e(k-q) \tag{2.1}$$

Sendo e (k) o ruído e n_e é o maior atraso no modelo de ruído.

Modelos racionais são formados pela razão entre dois polinômios (Equação 3.5), ou seja:

$$y(k) = \frac{\sum_i c_i \prod_{j=1}^{n_y} y(k-j) \prod_{r=1}^{n_u} u(k-r) \prod_{q=1}^{n_e} e(k-q)}{\sum_i d_i \prod_{j=1}^{d_y} y(k-j) \prod_{r=1}^{d_u} u(k-r) \prod_{q=1}^{d_e} e(k-q)} + e(k) \tag{2.2}$$

Apesar de permitir utilizar regressores não lineares nos sinais de saída, entrada e ruído, os modelos polinomiais do tipo mostrado na Equação (2.2) são lineares nos parâmetros, ou seja, podem ser representados $y(k) = \psi^T \theta$ e consequentemente seus parâmetros podem ser resolvidos por um método direto, como o método dos mínimos quadrados. Por outro lado, os modelos racionais, por terem uma estrutura mais flexível, podem ser mais eficientes na modelagem de certos sistemas quando comparados com modelos polinomiais. Entretanto, eles não são lineares nos parâmetros e algoritmos alternativos são necessários para estimá-los, além dos modelos racionais serem mais sensíveis ao ruído (Aguirre L. A., 2007).

Uma das técnicas para calcular os coeficientes dos modelos NCARMA racional é linearizá-lo nos parâmetros, conforme Anexo B. Esta foi a técnica utilizada na metodologia proposta.

Além disso, vale salientar que os modelos NCARMA racional não possuem representação com expoentes reais, ou seja, os expoentes informam na verdade quantas vezes um termo é repetido em um regressor. As técnicas aplicadas até então utilizam de um domino pré-estabelecido de regressores que serão testados no processo

de identificação, o que dificulta muito gerar um domínio onde estes regressores possuem expoentes com valores quaisquer, ou melhor, valores reais.

Neste trabalho utiliza-se de um algoritmo estocásticos (DE) que gera este domínio a todo instante usando técnicas de diferenciação capaz de refinar estes expoentes e assim obter um valor mais preciso que às vezes não é inteiro. Este algoritmo será apresentado no próximo capitulo.

2.3 - SELEÇÃO DE ESTRUTURA

Existem diversos critérios para a seleção de ordem de modelos lineares monovariáveis, como aqueles baseados na razão entre determinantes, critério de informação de Akaike (AIC – *Akaike's Information Criterion*) e critério do erro de predição final (FPE – *Final Prediction Criterion*). No entanto, quando o sistema é não linear, são poucas as ferramentas para auxiliar nesta etapa. (Aguirre L. A., 2007) propõe a aplicação da taxa de redução do erro, ERR (*Error Reduction Ratio*), conforme Anexo A, aplicadas a modelos NCARMA. Esta estratégia permite a detecção de quais parcelas do modelo são mais relevantes para serem incluídas e quais podem ser consideradas desprezíveis (Equação 3.6 e 3.7):

$$SSE = \sum_{t=1}^{N} [y(t) - \hat{y}(t)]^2 \tag{2.3}$$

$$[ERR]_i = \frac{g_i^2 w_i^2 w_i}{SSE}, \qquad 1 \leq i \leq n_\theta \tag{2.4}$$

Onde w_i indica o termo incluído e g_i o seu respectivo parâmetro.

A quantidade de combinação para a obtenção dos regressores que comporão uma estrutura na maioria dos casos é muito grande, podendo levar até mesmo anos de testes para percorrer todas as possibilidades, o que é inviável. Por outro lado, a

computação evolutiva é uma aliada poderosa para solucionar este tipo de problema (Coelho & Coelho, 1999).

2.3.1 - COMPUTAÇÃO EVOLUTIVA

Um algoritmo genético (AG) é uma técnica de busca utilizada na ciência da computação para encontrar soluções aproximadas em problemas de otimização e busca, sendo fundamentado principalmente pelo americano John Henry Holland. Algoritmos genéticos é uma classe particular de algoritmos evolutivos que usam técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e recombinação (ou *crossing over*) (Storn & Price, 1995).

Algoritmos genéticos são implementados como uma simulação de computador, em que uma população de representações abstratas de solução é selecionada em busca de soluções melhores. A evolução geralmente se inicia a partir de um conjunto de soluções criado aleatoriamente e é realizada por meio de gerações. A cada geração, a adaptação de cada solução na população é avaliada, alguns indivíduos são selecionados para a próxima geração, são recombinados ou sofrem mutação para formar uma nova população. A nova população então é utilizada como entrada para a próxima iteração do algoritmo (Dasrupta, Das, Biswas, & Abraham, 2009).

Algoritmos genéticos diferem de alguns algoritmos tradicionais em basicamente cinco aspectos (Linden, 2008):

1. Baseiam-se em uma codificação do conjunto das soluções possíveis e não nos parâmetros da otimização em si;
2. Os resultados são apresentados como uma população de soluções e não como uma solução única;

3. Não necessitam de nenhum conhecimento derivado do problema, apenas de uma forma de avaliação do resultado;
4. Usam transições probabilísticas e não regras determinísticas.
5. Os algoritmos genéticos são em geral algoritmos simples e fáceis de serem implementados. Na Figura 2.2 temos um fluxograma de um programa básico.

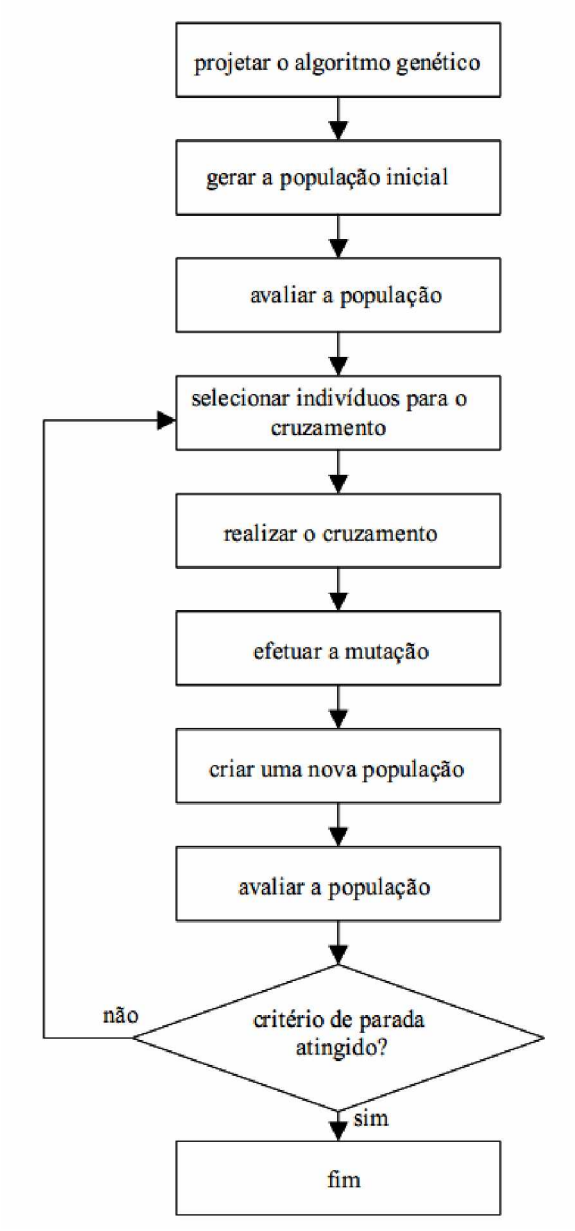


Figura 2.2 – Fluxograma de uma estrutura básica para algoritmos genéticos (Coelho & Coelho, 1999).

A grande vantagem dos algoritmos genéticos está no fato de não precisarmos saber como funciona o cálculo de aptidão, apenas temos que tê-lo disponível para ser aplicado aos indivíduos e comparar os resultados (Coelho & Coelho, 1999).

O indivíduo é meramente um portador do seu código genético, sendo este uma representação do espaço de busca do problema a ser resolvido, em geral na forma de sequências de *bits*. Por exemplo, para otimizações em problemas cujos valores de entrada são inteiros positivos de valor menor que 255, podemos usar 8 bits com a representação binária normal, ou ainda uma forma de código *gray*. Problemas com múltiplas entradas podem combinar as entradas em uma única sequência de *bits*, ou trabalhar com mais de um cromossomo, cada um representando uma das entradas. O código genético deve ser uma representação capaz de representar todo o conjunto dos valores no espaço de busca, além de ter tamanho finito (Coelho & Coelho, 1999).

A seleção também é outra parte chave do algoritmo. Em geral, usa-se o algoritmo de seleção por roleta, onde os indivíduos são ordenados de acordo com a aptidão e lhes são atribuídas probabilidades decrescentes de serem escolhidos. A escolha é feita então aleatoriamente, de acordo com essas probabilidades. Dessa forma consegue-se escolher como pais os mais bem adaptados, sem deixar de lado a diversidade dos menos adaptados. Outra forma de seleção é o Torneio, onde é selecionado aleatoriamente um grupo de "k" elementos, sendo que neste subgrupo de forma também aleatória é escolhido um elemento e repete esta operação até completar a população (Linden, 2008).

A reprodução tradicionalmente é dividida em três etapas, sendo acasalamento, recombinação e mutação. O acasalamento é a escolha de dois indivíduos para se reproduzirem, geralmente gerando dois descendentes para manter o tamanho populacional. A recombinação, ou *crossing-over*, é um processo que imita o processo

biológico homônimo na reprodução sexuada, em que os descendentes recebem em seu código genético parte do código genético do pai e parte do código da mãe. Esta recombinação garante que os melhores indivíduos sejam capazes de trocar entre si as informações que os levem a serem mais aptos a sobreviver, além de gerarem descendentes ainda mais aptos. Já as mutações são feitas com o objetivo de permitir maior variabilidade genética na população, impedindo que a busca fique estagnada em um mínimo local, mas quase que esporadicamente para não tornar o sistema instável (Russel & Norvig, 2004).

2.3.1.1 - ALGORITMO DIFERENCIAL EVOLUTIVO (DE)

Evolução Diferencial (*Differential Evolution* – DE) é um algoritmo de otimização simples e eficiente, que tem recebido cada vez mais destaque no âmbito da otimização não linear com variáveis contínuas. Ele foi proposto por (Storn & Price, 1995) e ganhou destaque internacional nas competições do *International Contest on Evolutionary Computation* – ICEC, do IEEE, em 1996.

Por seguir a linha dos algoritmos que evoluem uma população de soluções, o método de Evolução Diferencial é classificado como um algoritmo evolutivo, embora não tenha qualquer inspiração em um processo evolutivo natural. O seu operador de mutação, por exemplo, utiliza conceitos matemáticos e estratégias heurísticas (LACERDA, 2010).

Primeiramente, é preciso estabelecer limites superior e inferior para cada parâmetro utilizado. Feito isso, a população inicial é gerada aleatoriamente dentro desse espaço de busca estabelecido. Sendo os valores gerados de maneira uniformemente distribuída, a tendência é que o algoritmo realize uma varredura por todo o espaço de busca (LACERDA, 2010).

Após a inicialização, o algoritmo DE realiza mutação e cruzamento sobre os indivíduos. Para a mutação, são selecionados três indivíduos diferentes da população e, em seguida é realizada a operação conhecida como mutação diferencial vide Equação (2.5):

$$u_i = x_{r1} + \eta(x_{r3} - x_{r2}) \tag{2.5}$$

Nessa operação é realizada uma perturbação em um indivíduo x_{r1}, denominado vetor base. Tal perturbação consiste em um vetor diferencial entre dois outros indivíduos x_{r2} e x_{r3}. Esse vetor é então multiplicado por um peso $\eta \epsilon [0,1]$, que irá determinar o tamanho do passo. Para completar o processo de busca, o algoritmo DE programa o cruzamento, também conhecido como recombinação discreta. Tal processo consiste em criar um novo indivíduo, a partir de dados copiados de outros dois. No caso da Evolução Diferencial, cada vetor da população atual é cruzado com um vetor mutante criado com o procedimento anterior (Chakraborty, 2008).

Após a geração do novo indivíduo, o valor de sua função objetivo é calculado, sendo que caso esta seja igual ou melhor a seu correspondente na população atual x_i, o indivíduo é substituído. Caso contrário, o indivíduo atual é mantido. Uma vez que a nova população é criada, os processos de mutação, cruzamento e seleção são repetidos até que um critério de parada seja alcançado (Chakraborty, 2008).

A Equação (2.5) não é a única forma de estabelecer o processo de mutação existem outras nas literaturas, mas neste trabalho gostaria de destacar a RAND-TO-BEST apresentada na Equação (2.6) que consiste de selecionar um vetor base denominado $X_{r1,G}$ (que é um cromossomo da população em testes) e utilizando de outros dois vetores também selecionados de forma aleatória e o melhor individuo da população em teste ($X_{best,G}$) cria-se um novo vetor.

$$V_{i,G} = X_{r1,G} + \mu\,(X_{best,G} - X_{r1,G}) + F\,(X_{r_2,G} - X_{r_3,G}) \tag{2.6}$$

Onde:

$V_{i,G}$ é o cromossomo mutante;

$X_{r1,G}$ é o cromossomo base;

$X_{best,G}$ é o melhor cromossomo;

$X_{r_2,G}$ e $X_{r_3,G}$ são os cromossomos escolhidos aleatóriamente na população para criar o vetor diferença;

F e μ são os coeficientes de mutação que variam de [0, 1] (Price K.V., 2005).

2.4 - ESTIMAÇÃO DE PARÂMETROS

Selecionada a estrutura para obter os parâmetros de cada elemento da mesma, utiliza-se interpolação de uma tabela de valores para a obtenção dos mesmos. Se a tabela é resultado de algum experimento físico, os valores da tabela podem conter erros inerentes que usualmente não são previsíveis com nenhum grau de certeza. Isto equivale a dizer que os erros inerentes são distribuídos de acordo com algum padrão estatístico, sendo que há uma razoável probabilidade de que alguns deles sejam bastante grandes (Madsen, Nielsen, & Tingleff., 2004). Suponha que tenhamos uma fórmula relacionando y a x conforme Equação (2.7):

$$\bar{y} = f(x) \tag{2.7}$$

Onde a barra indica que este é um valor aproximado de y. A interpolação, portanto, produziria resultados que também teriam erros substanciais. Antes de poder usar tais dados, deve-se "suavizá-los" para anular os erros estatísticos tanto quanto possível. Isto pode acontecer, também, no caso de extrapolação, particularmente para dados econômicos. Por estas considerações, procura-se um método em que são usados

os dados de uma tabela de dados experimentais, para produzir uma fórmula que relacione y e x (Madsen, Nielsen, & Tingleff., 2004).

Deseja-se escolher $f(x)$ de modo que os desvios sejam pequenos, ou seja, procura-se fazer a soma dos desvios pequena (Equação 3.10), isto é:

$$\sum_{i=1}^{m} |d_i| = \sum_{i=1}^{m} |y_i - \bar{y}| \tag{2.8}$$

Visto serem estes termos que estão causando o problema, o passo mais lógico é minimizar a soma dos valores absolutos dos desvios.

Este método produz resultados úteis, porém é um difícil problema numérico, em parte porque a derivada da função valor absoluto não existe na origem. No entanto, existem alguns algoritmos para minimizar o problema (Madsen, Nielsen, & Tingleff., 2004). Nesta seção são apresentadas algumas estratégias de estimação de parâmetros para sistemas MISO não lineares, baseadas nas medidas de entrada e saída do sistema. Para a formalização das técnicas optou-se por um modelo representado por uma estrutura do tipo NARMAX Racional, conforme Equação (2.2), sendo possível a aplicação em outros tipos de sistemas.

No Anexo C é apresentado um algoritmo para a estimação dos parâmetros empregado em modelos racionais.

No Anexo D é apresentado um algoritmo para o cálculo de termos lineares do ruído correlacionado com o modelo NCARMA racional.

Os algoritmos apresentados no Anexo C e D foram utilizados na metodologia proposta de identificação de sistemas.

2.5 - ÍNDICES DE AVALIAÇÃO

São necessários índices para que se possa comparar um modelo com outro num processo de identificação de sistemas. Existem vários índices e cada um avalia um determinado aspecto do modelo, conforme especificado a seguir:

2.5.1 - SOMATÓRIO DO ERRO QUADRÁTICO – SSE (SUM OF SQUARED ERROR)

Este índice representa a soma dos quadrados do erro de estimação e, quanto menor for o seu valor, melhor é a qualidade do modelo, seu calculo é obtido pela equação (2.9).

$$SSE = \sum_{t=1}^{N} [y(t) - \hat{y}(t)]^2 \tag{2.9}$$

2.5.2 - PRINCÍPIO DA PARCIMÔNIA

Embora forneçam uma indicação importante em relação à qualidade do modelo obtido. O SSE não leva em consideração a complexidade (número de parâmetros) do modelo. A seleção do modelo mais adequado deve sempre levar em conta a relação entre capacidade de representação e simplicidade, estabelecendo um compromisso entre precisão do modelo x esforço computacional, por exemplo, por meio do critério de informação de Akaike (LJUNG, System Identification, 1996).

Critérios de informação podem ser avaliados para manter a ordem do modelo estimado para o processo tão simples quanto possível. Critérios como de informação Bayesiana e de Akaike combinam a variância residual do erro de estimação e a ordem do modelo (COELHO & COELHO, 2003).

Inicialmente, um modelo de baixa ordem é selecionado, aumenta-se a ordem do modelo estimado e o critério é avaliado para cada incremento na ordem do modelo. Assim, a estrutura adequada do modelo estimado é a que proporciona a menor taxa de variação do critério de informação. Diversos critérios de informação podem ser utilizados na complementação da função custo básica dos mínimos quadrados (2.10), isto é,

$$J_N = \frac{1}{N}\sum_{t=1}^{N}[y(t)-\hat{y}(t)]^2 \tag{2.10}$$

com alguns termos extras que penalizam a complexidade do modelo matemático da planta e podem ser selecionados de acordo com um dos seguintes testes:

2.5.3 - CRITÉRIO DE INFORMAÇÃO DE AKAIKE (AKAIKE INFORMATION CRITERION)

Ao selecionarmos modelos é preciso ter em mente que não existem modelos verdadeiros. Há apenas modelos aproximados da realidade que, causam perda de informações. Deste modo, é necessário fazer a seleção do "melhor" modelo, dentre aqueles que foram ajustados, para explicar o fenômeno sobre o estudo.

Akaike (1974) utilizou a Informação de Kullback-Leibler para testar se um dado modelo é adequado. Porém seu uso é limitado, pois depende da distribuição g (modelo verdadeiro), que é desconhecida. Aikaike mostrou que o viés é dado assintoticamente por p, em que p é o número de

parâmetros a serem estimados no modelo, e definiu seu critério de informação conforme equação (2.11).

$$AIC = N\ln[J_N] + 2p \tag{2.11}$$

Onde N é o número de medidas e p é o número de parâmetros do modelo estimado.

2.5.4 - CRITÉRIO DE INFORMAÇÃO BAYESIANA (BAYESIAN INFORMATION CRITERION)

O Critério de Informação Bayesiano(BIC) , proposto por Schwarz (1978) é dado pela equação (2.12) onde o modelo é escolhido considerando a quantidade de termos e o custo básica dos mínimos quadrados. Na equação (2.12) p é o número de parâmetros a serem estimados e n é o número de observações da amostra.

$$BIC = N\ln[J_N] + p\ln[N] \tag{2.12}$$

Em (Emiliano, Veiga, Vivanco, & Menezes, 2010), constatou-se que para modelos normais o uso do BIC mostrou-se superior para amostras grandes, sendo que para amostras pequenas teve um desempenho similar ao AIC e AICc. No caso de modelos de séries temporais, mesmo para amostras de tamanho pequeno (< 500), o desempenho dos critérios AIC e AIC corrigido foram similares e também inferiores ao BIC, para todos os modelos simulados. Assim, para os casos que foram simulados por (Emiliano, Veiga, Vivanco, & Menezes, 2010), o uso do BIC para séries foi o mais indicado. Por isso, optou-se por utilizar o BIC na metodologia proposta.

2.5.5 - ERRO DE PREDIÇÃO

O uso de predições k passos à frente é visto como uma forma mais eficiente de se comparar um modelo obtido visto que como a maioria dos algoritmos de

identificação minimiza o erro de predição K passos a frente. Uma forma de avaliar a qualidade do modelo é a análise do índice RMSE (*root mean square error*) k passos à frente (Equação (2.13)):

$$RMSE(i) = \frac{\sqrt{\sum_{t=1}^{N}\left[y(t) - \hat{y}(t)\right]^2}}{\sqrt{\sum_{t=1}^{N}\left[y(t) - y(t-i)\right]^2}} \tag{2.13}$$

Onde i representa o horizonte de predição empregado.

Na comparação entre modelos distintos, o menor RMSE denota um melhor desempenho do modelo (Aguirre L. A., 2007).

2.6 - CONSIDERAÇÕES FINAIS

Foram apresentadas as diversas linhas de pesquisa relacionadas à identificação de sistemas, que são responsáveis pelo direcionamento deste trabalho. Constatou-se que a aplicação dos algoritmos evolucionários na identificação é amplamente utilizada e como já foi citado anteriormente isto se deve principalmente, à facilidade de tais métodos em trabalhar com problemas complexos e à qualidade das soluções encontradas. Também observou nestas referencias que o algoritmo diferencial evolutivo (DE) mostra-se ainda melhor, mas que apresenta certa dificuldade em ser aplicado em problema de combinação, já que sua característica fundamental é para problemas de otimização.

É identificado também que a representação NARMAX racional teoricamente é capaz de representar uma maior classe de sistemas não-lineares do que a NARMAX polinomial, mas que o processo para sua obtenção é mais onerosa. Porém este

trabalho ressaltou que estas representações não se utilizam de expoentes reais, pois as técnicas para a obtenção dos mesmo tornaria ainda mais onerosa. Mas mesmo com esta ponderação optou-se, neste projeto, em obter uma representação NARMAX Racional com expoentes reais que será discutida no próximo capitulo, utilizando o algoritmo diferencial evolutivo (DE) na seleção dos elementos da estrutura matemática, mesmo sabendo que este é um problema de combinação e que isto pode ser uma dificuldade, porém compensada com a possível vantagem que o DE tem sobre o AG. Também com base nas referencias optou-se em utilizar a Taxa de Redução de Erro (ERR) para a seleção de regressores e Bayesiana Information Criterion (BIC) como avaliação da aptidão e optou-se por não fazer nenhum teste de não-linearidade visando equilibrar o compromisso entre a complexidade do modelo e do esforço computacional para encontra-lo versus a capacidade de representação do processo. Diversas técnicas de estimação dos parâmetros do modelo também foram discutidas, porém optou-se por apresentar as estratégias de validação dos modelos obtidos no capitulo subsequente.

CAPÍTULO 3

TÉCNICAS UTILIZADAS NA METODOLOGIA PROPOSTA DE IDENTIFICAÇÃO DE SISTEMAS

A metodologia proposta utiliza técnicas estocásticas e determinísticas. A técnica estocástica predominante é o algoritmo diferencial evolutivo (DE). Este capítulo apresenta a aplicação destas técnicas na metodologia proposta de identificação de sistemas.

3.1 - VISÃO GERAL DA METODOLOGIA PROPOSTA

A Figura 3.1 mostra o fluxograma das etapas do algoritmo DE empregado na metodologia proposta para a identificação de sistemas, sendo que ela pode ser aplicada em sistemas SISO ou MIMO.

O fluxograma consiste das seguintes etapas:

- **Etapa 1:** Os dados de entrada e de saída do sistema a ser identificado são alocados em uma matriz;
- **Etapa 2:** Esta etapa gera uma população de cromossomos, sendo esta população de tamanho informado pelo usuário;
- **Etapa 3:** Nesta etapa o operador mutação é aplicado, criando-se uma população mutante com diversas características alteradas em comparação a população que a originou;
- **Etapa 4:** Por meio do cruzamento dos Regressores da População Original com os Regressores da População Mutante, gera-se os Regressores de uma Nova População;

- **Etapa 5:** Agrupa-se os Regressores da População Original, Regressor da População Mutante e os Regressores do Cruzamento e elimina-se os piores regressores com o auxílio do algoritmo ERR e em seguida efetuar uma nova seleção aleatória e assim estabelecer o domínio final de regressores para cada cromossomo;
- **Etapa 6:** Nesta etapa, calcula-se os coeficientes, utilizando-se o método dos mínimos quadrados estendidos e a aptidão com o método BIC.
- **Etapa 7:** Calcula-se a aptidão de cada cromossomo da população mutante e compara com o seu respectivo da população original utilizando-se o método do BIC e seleciona o melhor.
- **Etapa 8:** De posse do valor da aptidão, verifica-se se existe algum elemento que atenda o critério de parada. Caso algum elemento atenda o critério de parada, o processo de identificação é finalizado, caso contrário, volta para a etapa 3.
- **Etapa 9:** Mede a capacidade que o modelo tem de representar o sistema e caso ele seja adequado o processo de identificação é finalizado, caso contrário, volta para a etapa 2.
- **Etapa 10:** Finaliza mostrando os resultados na tela.

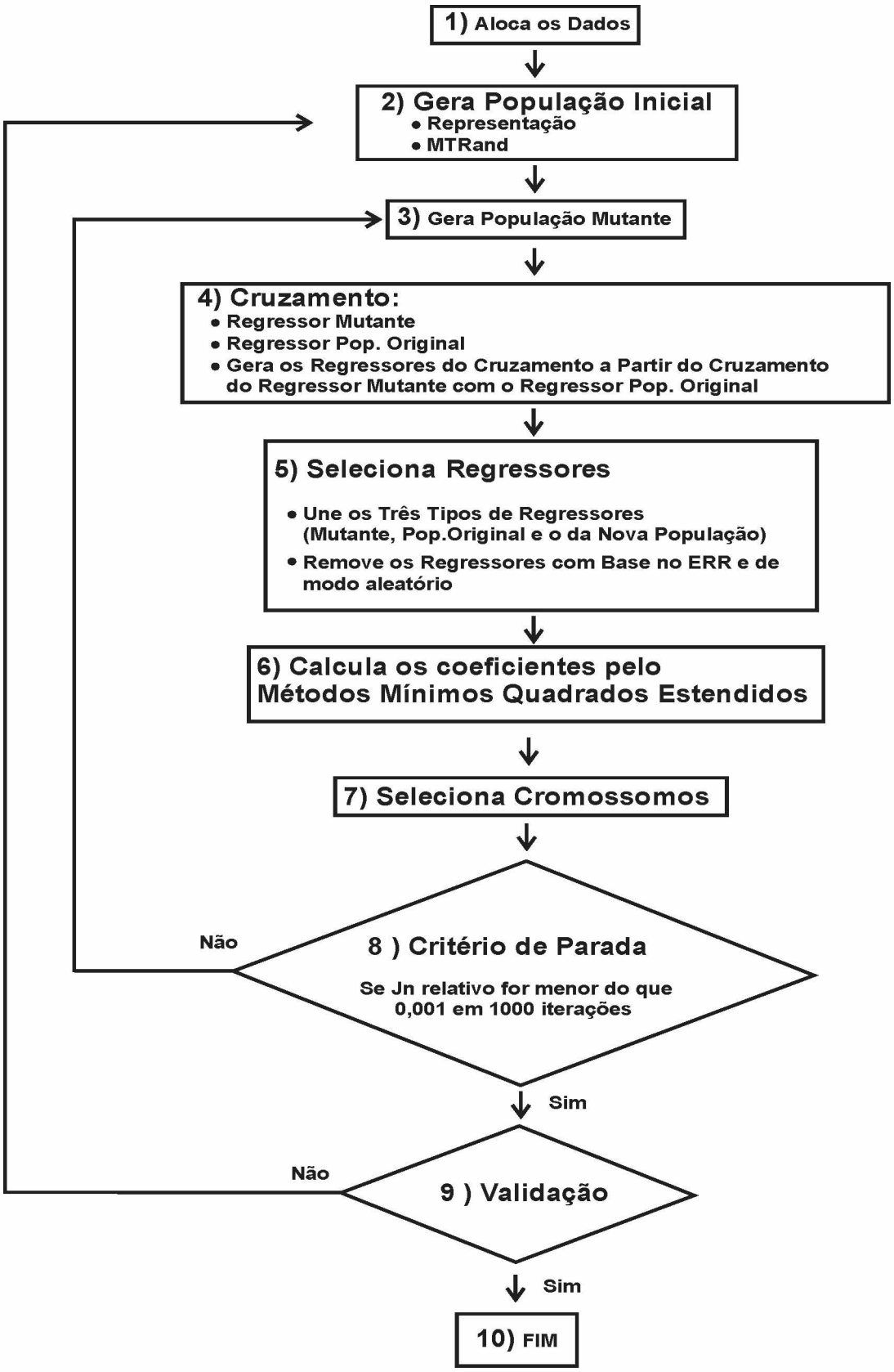


Figura 3.1 – Fluxograma da metodologia proposta de identificação de sistemas.

3.2 - DETALHAMENTO DAS ETAPAS

O detalhes de como foram implementadas as etapas são apresentados nos subitens seguintes.

3.2.1 - ALOCAÇÃO DOS DADOS NO APLICATIVO

Os dados foram dispostos de forma que cada linha contenha os dados de cada variável, sendo as primeiras linhas as variáveis dependentes e as demais as variáveis independentes. As colunas representam os atrasos de amostragem de forma crescente com a ordem da coluna, ou seja, quanto maior é o índice da coluna maior é o atraso, conforme Equação (3.1).

$$\begin{bmatrix} y_1(k) & y_1(k-1) & \dots & y_1(k-m-1) & y_1(k-m) \\ y_2(k) & y_2(k-1) & \dots & y_2(k-m-1) & y_2(k-m) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ y_{n-1}(k) & y_{n-1}(k-1) & \dots & y_n(k-m-1) & y_n(k-m) \\ y_n(k) & y_n(k-1) & \cdots & y_n(k-m-1) & y_n(k-m) \\ u_1(k) & u_1(k-1) & \cdots & u_1(k-m-1) & u_1(k-m) \\ u_2(k) & u_2(k-1) & \dots & u_2(k-m-1) & u_2(k-m) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ u_{n-1}(k) & u_{n-1}(k-1) & \cdots & u_{n-1}(k-m-1) & u_{n-1}(k-m) \\ u_n(k) & u_n(k-1) & \dots & u_n(k-m-1) & u_n(k-m) \end{bmatrix}_{m\,x\,n} \tag{3.1}$$

3.2.2 - GERAÇÃO DA POPULAÇÃO INICIAL

É criada uma lista de população sendo uma população para cada saída solicitada, já que o sistema proposto é capaz de identificar um sistema multivariável do tipo MIMO (múltiplas entradas e múltiplas saídas).

No trabalho proposto, os cromossomos que compõem cada população são criados e representados por uma matriz de três dimensões, onde a posição das linhas indica a variável, as colunas o index do atraso, as paginas os regressores e cada célula indexada pela respectiva linha, coluna e pagina, armazena o valor do expoente, conforme a Figura 3.2.

Então, se o valor do expoente na posição lx1x1 for 2, significa que o termo representado é $y_1(k-1)^2$. Se o expoente for igual a zero, significava que o respectivo termo não esta no conjunto solução.

Para não alocar memória desnecessariamente, representando os termos com expoente igual a zero, utilizou-se uma representação binária para alocação desta matriz no sistema proposto, conforme Tabela 3.1.

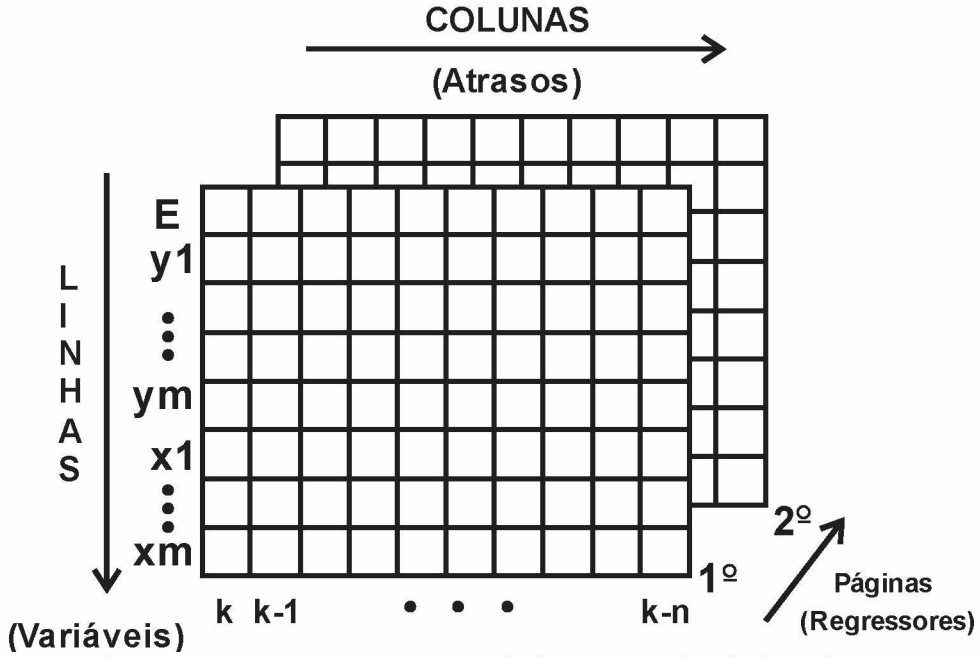


Figura 3.2 – Representação cromossômica por matriz tri-dimensional.

Tabela 3.1: Representação dos regressores no aplicativo.

Numerador / Denominador	Regressor que pertence	Atraso	Variável
1bit	9bit	11bit	11bits

Lembrando que se deve tomar certos cuidados, pois a representação de ε(k) ou de y(k), ou seja, a própria saída que deseja-se identificar ou o resíduo obtido da amostragem em questão, ser um termo representável em algum regressor da estrutura obtida, o que não pode ocorrer.

Partindo deste princípio a representação dos cromossomos deu-se por uma estrutura de dados constituída conforme Figura 3.3.

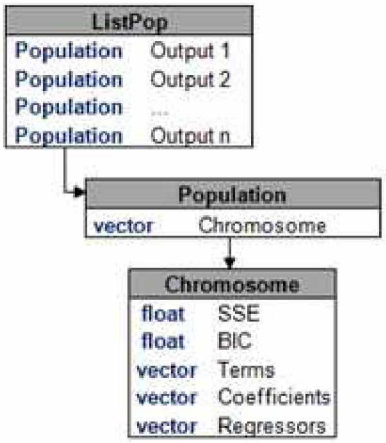


Figura 3.3 – Estrutura das populações.

De forma aleatória, utilizando o algoritmo MTRand, é gerado os regressores, que compõem os cromossomos, com seus termos e coeficiente tomando-se o cuidado de identificá-los como numerador ou denominador, conforme a Equação (3.2):

$$P_{o,y1} = X_{i,0} = x_{i,j,0} = \mathrm{rand}_j[0,1](u_j - l_j) + l_j \tag{3.2}$$

Onde $X_{i,0}$ é o i-ésimos cromossomo da população inicial, $x_{i,j,0}$ é o j-ésimos parâmetro do i-ésimos cromossomo, podendo ser este parâmetro o coeficiente do regressor j-ésimos (C_j) ou o expoente de um dos termos do regressor j-ésimos ($n_{j,term}$) e l_j e u_j é o limite inferior e o superior respectivamente do parâmetro j-ésimos.

Toda a parte de geração de números aleatórios, inclusive a geração dos cromossomos, são feitas pelo algoritmo Mersenne twister, também denominado MTRand. Este método consiste de um gerador de números pseudo-aleatórios desenvolvido em 1997 por Makoto Matsumoto e Nishimura Takuji (Matsumoto & Nishimura, 1998), que fornece uma geração rápida e de alta qualidade de números pseudo-aleatórios, tendo sido projetado especificamente para corrigir muitas das falhas encontradas nos algoritmos antigos.

Há pelo menos duas variantes comuns do algoritmo, diferindo apenas no tamanho dos números primos de Mersenne utilizados. O mais novo e mais comumente usado é o Mersenne Twister MT19937, com 32-bit de comprimento de palavra e utilizado neste trabalho. Existe também uma variante com 64 bits de comprimento de palavra, denominado de MT19937-64, o qual gera uma sequência diferente.

3.2.3 - GERAÇÃO DA POPULAÇÃO MUTANTE

O primeiro passo para gerar a população mutante é utilizar o operador de mutação que consiste em selecionar três indivíduos diferentes da população e, em seguida realiza-se a operação conhecida como mutação diferencial RAND-TO-BEST-MODIFICADO apresentada na Equação (3.3).

$$V_{i,G} = X_{best,G} + \mu\,(X_{new,G} - X_{best,G}) + F\,(X_{r_2,G} - X_{r_3,G}) \tag{3.3}$$

O algoritmo DE é aplicado em problemas de otmização, no entanto, na metodologia proposta ele foi aplicado para resolver problema de seleção de estrutura, que consiste me um problema de combinação.

Na operação RAND-TO-BEST tradicional, é refinado os expoentes apenas dos regressores criados na população inicial. Deste modo, caso um regressor necessário para representar o sistema estudado não tenha sido criando na população inicial, ele não irá aparecer no resultado final. Por isso, optou-se por modificar o algoritmo RAND-TO-BEST tradicional, adicionando-se um novo cromossomo denominado $X_{new,G}$ que também participa da operação de mutação, como mostrado na Figura 3.4.

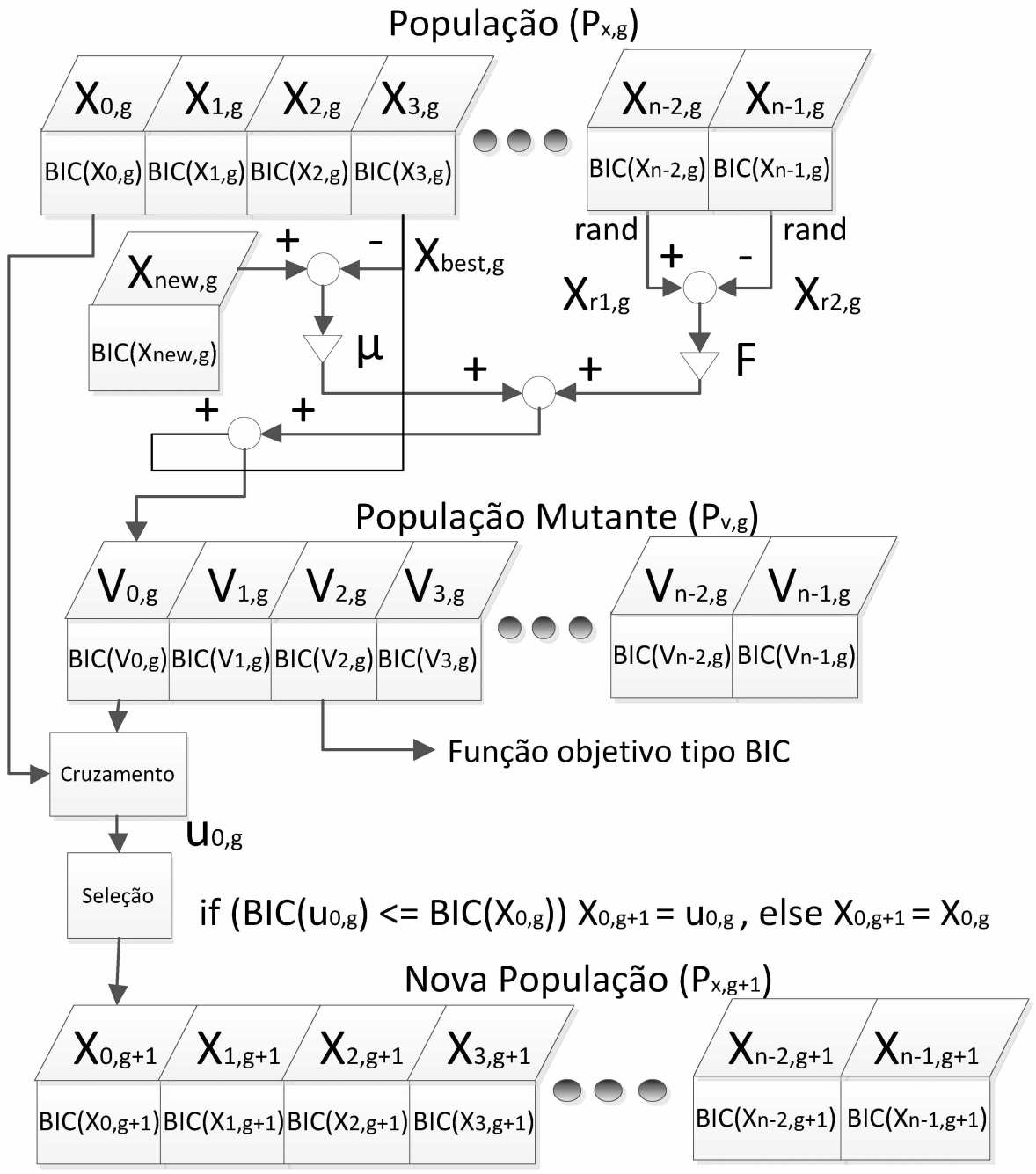


Figura 3.4 – Fluxograma do Algoritmo Diferencial Evolutivo Implementado.

O Fluxograma do Algoritmo Diferencial Evolutivo Implementado é semelhante ao DE tradicional. Ele difere no processo de mutação, pois utiliza o algoritmo RAND-TO-BEST modificado.

Esta diferença proporciona a metodologia proposta um vantagem importante sobre as metodologias propostas pela literatura, pois ela passa a ter um domínio

infinito de regressores possíveis de serem testados durante a etapa de seleção. Vale salientar que as metodologias atuais utilizam um domínio pré-determinado de regressores para serem testados, o que dificulta a obtenção dos modelos, principalmente se o método utilizado para obter este domínio não comtemplar os regressores necessários.

Atualmente, o modelo NCARMA possui a maior capacidade de representação e consequentemente é o mais complexo para se obter (Santos, 2003). Todavia, pelo fato da estrutura gerada pela metodologia proposta possuir expoentes reais, ela possui maior representatividade de sistemas, conforme mostrado na Figura 3.5, se comparada com as demais estruturas apresentadas na literatura até o momento.

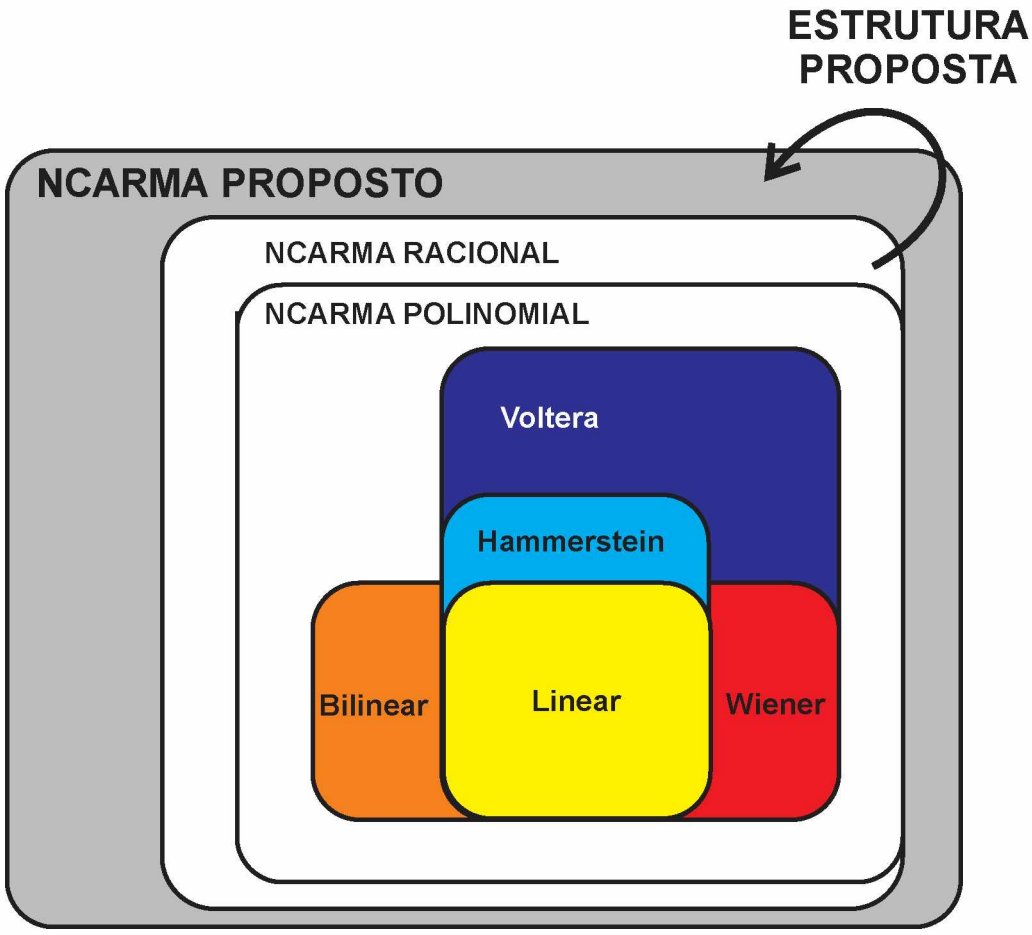


Figura 3.5 – Diagrama de comparação dos modelos.

3.2.4 - CRUZAMENTO

Inicialmente a técnica de mutação escolhida pode levar à convergência prematura da população já que sempre o melhor termo é utilizado como vetor base, mesmo que características de um vetor novo seja inserido em cada processo de cruzamento. Então, para o processo de cruzamento optou-se por mudar uma faixa de elementos garantindo assim que um conjunto de características do vetor de avaliação sejam inseridas no vetor mutante.

A técnica consiste em, inicialmente, efetuar um teste para cada cromossomo em cada geração, escolhendo-se aleatoriamente se o cruzamento será em nível de termos ou de coeficientes ou de ambos. Feito isto, uma faixa neste vetor é selecionada e, então, os termos e ou coeficientes que estão nesta faixa, são substituídos pelos seus correspondentes do cromossomo avaliado no cromossomo mutante.

Caso não exista correspondentes no cromossomo em cruzamento, então, eles são removidos do cromossomo mutante e os que somente existirem no cromossomo em cruzamento são simplesmente inseridos no mesmo. É garantido que pelo menos um termo do cromossomo em cruzamento seja substituído gerando os regressores do cruzamento.

3.2.5 - SELEÇÃO DOS REGRESSORES

No processo de Seleção dos Regressores, para cada cromossomo da população inicial insere-se os regressores gerados no processo de mutação da Etapa 3 e de cruzamento da Etapa 4, gerando-se um novo conjunto de cromossomos.

Então, por meio do algoritmo ERR, seleciona-se os melhores cromossomos no novo conjunto de cromossomos gerados e, posteriormente, realiza-se uma nova seleção de forma aleatória, obtendo-se como resultado o conjunto de regressores que

farão parte da população (SISO) ou das populações (MIMO) que serão avaliadas nas próximas etapas.

Esta segunda seleção, feita aleatoriamente, é para corrigir uma característica do ERR que quando testado com dados contaminados por ruídos com valores consideráveis, ou entradas não suficientemente excitantes seleciona termos espúrios levando o modelo final à super-parametrização (Wei, 2008). Esta técnica também é uma contribuição desta tese, e a ação conjunta do BIC e desta seleção aleatória dos regressores que foram selecionados pelo ERR converge para um modelo mais adequado, conforme será confirmado no capitulo posterior.

3.2.6 - CÁLCULO DOS COEFICIENTES

Nesta etapa são calculados os coeficientes dos regressores de cada cromossomo, utilizando o método dos mínimos quadrados estendidos, para que seja possível avaliar o modelo obtido na próxima etapa. Os detalhes da metodologia utilizada no cálculo dos coeficientes são apresentados no Anexo D.

3.2.7 - SELECAO DOS CROMOSSOMOS

Na metodologia proposta optou-se por utilizar dois objetivos no processo de avaliação do cromossomo, ou seja, é necessário achar um índice que avalia ao mesmo tempo, a obtenção do menor valor para o Somatório do Erro Quadrático – SSE (Sum of Squared Error) vide Equação (3.4), e a menor quantidade de regressores na estrutura.

Para isto foi então utilizado o Critério de Informação Bayesiana (BIC) como função para o cálculo da aptidão apresentado na equação (3.6), porém para calculá-la

é necessária a função custo básica dos mínimos quadrados que é dado pela equação (3.5).

$$\mathrm{SSE} = \sum_{t=1}^{N} [y(t) - \hat{y}(t)]^2 \tag{3.4}$$

$$J_n = \frac{1}{N} * \mathrm{SSE} \tag{3.5}$$

$$\mathrm{BIC} = N * \ln[J_n] + p * \ln[N] \tag{3.6}$$

Onde: $y(t)$ é o valor desejado numa determinada amostragem, $\hat{y}(t)$ é o valor obtido pelo modelo avaliado, N é o número de medidas e p a quantidade de regressores.

Para cada cromossomo é calculado a aptidão utilizando-se o algoritmo BIC e, por fim, a seleção é feita, escolhendo aquele que apresente a melhor aptidão entre o cromossomo pai e o filho, conforme Equação (3.7).

$$U_{i,G+1} = \begin{cases} U_{i,G}, & \text{if } BIC(U_{i,G}) < BIC(X_{i,G}) \\ X_{i,G}, & \text{otherwise} \end{cases} \tag{3.7}$$

Acima, $U_{i,G}$ é o vector de julgamento, $U_{i,G+1}$, é o vector alvo e BIC(VETOR) é a função de aptidão calculada pelo BIC de um dado vetor.

3.2.8 - CRITÉRIO DE PARADA

O processo finaliza quando o algoritmo não está progredindo a aptidão das novas populações acima de um valor informado pelo usuário. No algoritmo implementado, se a aptidão do melhor cromossomo não sofrer um progresso maior que 0,001 no seu valor de BIC em 100 interações o processo de identificação é finalizado.

3.2.9 - VALIDAÇÃO DO MODELO

Os testes de validação medem a capacidade que um modelo tem de representar um sistema, estas técnicas são utilizadas no processo final de identificação. Os testes de modelos podem ser divididos em duas vertentes: (i) validação estatística e (ii) validação dinâmica.

Na validação estatística é feita com testes de autocorrelação e correlação entre os resíduos de identificação e o sinal de saída. Um modelo será válido estatisticamente se os resíduos de identificação não contiverem correlação com ele mesmo e nem com a saída. Na validação de modelos não-lineares, devem ser verificadas as seguintes identidades (Billings e Voon, 1983; Billings e Voon, 1986; Billings e Zhu, 1994):

$$\Phi_{\xi\xi}(\tau_1) = E\{\xi(t)\xi(t-\tau_1)\} = \delta(\tau_1), \tag{3.8}$$

$$\Phi_{\xi u}(\tau_1) = E\{\xi(t)u(t-\tau_1)\} = 0, \forall\tau_1 \tag{3.9}$$

$$\Phi_{u^2\xi}(\tau_1) = E\{(u^2(t) - E\{u^2(t)\})\xi(t-\tau_1)\} = 0, \forall\tau_1 \tag{3.10}$$

$$\Phi_{u^2\xi^2}(\tau_1) = E\{(u^2(t) - E\{u^2(t)\})\xi^2(t-\tau_1)\} = 0, \forall\tau_1 \tag{3.11}$$

$$\Phi_{\xi'\xi'}(\tau_1) = E\{(\xi(t) - E\{\xi(t)\})(\xi(t-\tau_1) - E\{\xi(t)\})\} = 0, \forall\tau_1 \tag{3.12}$$

$$\Phi_{\xi'\xi^{2\prime}}(\tau_1) = E\{(\xi(t) - E\{\xi(t)\})(\xi^2(t-\tau_1) - E\{\xi^2(t)\})\} = 0, \forall\tau_1 \tag{3.13}$$

$$\Phi_{\xi^{2\prime}\xi^{2\prime}}(\tau_1) = E\{(\xi^2(t) - E\{\xi^2(t)\})(\xi^2(t-\tau_1) - E\{\xi^2(t)\})\} = \delta(\tau_1), \forall\tau_1 \tag{3.14}$$

$$\Phi_{(y\xi)'\xi^{2\prime}}(\tau_1) = E\{(y(t)\xi(t) - E\{y(t)\xi(t)\})(\xi^2(t-\tau_1) - E\{\xi^2(t)\})\} = 0\,, \forall\tau_1 \tag{3.15}$$

$$\Phi_{(y\xi)'u^{2\prime}}(\tau_1) = E\{(y(t)\xi(t) - E\{y(t)\xi(t)\})(u^2(t-\tau_1) - E\{\xi u^2(t)\})\} = 0\,, \forall\tau_1 \tag{3.16}$$

Onde $\delta(\tau_1)$, é a função delta de Dirac e a apóstrofe indica que a média foi subtraída dos sinais. Baseado no comprimento do registro de dados disponíveis, N, é definido um intervalo probabilístico de 95% de confiança, dentro do qual as funções de correlação devem se manter para serem consideradas nulas. Os limites do intervalo de confiança de 95%, para funções normalizadas pelo desvio padrão, são: ± 1,96 N (Billings e Voon, 1983; Billings e Voon, 1986).

A validação estatística garante que não existem correlações não modeladas do sistema presentes nos resíduos. No entanto, mesmo atendendo a estes critérios não se pode assegurar que os modelos validados venham a apresentar o mesmo comportamento dinâmico do sistema original (Aguirre e Billings, 1994a), o que é verificado na validação dinâmica. Neste trabalho, este teste consistirá em aplicar o modelo obtido em uma massa de dados diferente daquela usada para a identificação. Em seguida será traçado um histograma dos seus resíduos para verificar se eles apresentam uma distribuição normal em torno dos resíduos com valor zero.

3.3 - CONCLUSÃO

Este capítulo apresentou a metodologia proposta. Mostrou que ela utilizou algoritmos diferenciais evolutivos para selecionar a estrutura. Além disso, foi necessário alterar o algoritmo RAND-TO-BEST no processo de mutação para aumentar a variabilidade genética.

O algoritmo BIC foi implementado para calcular a aptidão do algoritmo evolutivo, visando a melhora das estruturas tanto no tamanho quanto no SSE.

Além disso, pelo fato da estrutura gerada pela metodologia proposta possuir expoentes reais, ela possui maior representatividade de sistemas.

Por fim, foram apresentadas as técnicas de validação utilizadas na metodologia proposta.

CAPÍTULO 4

AVALIAÇÃO DO ALGORITMO PROPOSTO

4.1 - CONSIDERAÇÕES INICIAIS

O objetivo deste capítulo é avaliar o algoritmo proposto que uni técnicas estocásticas com técnicas determinísticas para a seleção da estrutura num processo de identificação de sistemas. Para o processo de estimação de parâmetros o presente trabalho não apresenta novas contribuições, consequentemente, todos os testes apresentados consistirão em validar a estrutura selecionada.

Para a validação do algoritmo de seleção das estruturas utilizou-se de modelos artificiais com características específicas que gerem problemas para a seleção da estrutura e assim possa se avaliar sua capacidade em superar estas dificuldades. As técnicas de validação de modelos serão apresentadas, mas só serão utilizadas se a estrutura obtida não for exatamente a mesma da estrutura real, se a estrutura for a mesma a validação se torna desnecessária.

Os modelos artificiais utilizados neste trabalho foram, em sua maioria, extraídos da literatura atual.

4.2 - MODELAGEM DE UM SISTEMA LINEAR POLINOMIAL

Como o processo de identificação é um processo recursivo existe a necessidade de se definir um critério de parada, neste projeto optou-se por executar o algoritmo proposto até que a taxa de variação da sua aptidão seja menor que 0.001 durante 100 iterações consecutivas.

Optou-se também por simular os modelos 10 vezes e em cada conjunto de 2 operações novos 500 pontos de entrada e ruído foram gerados de forma aleatória.

Destes 500 pontos 250 foram usados para a identificação e os outros 250 para a validação.

Nesta seção o procedimento de identificação foi aplicado a dois modelos lineares artificiais com estrutura polinomial conforme equações (3.4) e (4.2). Estes modelos foram retirados de (Wei, 2008) onde são usados para ilustrar alguns problemas que surgem quando os dados de treino são contaminados por ruídos com valores consideráveis, ou entradas não suficientemente excitantes. Nesta referencia é apresentado uma falha do ERR que leva à super-parametrização quando utilizado para a seleção dos regressores em dados com estas características.

Considerando que o sistema proposto neste trabalho utiliza-se do ERR, mas não somente dele para a seleção da estrutura, então, optou-se por avalia-lo com estes dois modelos.

$$y(k) = -1.7y(k-1) - 0.8y(k-2) + u(k-1) + 0.8u(k-2) + e(k) \quad (4.1)$$

$$y(k) = 0.7y(k-1) - 0.1y(k-2) + u(k-1) \quad (4.2)$$

Para a equação (3.4) a entrada u(k) é uniformemente distribuída em [-2,2], com um ruído e(t) ~N(0,0.1^2). Já para a equação (4.2) a entrada u(k) é um processo Auto Regressivo dado por: $u(k) = 1.6u(k-1) - 0.6375u(k-2) + \xi(k)$, com $\xi(k)$ ~ N(0,1).

Em todos os testes efetuados, tanto para o modelo da equação (4.1) quanto o da (4.2) os regressores originais foram identificados e em nenhum caso algum regressor que não existisse no modelo original foi contemplado no modelo identificado, não apresentando assim super-parametrização. Na Figura 2.2 um dos resultados para a equação (4.1) é apresentado e na Figura 4.2 um dos resultados para a equação (4.2) é apresentado.

É importante destacar que em nenhum dos casos anteriores foi definido um domínio de regressores a serem testados. O algoritmo proposto neste trabalho gera estes regressores de forma aleatória, e ainda testa a possibilidade de estruturas racionais contemplarem uma solução para o conjunto de dados e ou de algum termo possuir expoentes reais. Estas características não causaram nenhum desvio que levasse o modelo a um resultado diferente do real, mesmo para modelos problemáticos tais como os das equações (4.1) e (4.2).

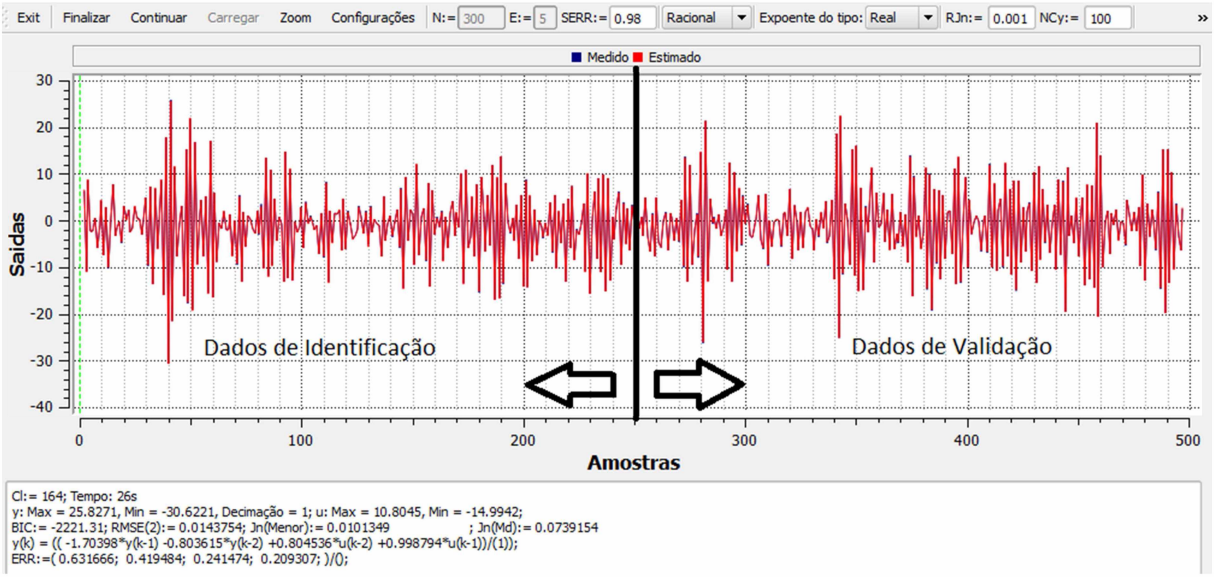


Figura 4.1 – Gráfico Medido X Estimado para o modelo da equação (4.1).

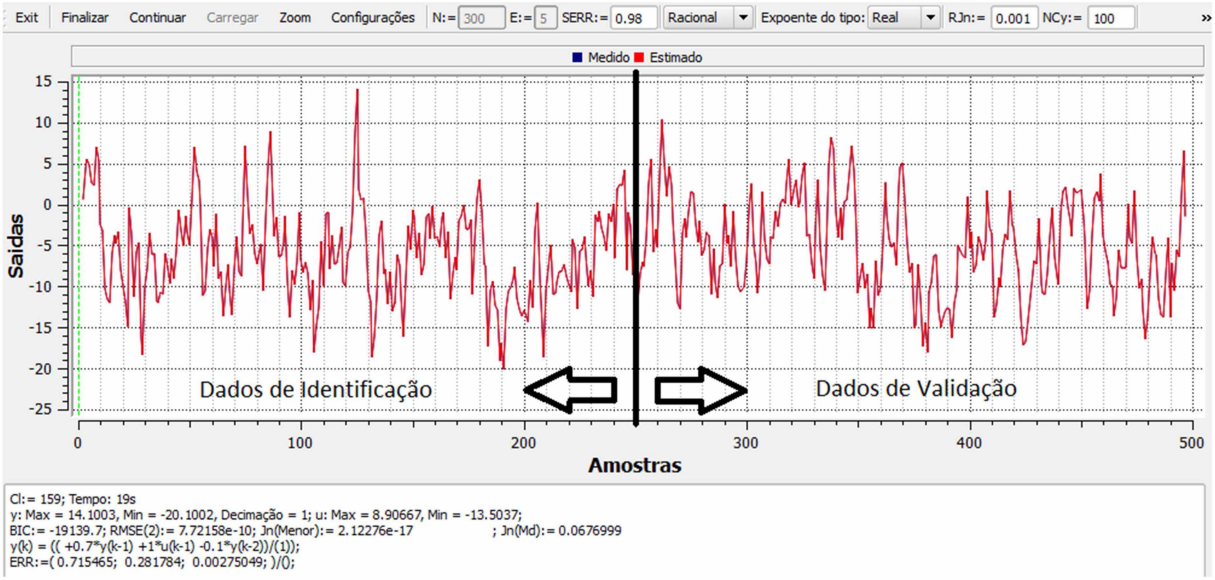


Figura 4.2 – Gráfico Medido X Estimado para o modelo da equação (4.2).

Como a rotina de seleção da estrutura, que é o objeto de estudo deste trabalho, identificou todos os regressores em todas as simulações não ouve a necessidade de executar nenhum teste de validação do sistema estimado.

4.3 - MODELAGEM DE UM SISTEMA NÃO LINEAR POLINOMIAL

O sistema artificial da equação (4.3) foi identificado utilizando o mesmo critério de parada da seção 4.2.

$$y(k) = 0.5y(k-2) + 0.7y(k-1)u(k-1) + 0.6u(k-2)^2 + 0.2y(k-1)^3 - 0.7y(k-2)u(k-2)^2 + e(k) \quad (4.3)$$

Onde: u(k) é uniformemente distribuída em [-1,1], com um ruído e(t) ~$N(0,0.02^2)$.

O modelo avaliado nesta seção foi retirado de (Billings K. Z., 1997), onde é discorrido sobre algumas singularidades que este modelo apresenta de forma a levar o ERR a selecionar com altas taxas de redução de erro alguns regressores espúrios. No algoritmo proposto o regressor que é selecionado pelo ERR sofre uma outra seleção, sendo esta feita de forma aleatória, o que permite a remoção destes regressores espúrios. Já o BIC, como função de aptidão, garante que este novo sistema seja mais apto, uma vez que possui uma menor quantidade de regressores, o que induz o sistema a convergir para o modelo real. Em todas as simulações efetuadas neste sistema todos os regressores do modelo real foram contemplados e nenhum regressor espúrio foi inserido. Na Figura 4.3 um dos resultados é apresentado.

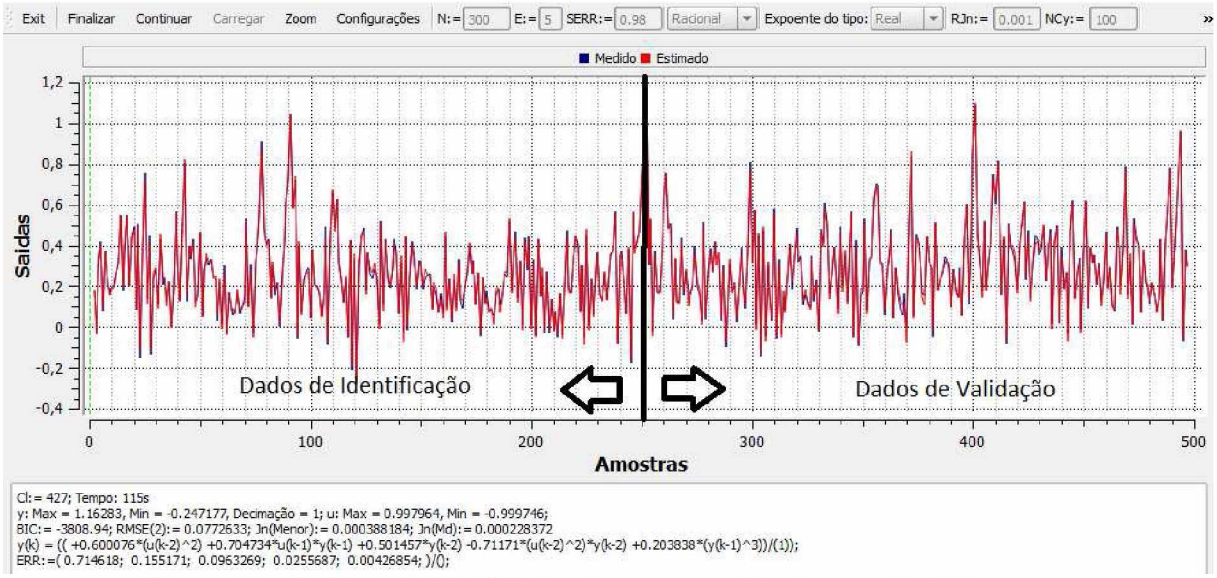


Figura 4.3 – Gráfico Medido X Estimado para o modelo da equação (4.3).

4.4 - MODELAGEM DE UM SISTEMA NÃO LINEAR RACIONAL SISO

As métricas de parada, testes e geração de sinais de entrada e de ruído são as mesmas utilizadas nos exemplos anteriores. O sistema artificial da equação (4.4) foi extraído de (Corrêa, 1997) onde foi utilizado para testar o algoritmo RME Modificado apresentado nesta referência.

$$y(k) = \frac{0.2y(k-1) + 0.1y(k-1)u(k-1) + u(k-1)}{1 + y(k-1)^2 + y(k-2)^2} + e(k) \qquad (4.4)$$

Onde: u(k) é uniformemente distribuído em [-1,1], e(k) é um ruído branco com média zero, variância $8{,}34x10^{-6}$, amplitude variando de $\pm 5{,}0x10^{-3}$ com uma relação sinal/ruído (SNR) de 73,38 dB.

Já neste caso, em 80% das vezes que ele foi testado todos os regressores eram encontrados e nos outros 20% alguns regressores espúrios eram. Então, optou-se por mudar a métrica de parada onde era 100 iterações passou-se para 1000 iterações e o resultado foi satisfatório, novamente obtendo 100% de conversão. A Figura 4.5 é um exemplo destes resultados.

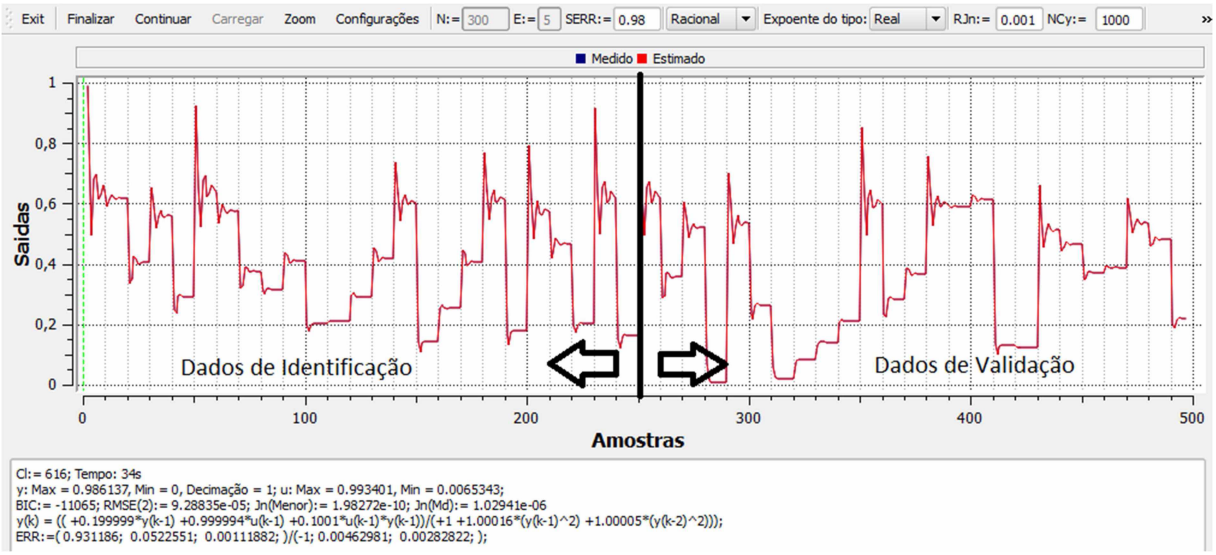


Figura 4.4 – Gráfico Medido X Estimado para o modelo da equação (4.4).

4.5 - MODELAGEM DE UM SISTEMA NÃO LINEAR RACIONAL MISO

A ideia neste tópico foi de verificar o desempenho deste sistema de identificação com múltiplas entradas. E para isto gerou-se um sistema conforme equação (4.5), nada foi modificado em relação aos testes anteriores com respeito ao critério de parada, de geração e testes dos sinais.

$$\mathrm{y(k)} = \frac{\mathrm{u1(k-1)^2 + u2(k-3)^3 + 2u1(k-1) + u2(k-2)}}{1 + \mathrm{u1(k-1) + y(k-1)^2}} + e(k) \qquad (4.5)$$

Nos primeiro testes, verificou-se que o algoritmo identificava o sistema, que os resíduos resultantes possuíam valores muito baixos, porém o modelo obtido era diferente do modelo real. Na validação dinâmica constatou-se que o modelo encontrado não era representativo conforme mostra a Figura 4.5.

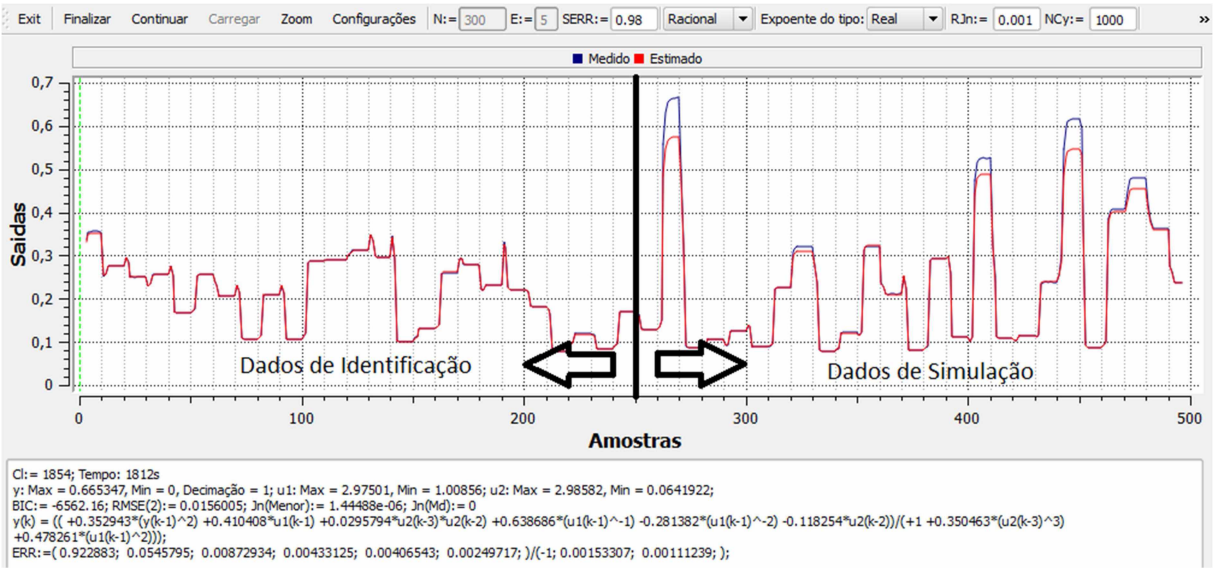


Figura 4.5 – Gráfico Medido X Estimado para o modelo com 500 pontos.

Supondo que os dados de identificação não estavam contemplando todas as informação necessárias para a identificação correta deste sistema, optou-se por aumentar o tamanho de amostras para 1000 pontos e repetir as simulações. Neste novo contexto os resultados mostraram-se muito melhores, porém o modelo obtido divergiu do modelo original.

Na Figura 4.6 é apresentado o resultado com melhor aptidão obtida de todos os testes. E na Figura 4.7 e apresentado o histograma onde pode ser verificado que a concentração de resíduos está entorno do eixo dos zeros, com uma representação próxima de uma gaussiana e com valores máximos muito pequenos.

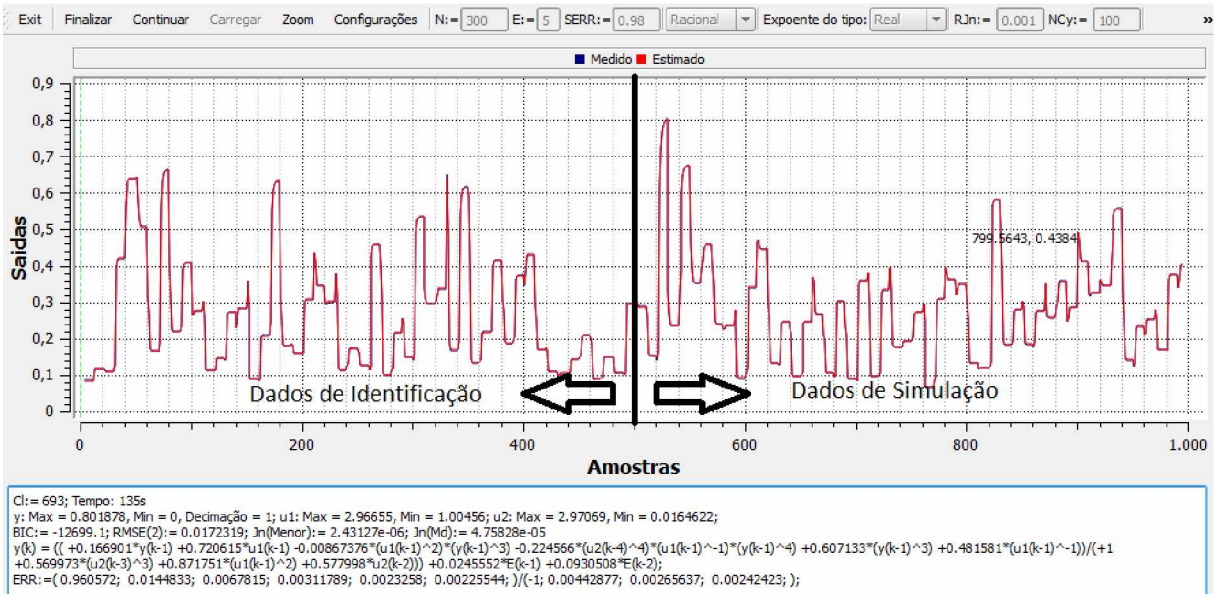


Figura 4.6 – Gráfico do melhor resultado Medido X Estimado com 1000 pontos.

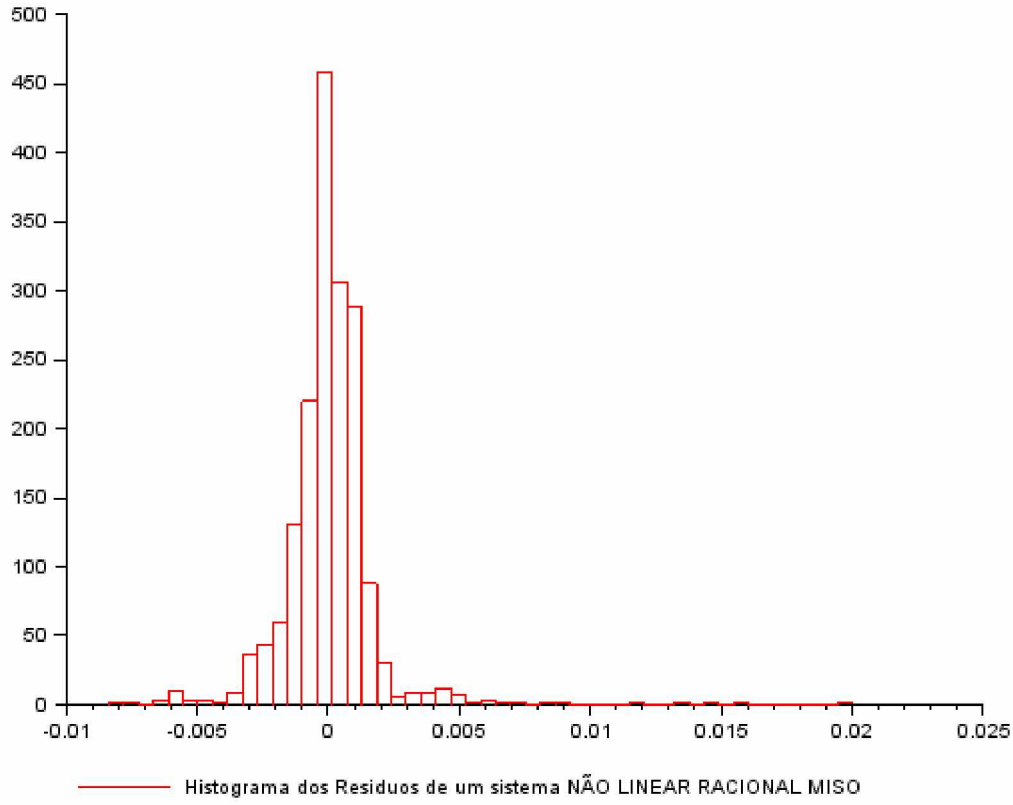
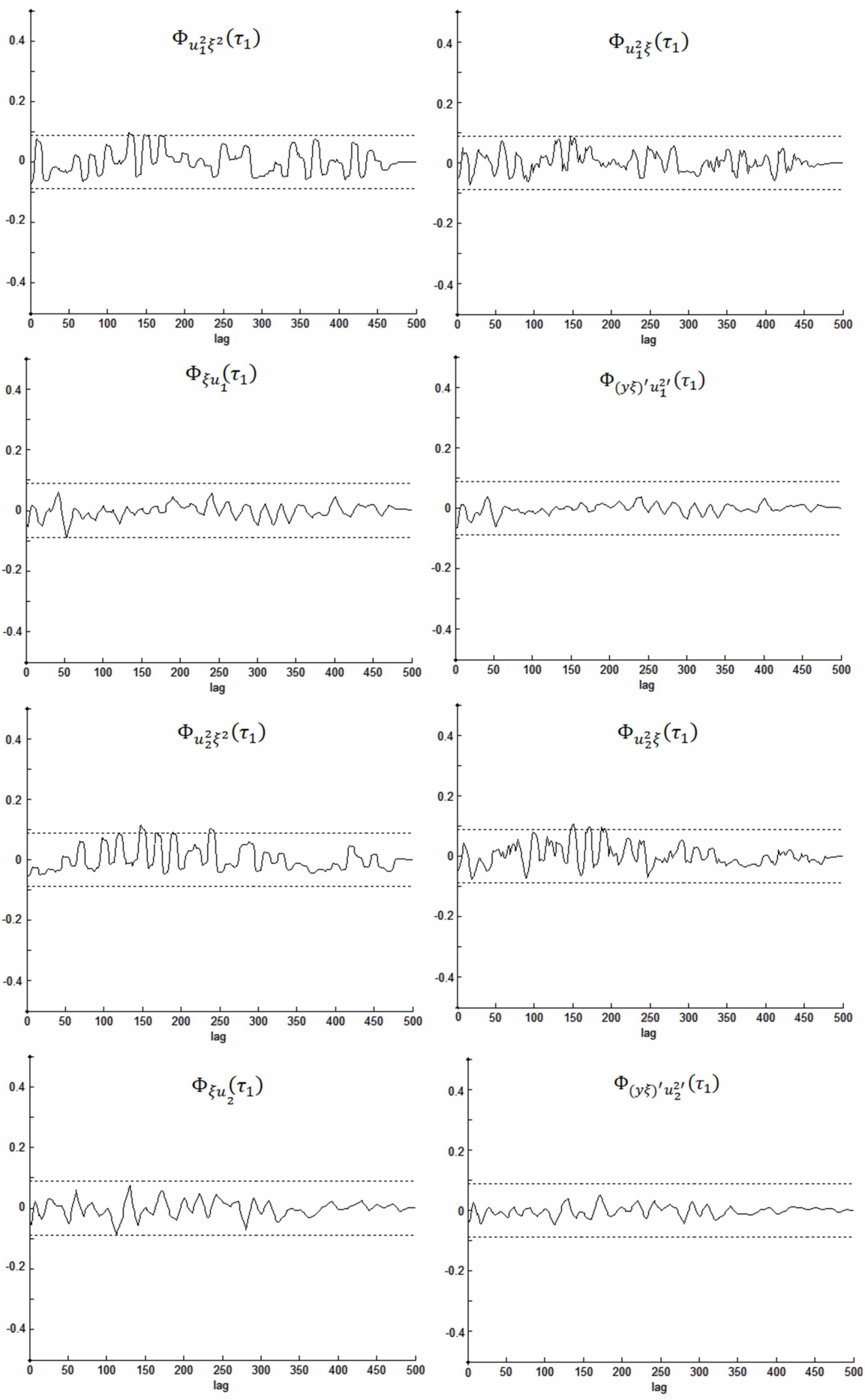


Figura 4.7 – Histograma do melhor resultado com 1000 pontos.

Os testes estatísticos foram feito e os gráficos são apresentados na Figura 4.8.


$\Phi_{u_1^2\xi^2}(\tau_1)$
$\Phi_{u_1^2\xi}(\tau_1)$
$\Phi_{\xi u_1}(\tau_1)$
$\Phi_{(y\xi)'u_1^{2\prime}}(\tau_1)$
$\Phi_{u_2^2\xi^2}(\tau_1)$
$\Phi_{u_2^2\xi}(\tau_1)$
$\Phi_{\xi u_2}(\tau_1)$
$\Phi_{(y\xi)'u_2^{2\prime}}(\tau_1)$
lag

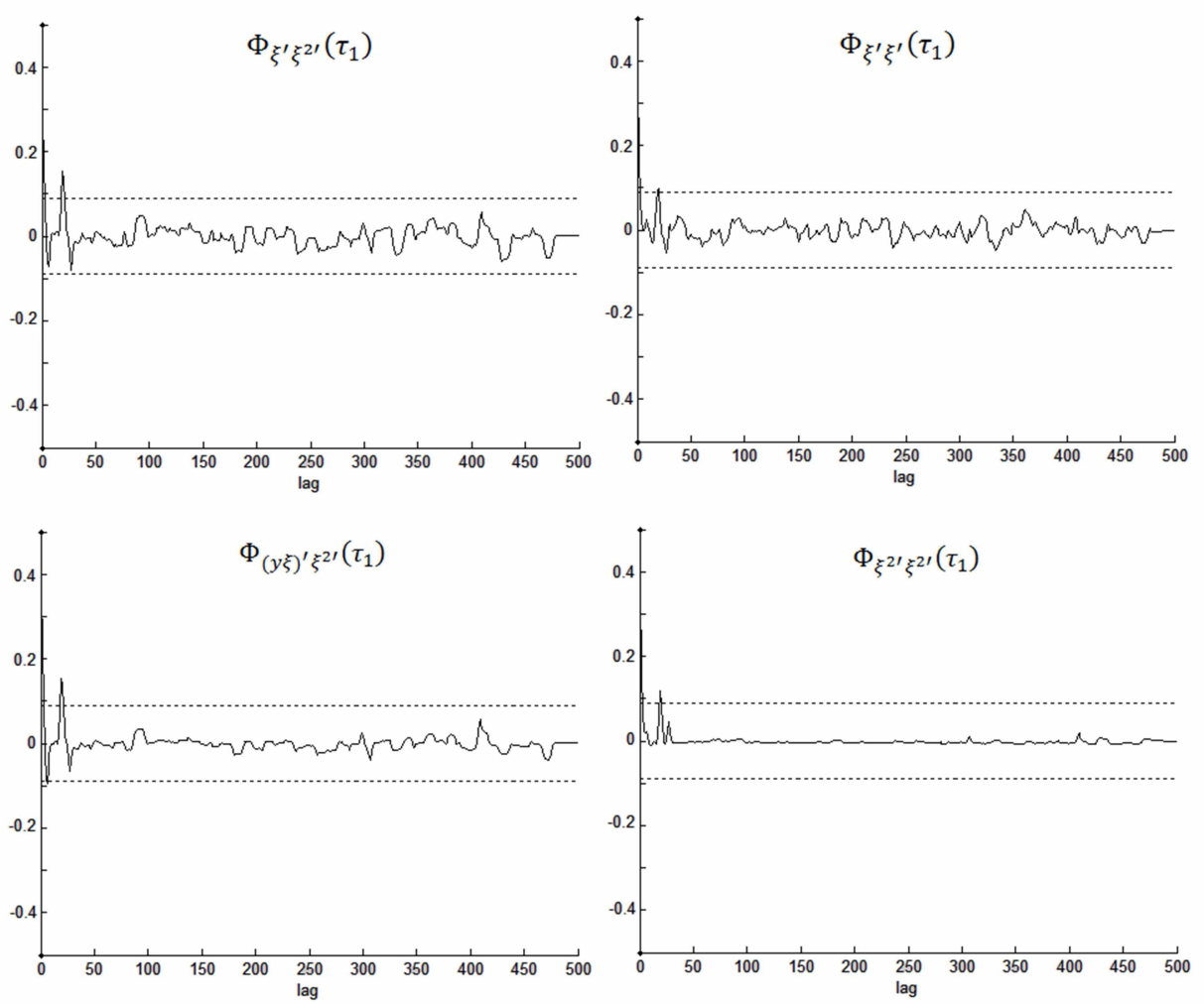


Figura 4.8 – Testes estocásticos.

Analisando estes testes podemos observar que poucos pontos estão fora da região de confiança garantindo assim que o modelo obtido representa estatisticamente o modelo real.

4.6 - MODELAGEM DE UM SISTEMA NÃO LINEAR RACIONAL MIMO

Este tópico tem como objetivo verificar o desempenho do algoritmo proposto para a identificação de sistemas com múltiplas entradas e múltiplas saídas. Os dados de teste foram gerados com a mesma métrica das sessões anteriores de acordo com as Equações (4.6) e (4.7).

$$y_1(k) = \frac{u_1(k-1)^2 + u_2(k-3) + y_1(k-1) + y_2(k-2)^3}{1 + u_1(k-1) + y_1(k-1)^2} + e(k) \quad (4.6)$$

$$y_2(k) = \frac{u_1(k-1)^2 + u_2(k-3) + y_2(k-1) + y_1(k-2)^3}{1 + u_1(k-1) + y_2(k-1)^2} + e(k) \quad (4.7)$$

Os resultados obtidos não apresentaram os mesmos regressores que o modelo real. Portanto testes dinâmicos e estatísticos foram aplicados e todos atenderam as expectativas. Na Figura 4.9 e na Figura 4.10 um dos resultados escolhidos de forma aleatória da saída 1 e da saída 2 respectivamente são apresentados.

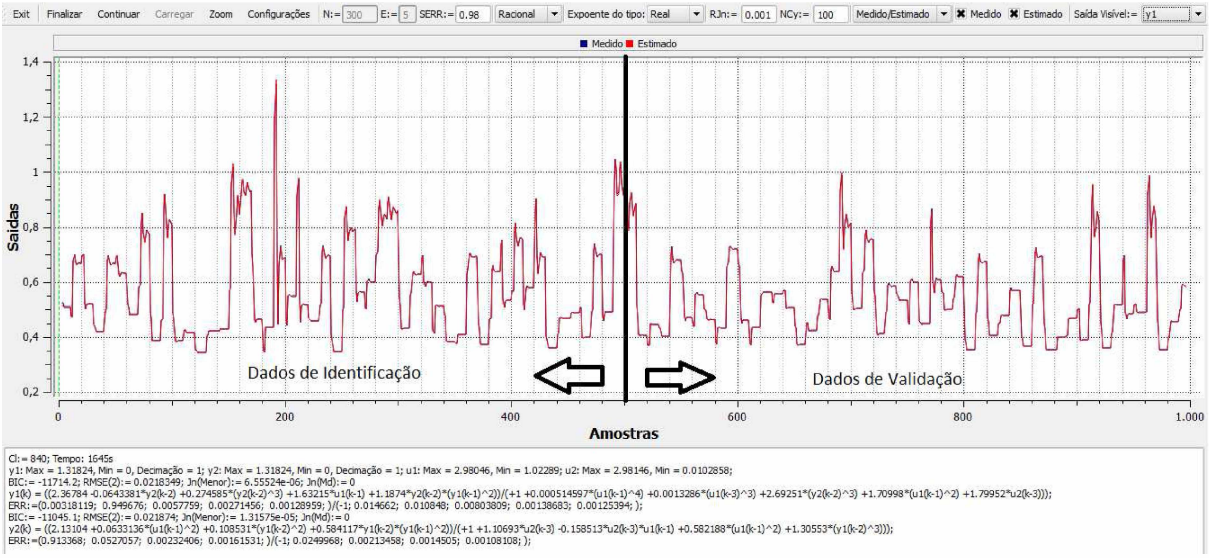


Figura 4.9 – Resultado para a saída 1.

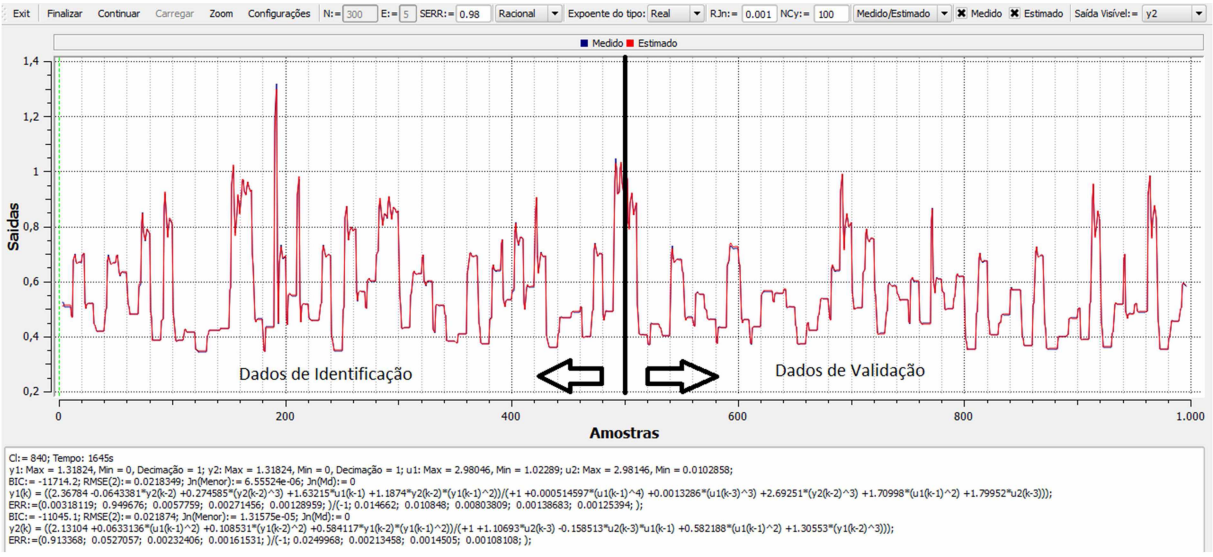


Figura 4.10 – Resultado para a saída 2.

Na Figura 4.11 estão apresentados o histograma dos resíduos das saídas 1 e 2 para o resultado escolhido.

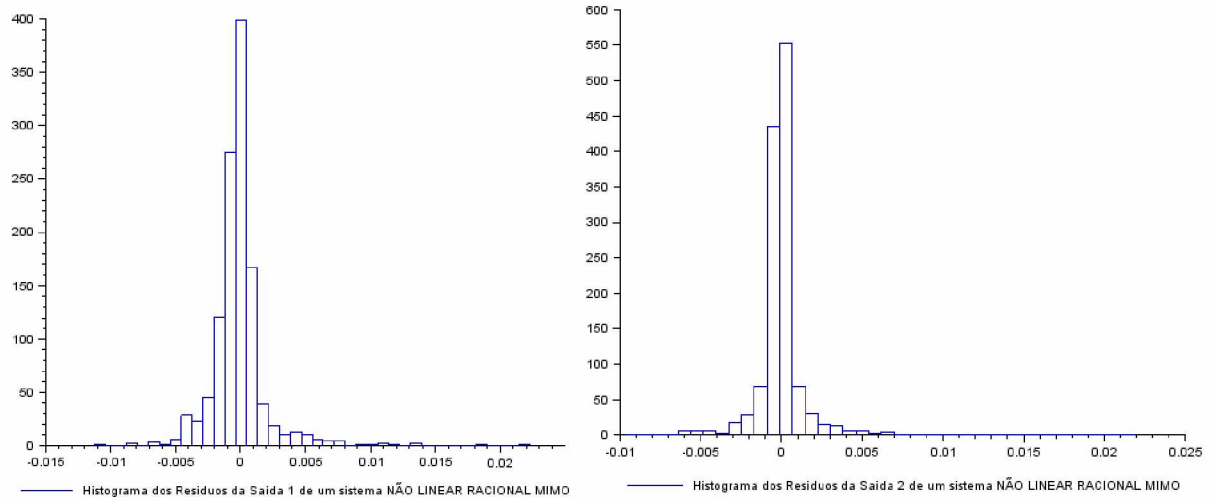
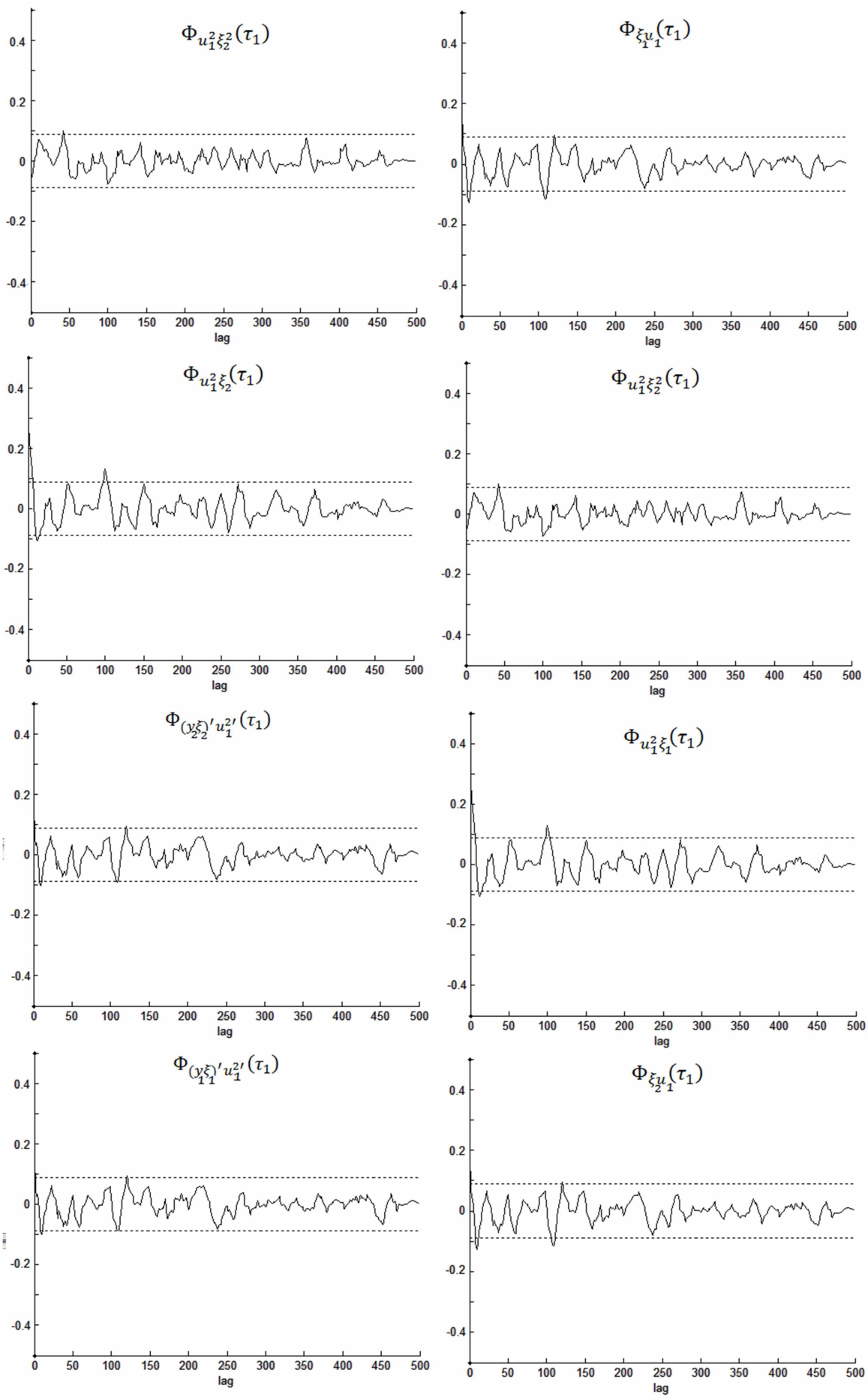
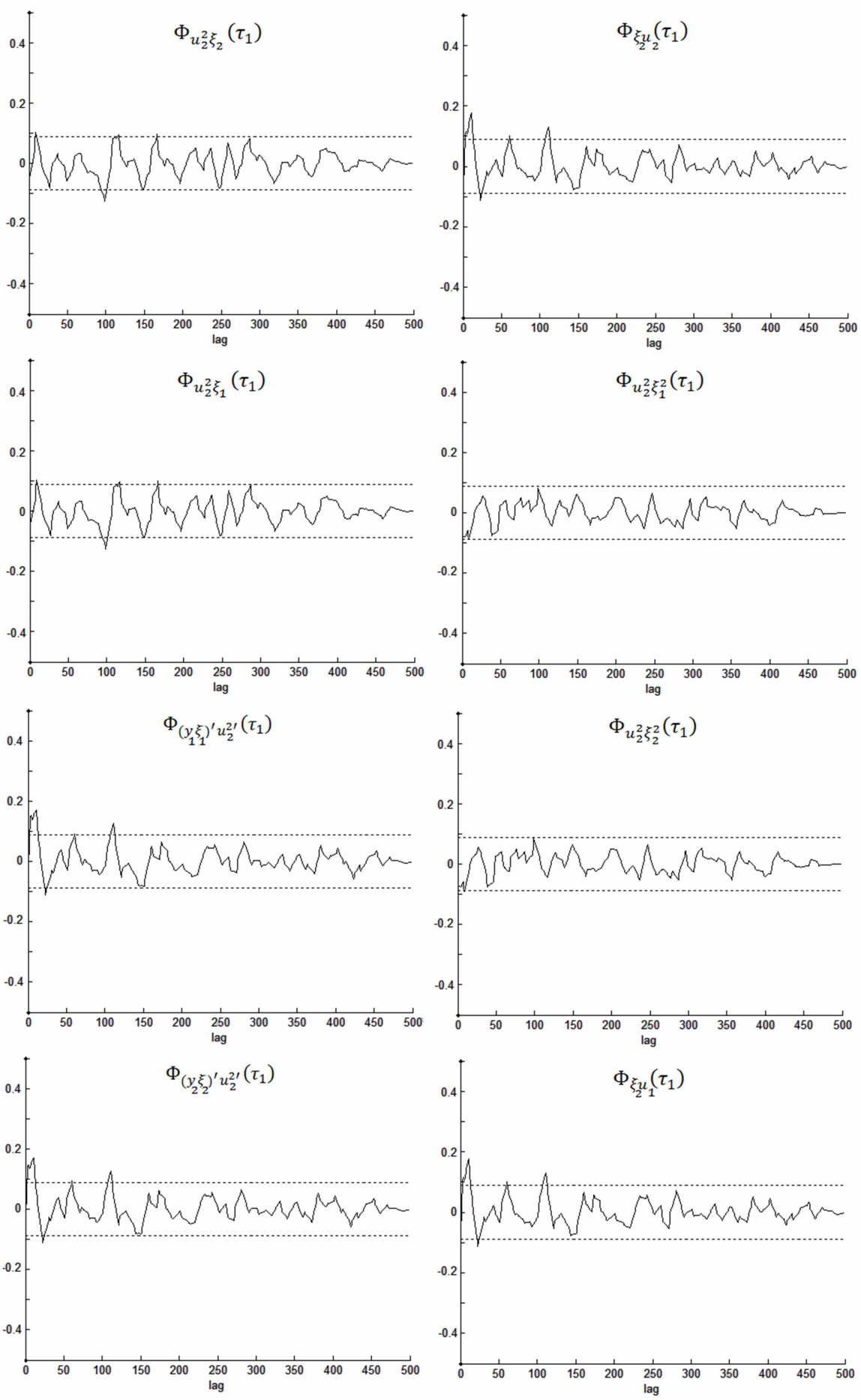


Figura 4.11 – Histograma dos resíduos das saídas 1 e 2.

A Figura 4.12 apresenta os testes estocásticos do modelo obtido. Nenhum dos testes ultrapassou a região de confiança garantindo que o modelo estatisticamente é equivalente ao sistema avaliado.


$\Phi_{u_1^2\xi_2^2}(\tau_1)$
$\Phi_{\xi_1 u_1}(\tau_1)$
$\Phi_{u_1^2\xi_2}(\tau_1)$
$\Phi_{u_1^2\xi_2^2}(\tau_1)$
$\Phi_{(y_2\xi_2)'u_1^{2'}}(\tau_1)$
$\Phi_{u_1^2\xi_1}(\tau_1)$
$\Phi_{(y_1\xi_1)'u_1^{2'}}(\tau_1)$
$\Phi_{\xi_2 u_1}(\tau_1)$
lag


$\Phi_{u_2^2\xi_2}(\tau_1)$
$\Phi_{\xi_2 u_2}(\tau_1)$
$\Phi_{u_2^2\xi_1}(\tau_1)$
$\Phi_{u_2^2\xi_1^2}(\tau_1)$
$\Phi_{(y_1\xi)'u_2^{2\prime}}(\tau_1)$
$\Phi_{u_2^2\xi_2^2}(\tau_1)$
$\Phi_{(y_2\xi)'u_2^{2\prime}}(\tau_1)$
$\Phi_{\xi_2 u_1}(\tau_1)$
lag

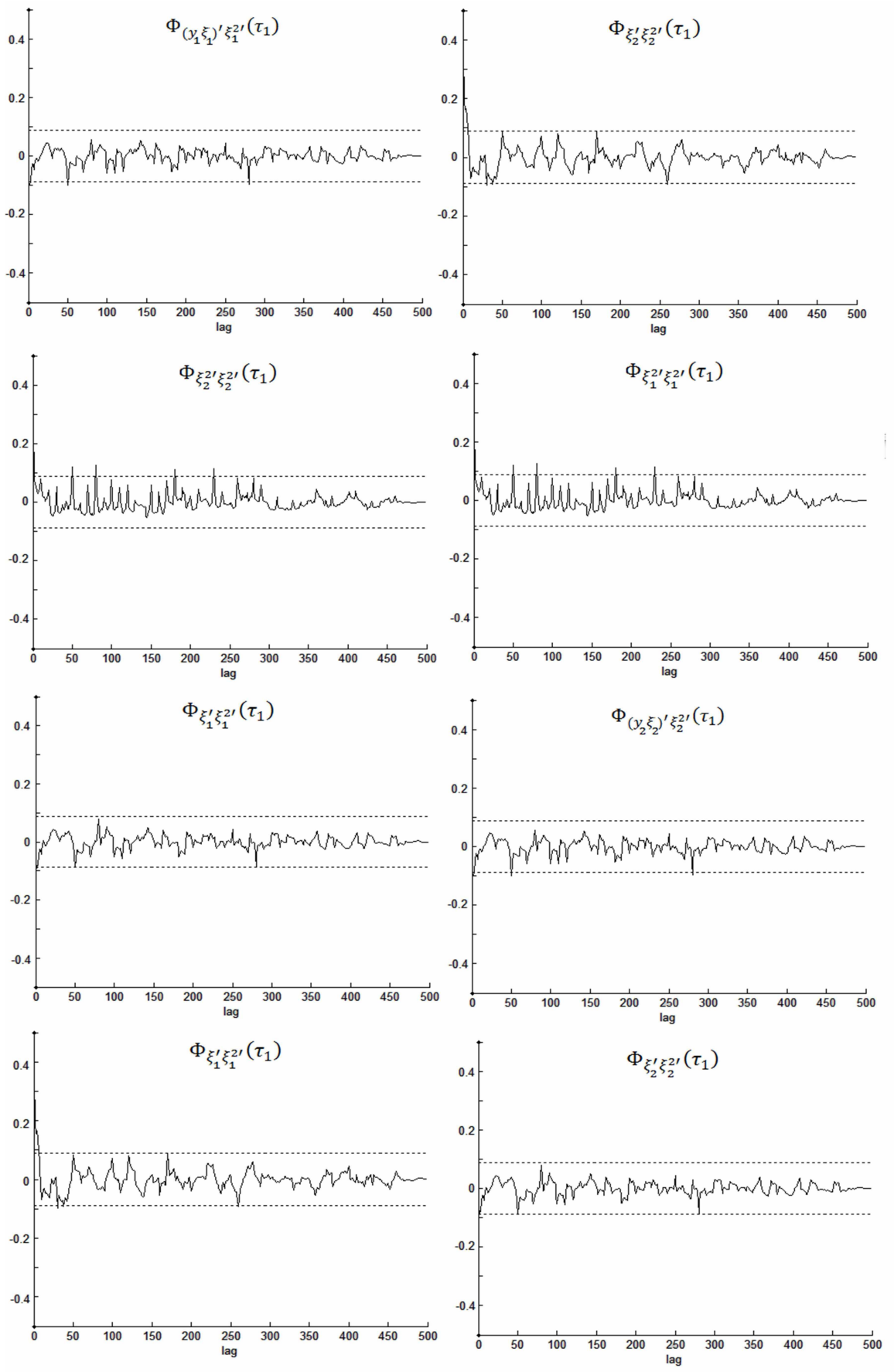


Figura 4.12 – Testes estocásticos do modelo obtido.

4.7 - COMENTÁRIOS FINAIS

Neste capitulo sistemas lineares e não lineares foram testados tendo estes estruturas polinomiais ou racionais e podendo ser os sistemas do tipo MIMO, SISO ou MISO e a metodologia proposta apresentou resultados satisfatórios. Mesmo para casos que apresentam certas dificuldades que foram aqui discutidas.

Quando necessário, testes estocásticos e dinâmicos foram aplicados aos modelos obtidos e nenhum dos testes foram insatisfatórios.

A metodologia utilizando o DE para gerar regressores, apresenta muito potêncial que ainda deve ser estudado e como consequência uma nova estrutura com expoentes reais também pode ser explorada neste capitulo, mas ainda existe muito para ser estudado com relação a estes dois aspectos.

CAPÍTULO 5

RESULTADOS DE SIMULAÇÕES

5.1 - CONSIDERAÇÕES INICIAIS

Neste capítulo aplicou-se a metodologia proposta, em um Reator de polímero e um conversor CC-CC do tipo BOOST QUADRÁTICO-BUCK e este por sua vez foi escolhido, porque necessita de controle mais complexo do que os conversores CC-CC que são somente elevadores ou abaixadores de tensão. Os dados do reator de polímero foram retirados de (FRANCIS J. DOYLE III, 1995) e os cálculos do conversor foram retirados de (SILVA, et al., 2011), sendo os dados gerados a partir de simulação no software PSIM® 9.0.3.

5.2 - CONVERSOR BOOST QUADRÁTICO-BUCK

O conversor avaliado possui uma única chave ativa e, dependendo do valor da razão cíclica, pode operar como elevador ou abaixador de tensão. Ele apresenta uma larga faixa de conversão de tensão, o que permite que opere desde 12 VCC até 180 VCC. No entanto, por apresentar seis polos e a necessidade de operar em uma larga faixa de conversão de tensão, o controle é mais complexo se comparado com os conversores CC-CC elevadores ou abaixadores tradicionais.

A Figura 5.1 mostra o diagrama esquemático do conversor Boost2-Buck (SILVA, et al., 2011)

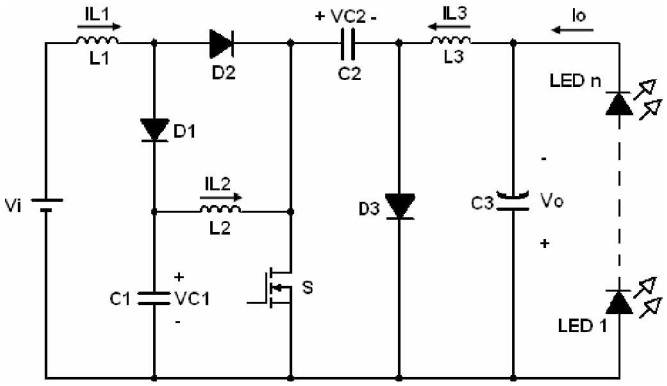


Figura 5.1 – Esquema do conversor Boost2-Buck.

Com o objetivo de validar a metodologia proposta, o conversor foi simulado no software PSIM® 9.0.3.

A Tabela 5.1 apresenta os valores dos parâmetros utilizados na simulação do mesmo.

Tabela 5.1: Parâmetros de simulação do conversor Boost2-Buck.

Parâmetros	Valores
I_0	40 mA
V_0	160 V
V_i	12 – 180 V
Chave S	*Ideal*
Frequência de chaveamento	*100 kHz*
Indutor L_1	*1,03 mHz*
Indutor L_2	*18,6 mHz*
Indutor L_3	*29,2 mHz*
Capacitor C_1	*100 nF*
Capacitor C_2	*9,4 nF*
Capacitor C_3	*100 uF*
Diodos D_1, D_2 *e* D_3	*(Um resistor de 4kΩ em série com um diodo ideal)*

O valor da tensão de entrada e da razão cíclica do conversor, foram variadas de forma aleatória, utilizando-se o recurso do programa PSIM denominado "CBloco".

Porém, para se evitar que a combinação entre razão cíclica e tensão de entrada gerem um valor de corrente de saída acima do valor especificado no projeto, as seguintes providências foram tomadas: (i) para um determinado valor aleatório de razão cíclica, o valor da tensão de entrada é limitado de acordo com a Equação (5.1)

sendo esta a equação do ganho estático deste conversor; (ii) Esta equação supra citada, minimiza o problema, porém, não garante que a corrente de saída não ultrapasse o valor máximo, por isso, uma outra sub-rotina foi criada para monitorar o valor da corrente de saída e, caso ela atinja o valor máximo, a rotina irá calcular um valor aleatório de corrente e zerar a razão cíclica, até que o valor da corrente se iguale ao valor de corrente calculado aleatoriamente. A partir deste ponto, os valores de tensão e de corrente voltarão a ser aleatórios até atingem novamente alguma das condições limites especificadas.

$$V_{imax} = \frac{(1-D).R.I_o}{D^2} \tag{5.1}$$

O bloco "CBloco" do PSim é executado a cada 0,1μs (micro segundos), mas a cada 10000 vezes que este bloco é executado ou seja a cada 1 ms, são coletados uma amostra da tensão de entrada (V_{in}), uma da razão cíclica (Duty) e uma da corrente de saída do LED (I_{Led}). O programa para quando 20s de simulação são atingidos, o que resulta num total de 2000 pontos, para cada variável.

A Equação (5.2) mostra o modelo encontrado para este conversor. É importante ressaltar que os dados de entrada e de saída, usados nesta identificação, foram escalonados para uma faixa de valores entre 0.1 e 1.0. O processo de escalonar os dados foi efetuado para que a estrutura com expoentes reais possa ser melhor explorada. Estruturas que antes seriam eliminadas por levar a indeterminações, por exemplo, uma divisão por zero ou uma raiz com radical par e um radicando com sinal negativo, passam a ser consideradas no processo de cálculo uma vez que os dados não estão mais nesta faixa problemática.

$$I_{Led}[k] = \frac{\text{Numerador}}{\text{Denominador}} \tag{5.2}$$

Para a obtenção deste modelo a variável de saída "I_{Led}" que varia de 0 a 0.1 e as variáveis de entrada "V_{in}" que varia de 0 a 180, assim como "Duty" que varia de 0 a 1 foram escalonadas para a faixa de 0.1 a 1.0.

O histograma da Figura 5.2 mostra que os resíduos do modelo encontrado para o conversor Boost2-Buck também se parecem com uma gaussiana com maior concentração no zero do eixo coordenado.

FALTA uma FIGURA aqui

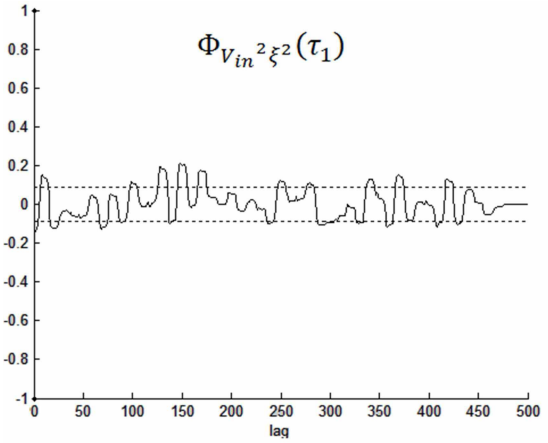
Figura 5.2 – Histograma dos resíduos do modelo do conversor Boost2-Buck.

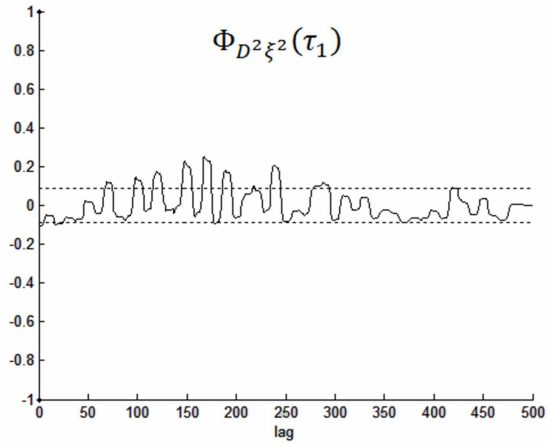
A Figura 5.3 apresenta as formas de ondas da corrente de saída do conversor Boost^2-Buck. A forma de onda azul representa a corrente medida e a forma de onda vermelha a estimada por meio do modelo obtido com o algoritmo implementado.

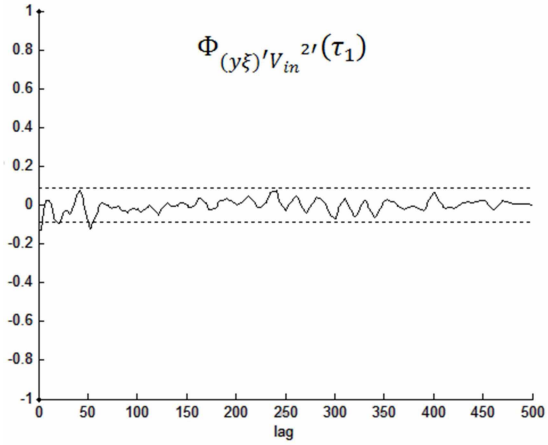
FALTA uma FIGURA aqui

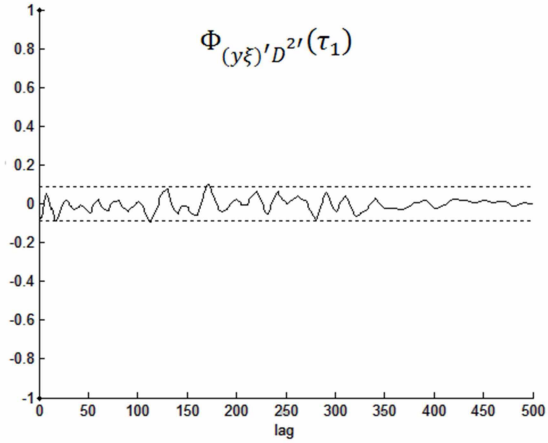
Figura 5.3 – Formas de onda da corrente de saída do medida (azul) e estimada (vermelha).

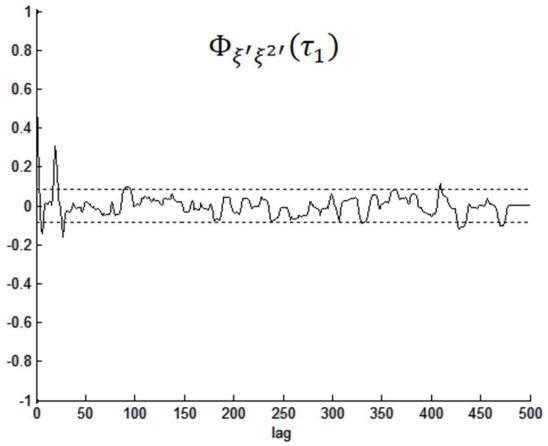
A Figura 5.4 apresenta as formas de ondas dos testes estocásticos feitos para a validação do modelo. É possível identificar que em alguns testes de correlação das variáveis de entrada e de saída com o ruído, as curvas saíram do limite de confiança e isto é uma sinalização que para este conversor existe a necessidade de regressores com termos do ruído e este sistema implementando não contemplou esta possibilidade ficando isto como trabalho futuro.

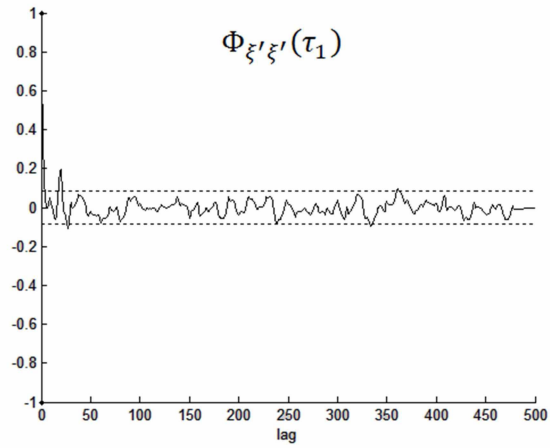

$\Phi_{V_{in}{}^2\xi^2}(\tau_1)$
lag

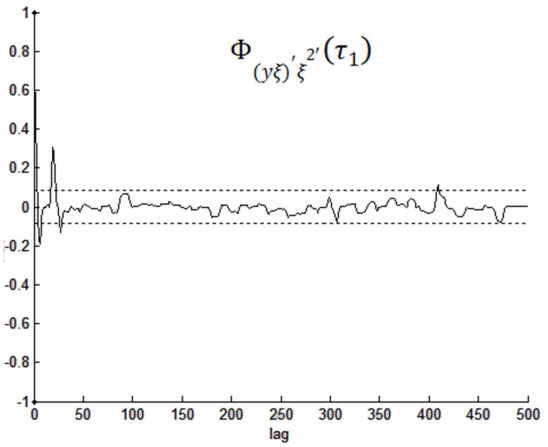

$\Phi_{D^2\xi^2}(\tau_1)$
lag

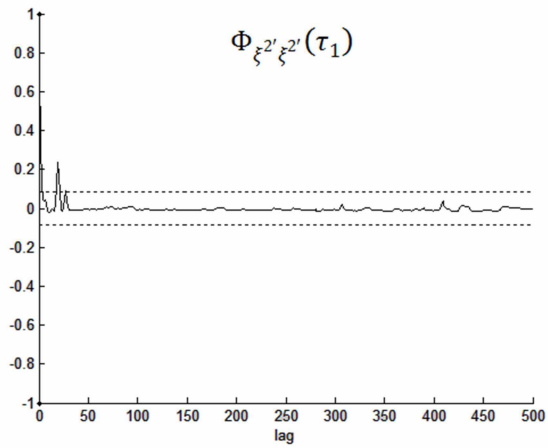

$\Phi_{(y\xi)'V_{in}{}^{2'}}(\tau_1)$
lag


$\Phi_{(y\xi)'D^{2'}}(\tau_1)$
lag


$\Phi_{\xi'\xi^{2'}}(\tau_1)$
lag


$\Phi_{\xi'\xi'}(\tau_1)$
lag


$\Phi_{(y\xi)'\xi^{2'}}(\tau_1)$
lag


$\Phi_{\xi^{2'}\xi^{2'}}(\tau_1)$
lag

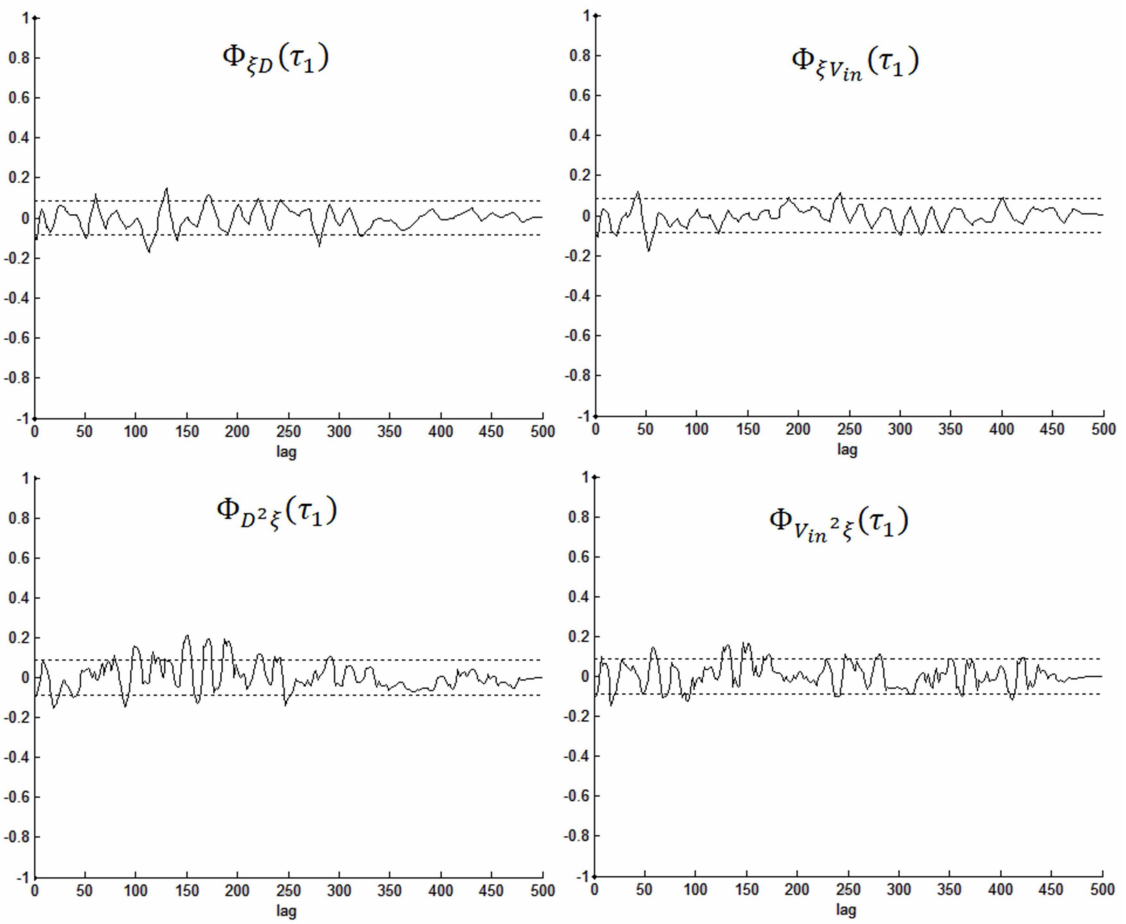


Figura 5.4 – Formas de onda da corrente de saída do medida (azul) e estimada (vermelha).

5.3 - REATOR BIOQUIMICO

Segundo Henson et al. (2001), reatores bioquímicos ou biorreatores são sistemas dinâmicos altamente complexos. Os modelos matemáticos que representam tais sistemas não são de fácil tratamento sistemático, resultado direto da alta complexidade associada aos processos bioquímicos. Contudo, modelos simplificados podem ser desenvolvidos negligenciando processos intracelulares, heterogeneidade da população celular, detalhamento de caminhos metabólicos e a influência de substratos intermediários e substâncias interferentes presentes nos meios intracelular e extracelular.

Dessa forma, modelos matemáticos que antes seriam compostos por sistemas constituídos por um grande número de equações diferenciais parciais e equações diferenciais ordinárias são constituídas por um número pequeno de equações diferenciais ordinárias ou equações de diferença.

Neste trabalho os dados retirados de (FRANCIS J. DOYLE III, 1995) foram aplicados no sistema proposto. O modelo obtido está apresentado na Equação (5.3).

$$y(k) = \frac{1{,}62449.\,y(k-1)^{0.847161} + 0{,}682539.\,(y(k-3)^{-0.00185515}).\,(y(k-2)^{0.8173})}{1 + 0{,}285459.\,u(k-1)} + e(k) \qquad (5.3)$$

Para a obtenção deste modelo a variável de saída “y” que varia de 19556.2 a 55591 foi escalonada para a faixa de 0.1 a 1.0 assim como a variável de entrada “u” que varia de 0 a 180.

Este modelo obtido apresentou um BIC de -1.85036e+06 e um RMSE(2) de 107.509. A Figura 5.5 apresenta os resultados obtidos deste processo de identificação.

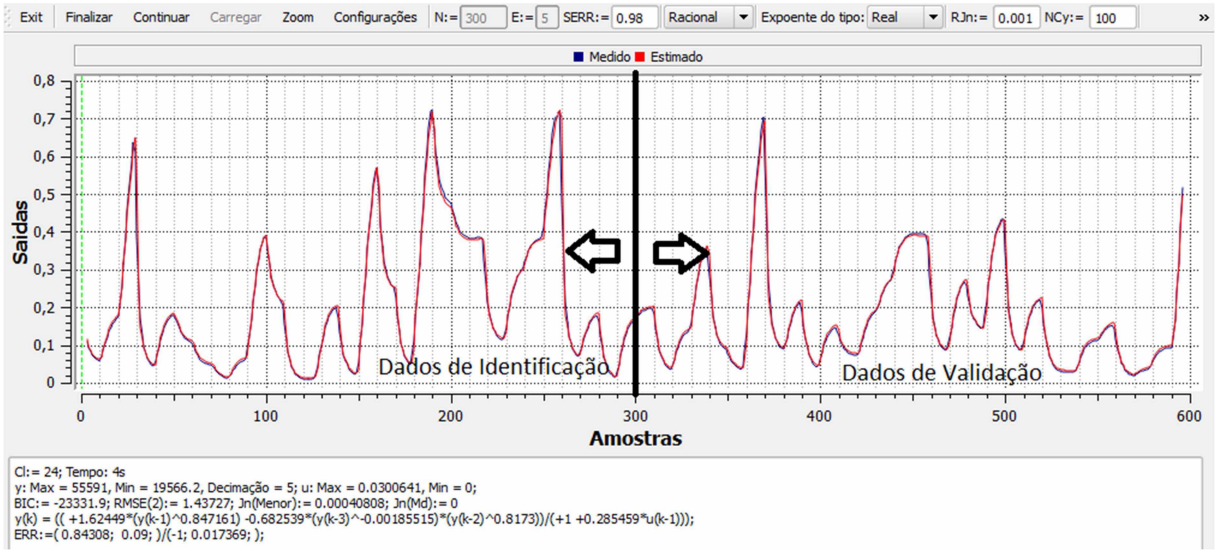


Figura 5.5 – Formas de onda do valor de saída medida (azul) e estimada (vermelha) do reator bioquímico.

A validação dinâmica deste modelo é confirmada pelo histograma dos resíduos que esta apresentado na Figura 5.6. Dessa forma, na Figura 5.7 os resultados

estatísticos são apresentados e nenhum teste ultrapassou a região de confiança pré-estabelecida o que confirma a validação do modelo com base nestes critérios.

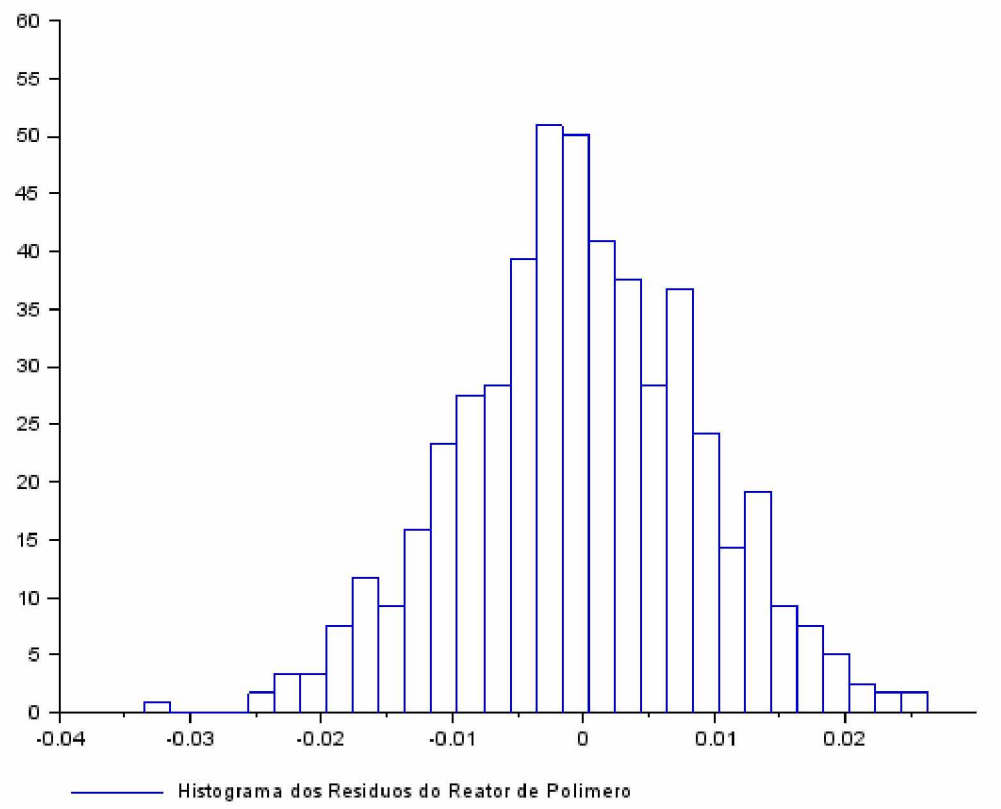


Figura 5.6 – Histograma dos resíduos obtidos na identificação do reator bioquímico.

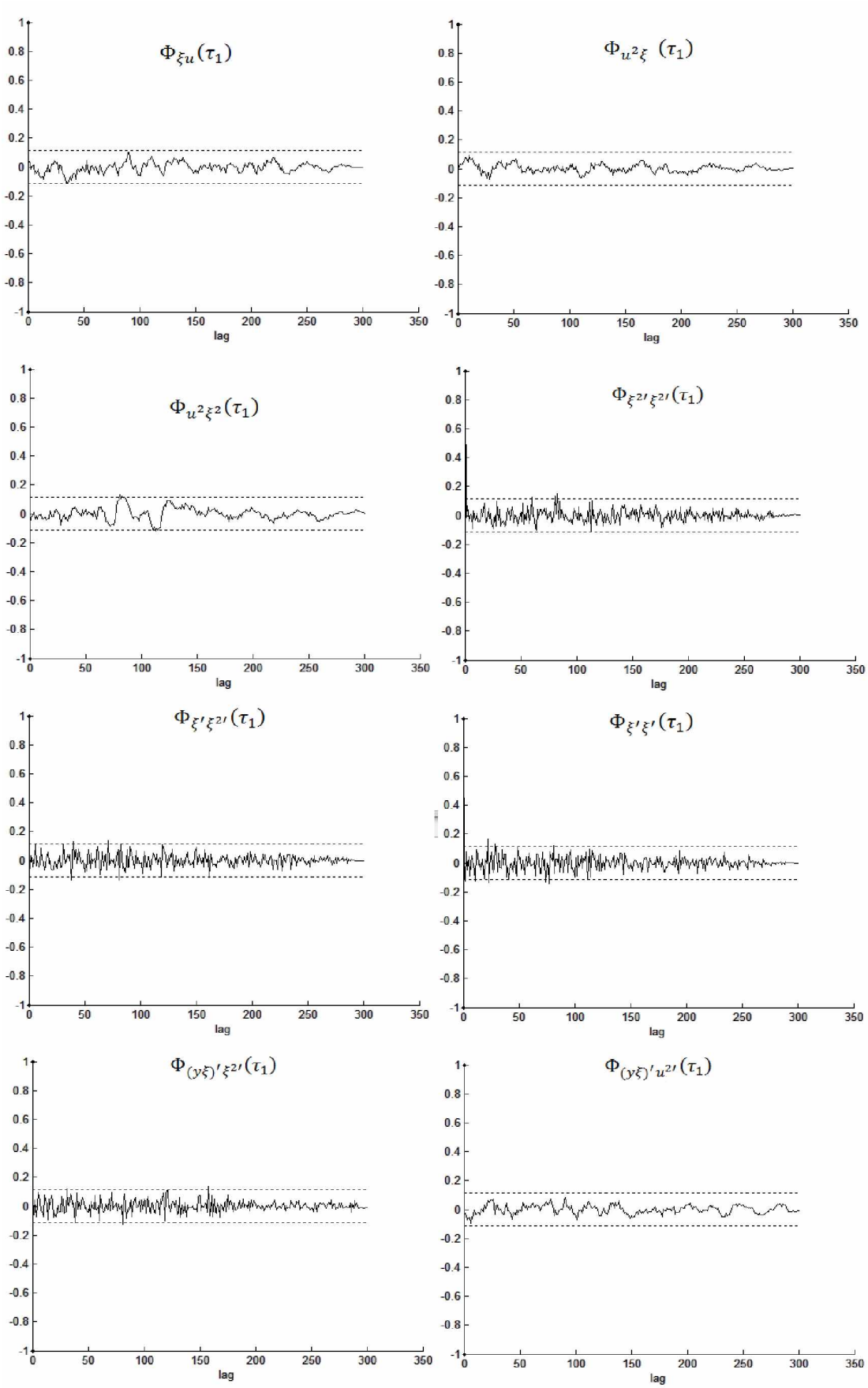


Figura 5.7 – Curvas de correlação para validação estatística do modelo obtido do reator bioquímico .

5.4 - CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os modelos matemáticos do conversor BOOST QUADRÁTICO-BUCK e de um reator bioquímico de polímero, obtidos pela metodologia proposta neste trabalho.

Os modelos encontrados atenderam todos os testes de validação efetuados. Sendo que para o conversor BOOST QUADRÁTICO-BUCK o modelo obtido é responsável por representar a dinâmica da corrente de saída para uma faixa de tensão de entrada de 12 à 180V e razão cíclica de 0 a 100%. Ou seja, foi capaz de identificar um sistema MISO com uma larga fixa de variação em uma das entradas.

Ele também apresentou uma solução com apenas 4 (quatro) regressores para representar o reator de polímero que através dos testes efetuados apresentaram excelentes resultados.

É importante ressaltar que para ambos os modelos os dados foram escalonados para uma faixa de 0.1 à 1.0 para que fosse melhor avaliado a estrutura com expoentes reais.

CAPÍTULO 6

CONCLUSÃO GERAL

Este trabalho apresentou três contribuições de destaque. Uma nova técnica de identificação de sistema utilizando o Algoritmo Diferencial Evolutivo, uma nova estrutura, do tipo NARMAX com expoentes reais e um novo operador mutação que possibilita aplicar o DE em problemas de combinação e otimização ao mesmo tempo.

Na nova técnica de identificação, algoritmos estocásticos foram aplicados para auxiliar em algumas falhas que alguns algoritmos determinísticos apresentam (ERR selecionando regressores espúrios) e ou para melhorar a capacidade de convergência (domínio infinito de regressores que são testados).

A estrutura com expoente real, apresentada neste trabalho, tem como inconveniente que algumas combinações de expoentes e com certas bases podem gerar números imaginários ou indeterminações tais como divisão por zero. Uma solução para resolver este problema foi apresentada neste trabalho, é o processo de escalonar os dados, ou seja, mudar o domínio dos dados de entrada e de saída e assim garantir que estes dados não irão comtemplar valores proibitivos.

Em contra partida as estruturas com expoente real apresentam como vantagem a capacidade de representar em um único termo com expoente real a combinação de vários outros termos com expoentes inteiros. Ou seja, apresenta capacidade de síntese e consequente redução do número de termos. Mas isto só é observado se o domínio dos dados avaliados não leva a números imaginários ou indeterminações, caso contrário, toda esta abrangência é perdida. Estas estruturas com característica desejáveis acabam sendo eliminadas por causa de um ou mais pontos indeterminados. Todas estas considerações foram observadas nos testes elaborados neste trabalho.

Além disso, também foi proposta uma estratégia de Gerenciamento de Tarefas para a Execução de Tarefas em Modo Contínuo, que fornece condições para um bom aproveitamento dos recursos de hardware disponível. Considerando que o processo de identificação de sistemas exige muito esforço computacional e, consequentemente, muito tempo, a estratégia proposta pode ser empregada para aperfeiçoar outros algoritmos de identificação de sistemas em uso na atualidade e é apresentada no Anexo A.

Os histogramas apresentados mostram que os modelos matemáticos encontrados possuem boa representação da dinâmica dos respectivos sistemas, pois a forma de onda do histograma dos resíduos estão dentro dos níveis desejados.

A validação estática foi efetuada e confirmou a boa representatividade dos modelos obtidos e consequentemente a boa capacidade da metodologia proposta em selecionar a estrutura num processo de identificação que foi o objeto de estudo deste trabalho.

Como um balanço final do presente trabalho pode se afirmar que o mesmo apresenta significativas contribuições à área de identificação de sistemas. Foram exploradas vertentes inéditas cujos resultados são entusiasmantes, não somente pelos excelentes resultados obtidos, mas também pela expectativa em explorar novas possibilidades.

REFERÊNCIAS BIBLIOGRÁFICAS

ADYA, M., COLLOPY, F., ARMSTRONG, J. S., & KENNEDY, M. (2001). Automatic Identification of Times Series Features for Rule-Based Forecasting. *International Journal of Forecasting., 17*(2), 143-157.

Aguirre, L. A. (1997). Nonlinear identification and cluster analysis of chaotic attractors from a real implementation of Chua's circuit. *Int. J. Bif. Chaos,, 7*(6), 1411-1423.

Aguirre, L. A. (2007). *Introdução à Identificação de Sistemas 3º Edição.* Belo Horizonte: Editora UFMG.

ALVES, M. A. (2007). "Influência do Compartilhamento de Cache L2 em um Chip Multiprocessado sob Cargas de Trabalho com Conjuntos de Dados Contíguos e Não Contíguos",. *VIII Workshop em Sistemas Computacionais de Alto Desempenho (WSCAD).*

Alves, M. A. (17 de Dezembro de 2009). *Utilização de Auto-Consistência como Ferramenta Auxiliar na Seleção de Estrutura de Modelos Narx Polinomiais.* Belo Horizonte.

Aquarius, S. (2012). Aplicação de Sensor Virtual em Processo de Saneamento. *InTech*(135).

ÅSTRÖM, K., & WITTENMARK, B. (1995). *Adaptive Control.* New York: Addison-Wesley.

Barbosa, B. H. (13 de Outubro de 2009). *Computação Evolucionária e Máquinas de Comitê na Identificação de Sistemas Não-Lineares.* Belo Horizonte.

BENINI, L. M. (2005). Network-on-chip architectures and design methods. *IEEE Proceedings Computers & Digital Techniques, 152*(2), 261-272.

Billings, K. Z. (1997). Algorithms for minimal model structure detection in nonlinear dynamic system identification. *International Journal of Control*, 311-330.

Billings, S. A. (1980). Identification of nonlinear systems - a survey. *In Proc. IEE*, 272-285.

Billings, S. A. (1989a). The identification of linear and non-linear models of a turbocharged automotive diesel engine. *Mechanical System and Signal Processing, 3*(2).

Billings, S. A. (1991). Rational model identification using an extended least-squares algorithm. *Int. J. Control, 54*(3), 529-546.

Bortolossi, H. J. (2002). *Cálculo Diferencial a Várias Variáveis - Uma Introdução À Teoria de Otimização* (1° Edição ed.). PUC-rio.

BOSSLEY, K. M. (1997). Neurofuzzy Modelling Approaches in System Identification. *Tese, University of Southampton*.

Braess, D. (1986). Nonlinear Approximation Theory. *Spinger-Verlag*.

CALIXTO, W. P., ALVARENGA, B. P., MOTA, J. C., BRITO, L. C., & WU, M. (2010). Electromagnetic Problems Solving by Conformal Mapping- A Mathematical Operator for Optimization. *Mathematical Problems in Engineering*, 63, 64, 65, 68.

Celso Camilo, K. Y. (2007). NOVO OPERADOR GENETICO EM UM FLUXO PARALELO DE EXECUÇÃO PARA MELHORAR O DESEMPENHO DO AG. *SBA*.

Chakraborty, U. K. (2008). *Advances in Differential Evolution.* New York: Springer.

Coelho, A. A., & Coelho, L. S. (1999). ALGORITMOS EVOLUTIVOS EM IDENTIFICAÇÃO E CONTROLE DE PROCESSOS:UMA VISÃO INTEGRADA E PERSPECTIVAS. *SBA Controle & Automação, 10*(01).

COELHO, A., & COELHO, L. (2003). *Identificação de Sistemas Dinâmicos Lineares.* Florianópolis, SC: Editora da UFSC.

Corrêa, M. V. (05 de 12 de 1997). *Identificação de Sistemas Dinâmicos Não-Lineares Utilizando Modelos NARMAX Racionais - Aplicação a Sistemas Reais.* UFMG. Belo Horizonte: Dissertação de Mestrado.

Cunha, M. J. (2011). *Descoberta do Conhecimento em Base de Dados como Ferramenta Aplicada em Processos Sucroalcooleiros.* Tese (Doutorado em Engenharia Mecânica São Carlos), Universidade de São Paulo, São Carlos.

Dasrupta, S., Das, S., Biswas, A., & Abraham, A. (2009). On Stability and Convergence of the Population-Dynamics in Differential Evolution. *AI Communications, 22*.

Emiliano, P. C., Veiga, E. P., Vivanco, M. J., & Menezes, F. S. (2010). CRITÉRIOS DE INFORMAÇÃO DE AKAIKE VERSUS BAYESIANO: ANÁLISE COMPARATIVA. *19º SINAPE.*

Faria, K. S. (2005). *Algoritimo Evolutivo Aplicado à identificação de Sistemas.* São Paulo: Universidade Presbiteriana Mackenzie.

FIESP/CIESP. (2011). *AMPLIAÇÃO DA OFERTA DE ENERGIA.* São Paulo: FIESP/CIESP.

Foster, D. C. (2009). *Arquiteturas Multicore.* Universidade Federal de Santa Maria, Santa Maria – RS – Brasil.

FRANCIS J. DOYLE III, B. A. (1995). Nonlinear Model-based Control Using Second-order Volterra Models. *Automática, 31*, 697-714,.

FREITAS, H. C. (2006). *Chip Multithreading: Conceitos, Arquiteturas e Tendências.* PPGC/UFRGS.

Gutierrez, J. (2009). Técnicas Avançadas para Controle Industrial. *MECATRÔNICA ATUAL*, 1-4.

IWASE, M., IIKUBO, H., HATAKEYAMA, S., & FURUTA, K. (2002). An Identification Method for Continuous-Time Transfer Functions Based on Nonlinear Optmization. *Proceedings of the IEEE Conference of the 28th Annual Industrial Electronics Society, 3*, pp. 1978-1983.

Jang, H. K. (1994). Identification of loudspeaker nonlinearities using the NARMAX techinique. *J. Audio Eng. Soc., 42*(1/2), 50-59.

Jorgensen, S. B. (1995). Grey box modelling for control: Qualitative models as a unifying framework. *Int. J. of Adaptative Control and Signal Processing,, 9*(6), 547-562.

LACERDA, A. S. (Junho de 2010). *PROPOSTA DE UM ALGORITMO EVOLUCIONÁRIO NEBULOSO.* Dissertação, Belo Horizonte.

Leontaritis, I. a. (1985a). Input-output parametric models for non-linear systems part I : deterministic non-linear systems. *Int. J. Control, 41*(2), 303-328.

Linden, R. (2008). *Algoritmos Genéticos - Uma importante ferramenta da Inteligência Computacional.* Rio de Janeiro: Brasport.

LJUNG, L. (1996). System Identification. *W.S. The Control Handbook, CRC Press & IEEE Press*, 1033-1054.

LJUNG, L. (1999). *System Identification- Theory for the User.* Englewood Cliffs: Prentice Hall.

ŁUKASZ MACIEJEWSKI, W. M. (14 de Fevereiro de 2007). APPLICATION OF DIFFERENTIAL EVOLUTION ALGORITHM FOR IDENTIFICATION OF EXPERIMENTAL DATA. *Editorial Board.*

M. Thoma, F. A. (2010). *Lecture Notes in Control and Information Sciences.*

Madsen, K., Nielsen, H., & Tingleff., O. (Abril de 2004). Methods for non-linear least squares problems. *Technical University of Denmark*.

MANER, B., DOYLE III, F., & OGUNNAIKE, B. e. (1994). A Nonlinear Model Predictive Control Scheme Using Second Order Volterra Models. *Proc. American Control Conference*, 3253-3257.

MARCHI, P. (1999). *Abordagem Não-Linear para Modelagem e Controle Preditivo: Estudo de* . Dissertação de Mestrado, PGEEL / UFSC, Florianópolis.

Maria Isabel Rodrigues, A. F. (2005). *Planejamento de Experimentos e Otimização de Processos.* Cárita Editora.

Matsumoto, M., & Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 3–30.

MIYADAIRA, A. N. (2009). *Microcontroladores PIC18: aprenda e programe emlinguagem C,* (1ª ed ed.). Érica.

Nascimento, T. C., & Mothé, C. G. (2007). Gerenciamento de resíduos sólidos industriais. *Revista Analytica*(27), 37-45.

Nikolas Aschermann, P. R. (2007). *Arquitetura Multicore.* Instituto de Computação Universidade Estadual de Campinas, Campinas.

Nocedal, J., & Wright, S. J. (2006). *Numerical Optimization* (2nd Edição ed.). Springer.

Noshiro, M. M. (1993). Nonlinear identification of the PCO2 control system in man. *Computer Methods and Programs in Biomedicine, 40*, 189-202.

PAIVA, R. P. (1999). Identificação Neuro-Difusa - Aspectos de Interoperabilidade. *Dissertação, Universidade de Coimbra.*

PEARSON, R. (2003). Selecting Nonlinear Model Structures for Computer Control. *Journal of Process Control, 13*, 1-26.

Price K.V., S. R. (2005). Differential evolution: a practical approach to global optimization. 538p.

Pröll, T. a. (1994). Model-predictive pH control using real-time NARMX approach. *AIChE Journal, 40*(2), 269-282.

Qt.Nokia, 2012. (s.d.). Acesso em 09 de Junho de 2012, disponível em Site da Qt.Nokia: http://qt.nokia.com/qt-in-use

RODRIGUES, G. (2000). Identificação de Sistemas Dinâmicos Não-Lineares Utilizando Modelos NARMAX Polinomiais - Aplicação a Sistemas Reais. *Dissertação, Universidade Federal de Minas Gerais*.

Rodríguez-Vázquez, K., Fonseca, C. M., & Fleming, P. J. (2004). Identifying the Structure of NonLinear Dynamic Systems Using Multiobjective Genetic Programming. *IEEETRANSACTIONSONSYSTEMS, 34*(4), 531.

Russel, S., & Norvig, P. (2004). *Inteligência Artificial* (2° ed.). Rio de Janeiro: Elsevier.

Samohyl, R. W. (2005). *Introdução à Estatística e Métodos de Previsão em Séries Temporais: Teoria Aprofundada e Prática Simplificada.* UFSC, Florianopolis.

Santos, J. E. (08 de 2003). *CONTROLE PREDITIVO NÃO-LINEAR PARA SISTEMAS DE HAMMERSTEIN.* UFSC. Florianópolis: UFSC.

SANTOS, S. C. (2000). Identificação Multivariável Aplicada aos Processos Químicos: Estudo de Caso. *Dissertação, Universidade Federal de São Carlos*.

Sarti, F., & Hiratuka, C. (2011). Desenvolvimento industrial no Brasil: oportunidades e desafios futuros. *IE/UNICAMP, 187*, 2-3.

SILVA, W. G., RODOVALHO, D. S., VICENZI, F. R., MORAIS, J. S., Freitas, L. C., COELHO, E. A., et al. (2011). Conversor Boost Quadrático-Buck Utilizado No Acionamento De Um Vetor De Led Empregado No Tratamento De Disfunções Celulares. *Eletrônica de Potência (Impresso), 16*, pp. 204-211.

SIMANI, S., FANTUZZI, C., & BEGHELLI, S. (setembro de 2000). Diagnosis Techniques for Sensor Faults of Industrial Processes. *IEEE Transactions on Control Systems Technology, 8*, pp. 848-855.

SJOBERG, J., HJALMARSSON, H., & LJUNG, L. (1994). Neural networks in system identification. *Proceedings of the 10th IFAC Symposium on System Identification*, pp. 49-72.

SOUZA, D. J., & LAVINIA, N. C. (2009). *Conectando o PIC – Recursos Avançados* (1° ed ed.). São Paulo: Érica.

Storn, R., & Price, K. (1995). Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. *ICSI*.

Thiago W. M. Abreu, J. A. (Janeiro de 2010). ESTRATÉGIAS DE IDENTIFICAÇÃO PARAMÉTRICA APLICADAS À MODELAGEM DINÂMICA DE UM SERVIDOR WEB APACHE. *Controle & Automação, 23*.

Vallverdu, M. K. (1992). Model indentification of the neural control of the cardiovascular system using NARMAX models.

Wei, H. a. (2008). Model structure selection using an integrated forward orthogonal search algorithm assisted by squared correlation and mutual information. *International Journal of Modelling Identification and Control, 3 (4)*, 341-356.

Yanjun Liu, F. D. (29 de Julho de 2011). A new parameter estimation algorithm for non-uniformly multirate sampled-data systems. *American Control Conference*.

Zadeh L. A. (1973). "Outline of a new approach to the analysis of complex systems and decision processes". *3*(1), 28-44.

Zhu, Q. M. (1993). Parameter estimation for stochastic nonlinear rational models. *Int. J. Control, 57*(2), 309-333.

ZORLU, S. O. (Junho de 2011). Identification of bilinear systems using differential evolution algorithm. *Indian Academy of Sciences, 36*(3), 281-292.

APÊNDICE A

DESCRITIVO DO SOFTWARE

Este capítulo foi descrito com o intuito de ilustrar as funções implementadas no aplicativo desenvolvido nesta tese. A ideia consiste em dar uma visão panorâmica do mesmo.

A.1 - CARREGANDO UM PACOTE DE DADOS:

O primeiro passo a ser seguido após abrir o programa é carregar um arquivo contendo os dados de entradas e saídas que serão utilizadas. Desta forma, basta clicar no ícone "Carregar" como ilustrado na Figura A.1.

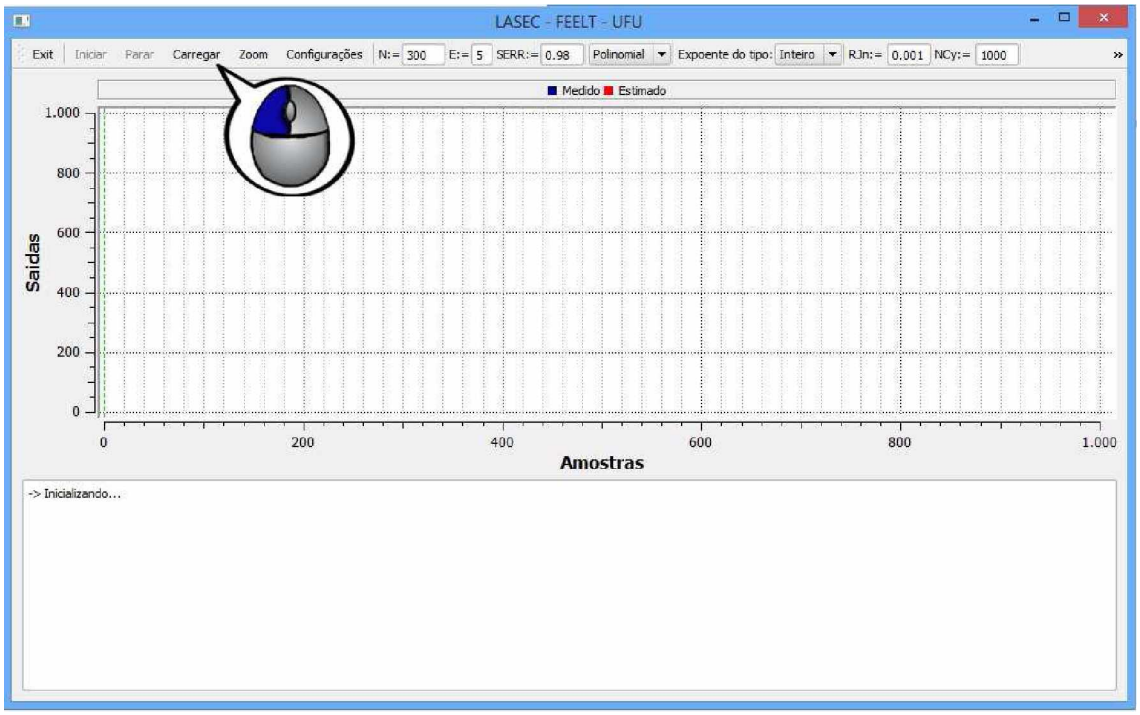


Figura A.1 – Função "Carregar"

Após a realização do primeiro passo, basta clicar nas reticências encontrada na caixa de diálogo que fora aberta, selecionar o arquivo desejado na janela do Explorer e

clicar em “Abrir”, assim como mostrado na Figura A.2. É importante frisar que o arquivo deve estar em formato “*.txt”.

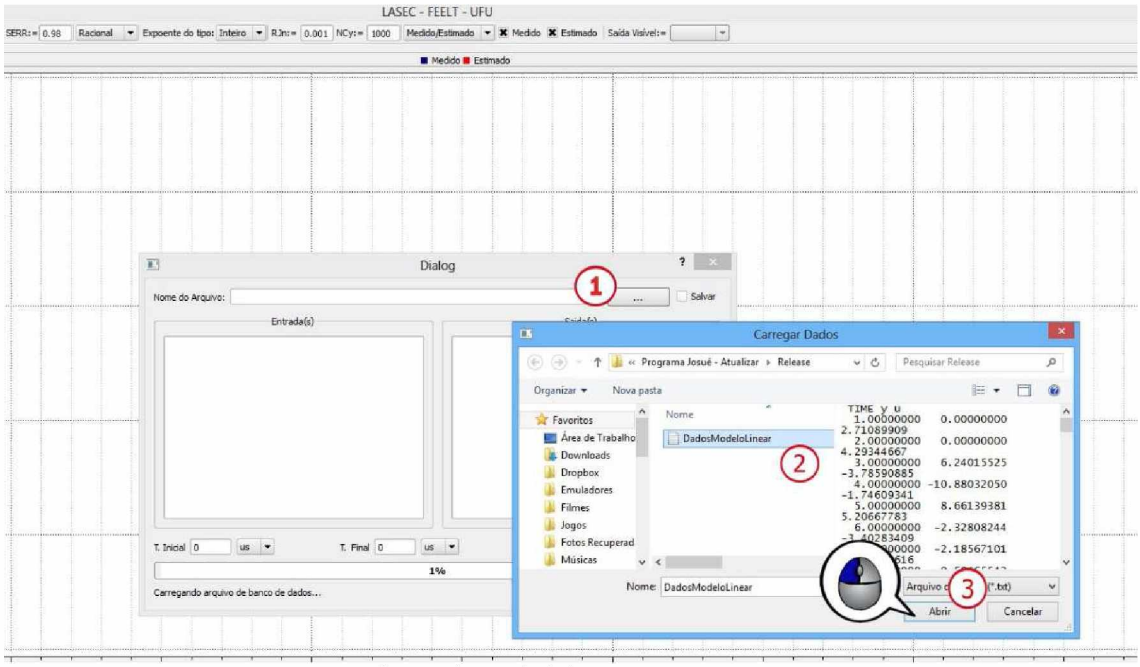


Figura A.2 – Adicionando um dado

As variáveis contidas no arquivo serão importadas para o software após selecionadas as entradas e saídas desejadas. Pode-se ainda selecionar o ponto de início da leitura dos dados e ponto de parada de leitura preenchendo os campos “tempo inicial” e “tempo final” respectivamente. Para iniciar o processo de leitura dos dados basta clicar no botão “Carregar”.

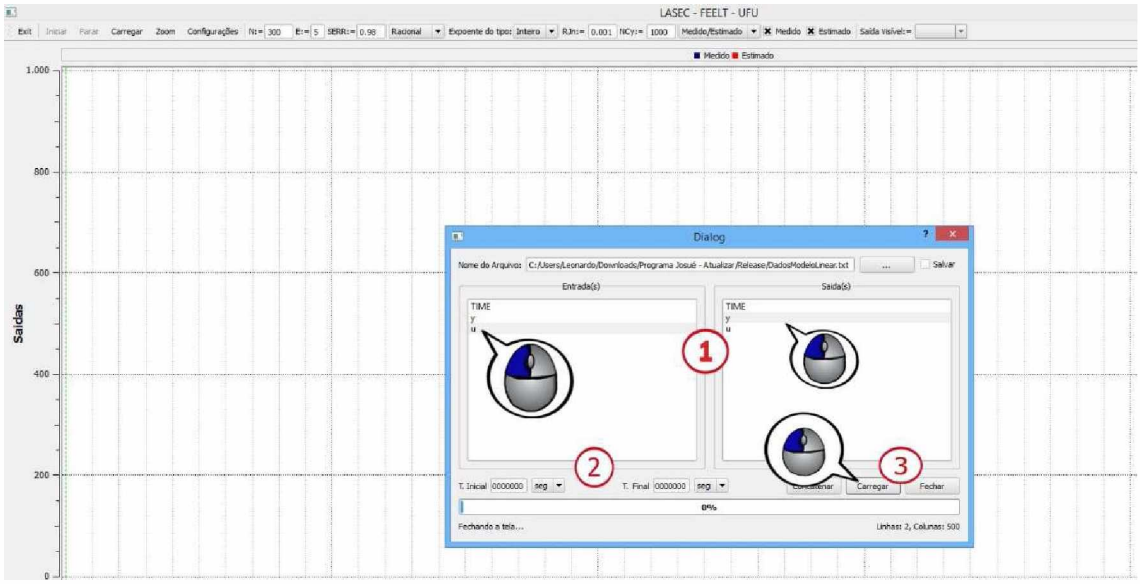


Figura A.3 – Determinando entradas e saídas

Será aberta uma nova janela a qual contém algumas informações sobre as variáveis. Esta janela é para o caso em que os dados podem ser escalonados para uma faixa de [0,1] e isto é feito preenchendo os valores de máximo e mínimo da curva que serão convertidos para esta faixa supracitada. O software calcula faixa de decimação para as saídas e é nesta tela que este valor é apresentado e o usuário escolhe um valor. Estes valores já veem pré-definidos pelo software, no entanto é passível de modificações.

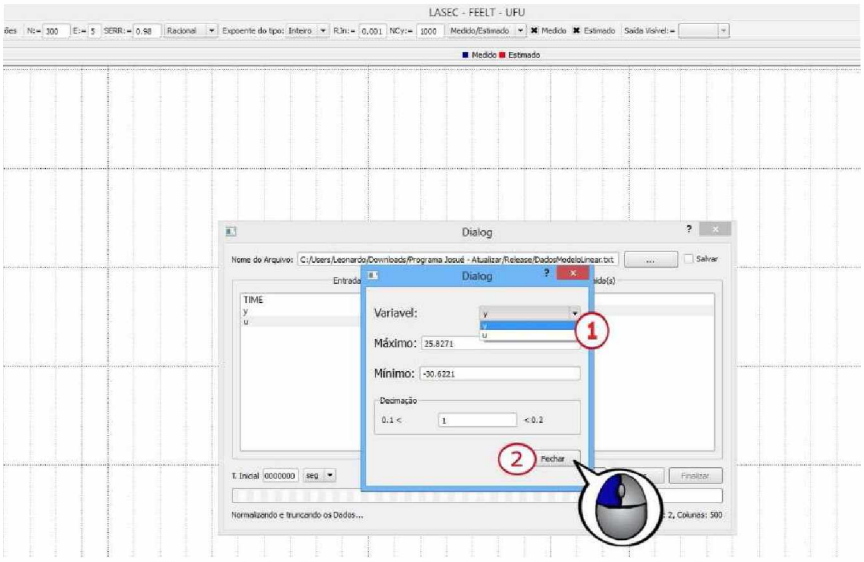


Figura A.4 – Propriedades da variável de entrada

Caso não seja necessária nenhuma alteração, basta clicar na aba “Fechar” duas vezes, na janela de propriedades da variável e na de carregamento de dados.

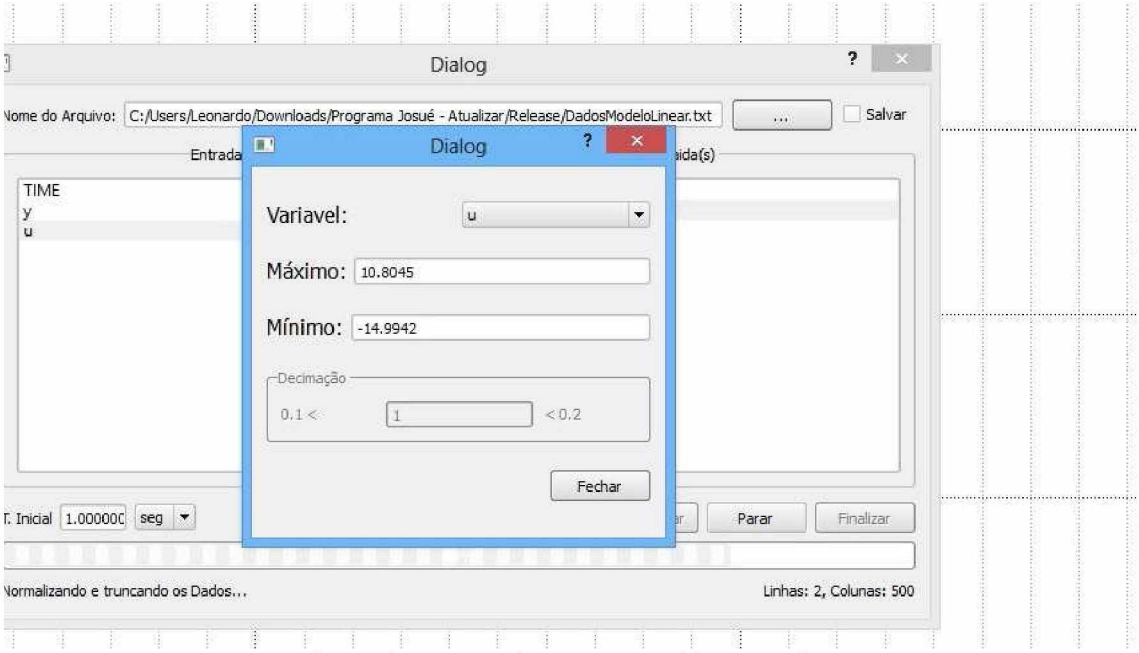


Figura A.5 – Propriedades da variável de saída

A.2 - INICANDO OS CÁLCULOS

O próximo passo agora é iniciar os cálculos baseado nos dados em questão. Para completar tal tarefa é necessário clicar no botão "Iniciar", situado no canto superior direito da tela, tal como mostrado na figura abaixo.

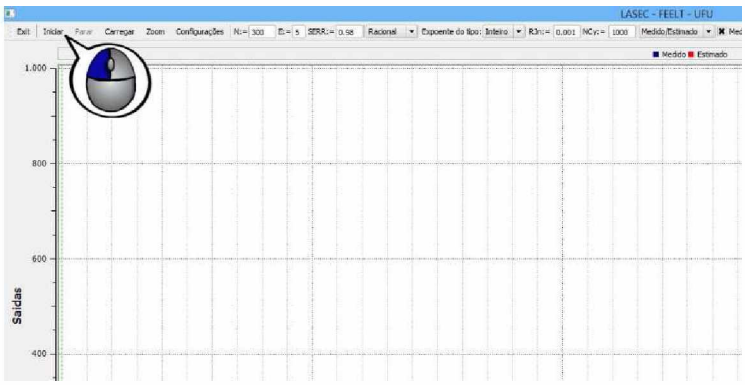


Figura A.6 – Inicialização da identificação do sistema

O sistema em conjunto com o software começa então a processar os dados recebidos pelo operador e a realizar os cálculos para a obtenção de um modelo que represente os dados carregados, ou seja, com os erros minimizados. Os resultados do melhor modelo são apresentados graficamente e numericamente na tela, o tempo que

levará para que estes sejam definidos dependerá da capacidade física do computador que estará portando o programa.

Na imagem abaixo o gráfico mostrado representa duas curvas concomitantes: Medido e Esperado. Isto significa que é mostrada a curva do resultado que se busca encontrar, o ideal, e daquele que fora calculado. Percebe-se que a diferença, ou o erro, entre estes é extremamente pequeno, frisando a eficiência do software.

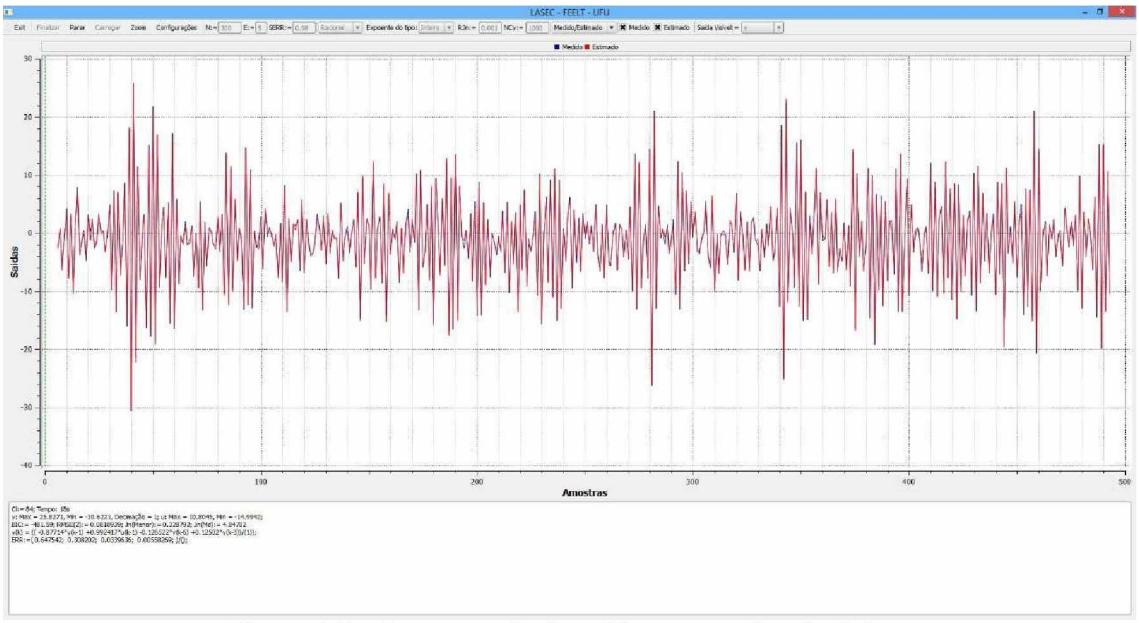


Figura A.7 – Apresentação do gráfico e equação calculados

A.3 - SELEÇÃO DE CURVA

Pode-se ainda obter um gráfico de distribuição de resíduos, é nele que é feito a validação dinâmica do melhor modelo daquele instante.

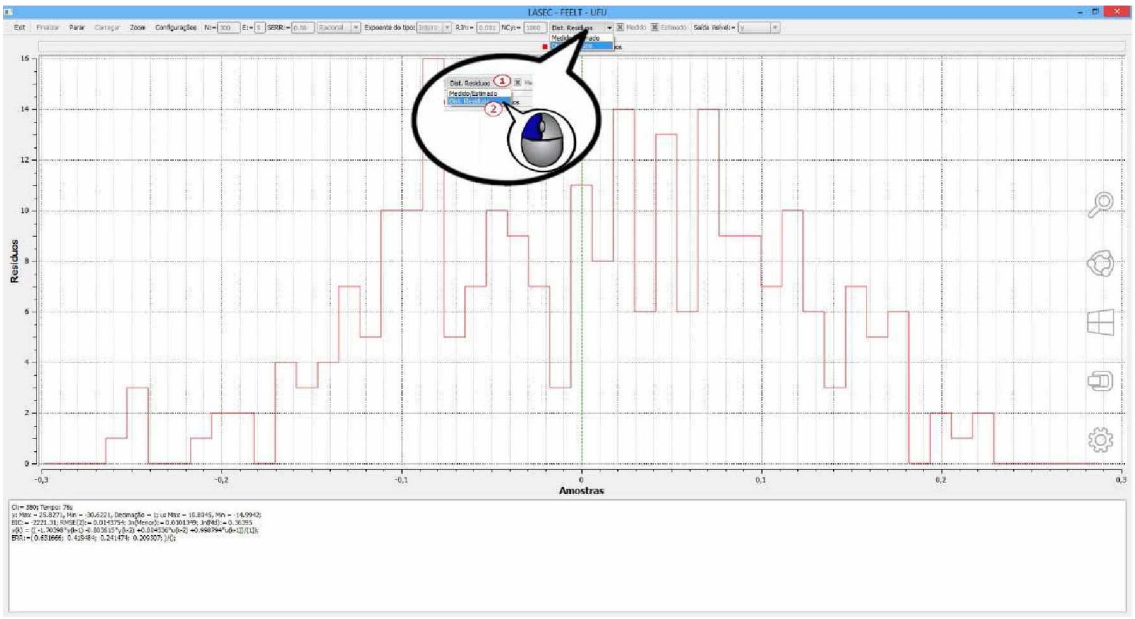


Figura A.8 – Gráfico de distribuição de resíduos

A imagem mostrada na Figura A.9 representa a curva com os resultados de cálculo esperado. Para que apenas ela apareça é necessário desmarcar a caixa de seleção nomeada como “Medido”.

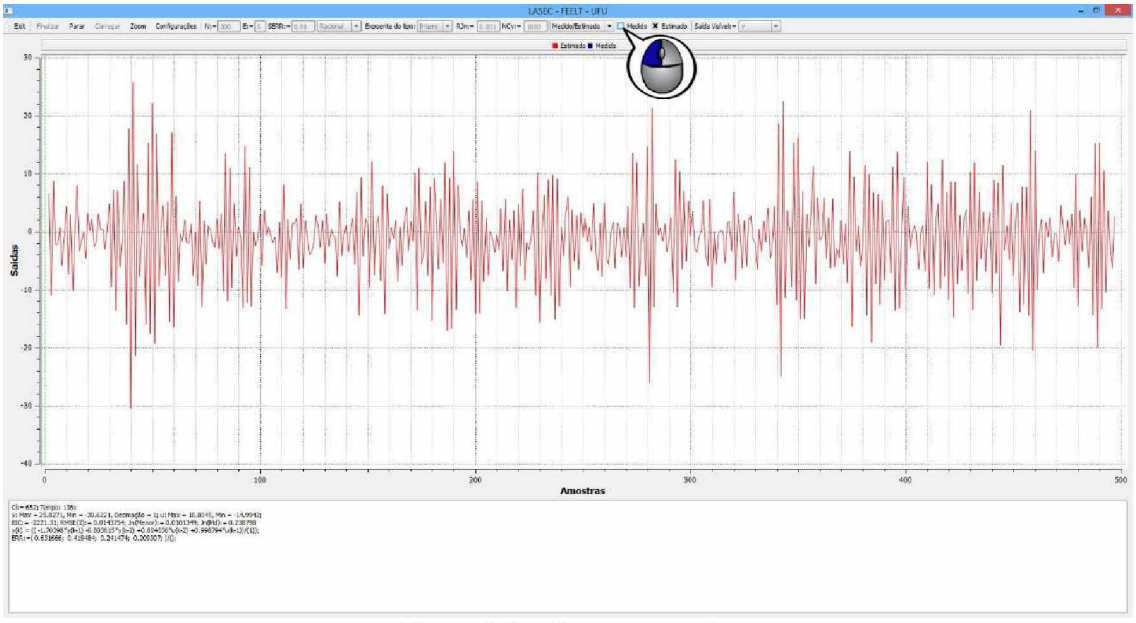


Figura A.9 – Curva esperada

Caso o operador queira visualizar somente a curva medida, basta desmarcar a caixa de seleção nomeada de “Esperado”, assim como foi feito no passo acima. Desta

forma é possível verificar os gráficos separadamente, e facilita assim a análise para alguns casos.

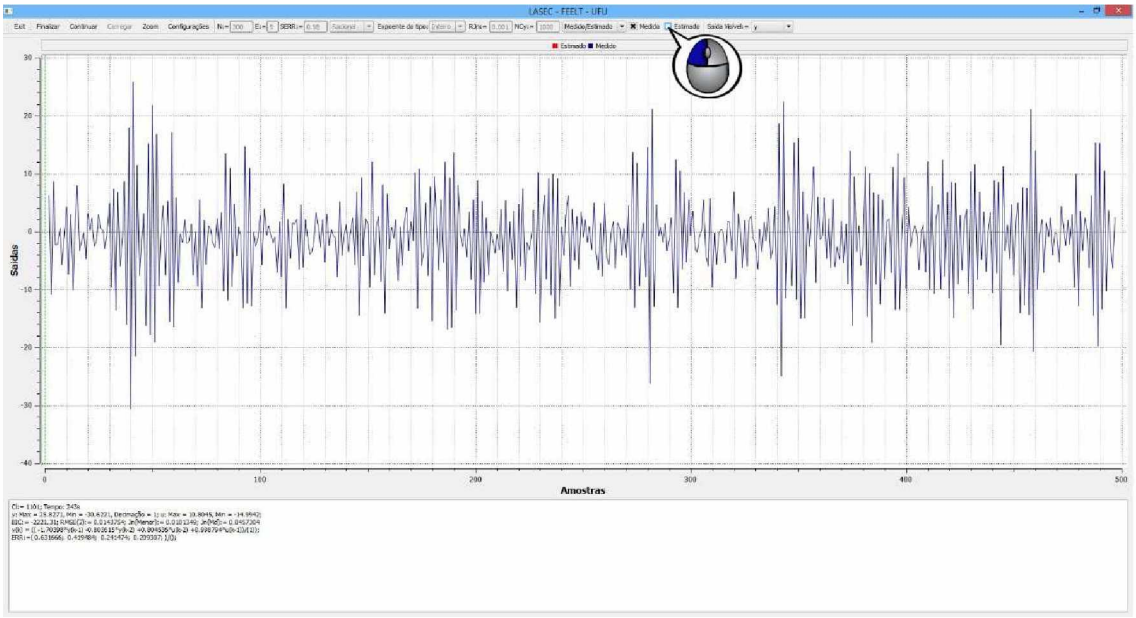


Figura A.10 – Curva medida

A.4 - PARAR OS CÁLCULOS

Caso seja necessário parar a inicialização do sistema para fazer modificações que são disponibilizadas pelo software apenas com a função "Parar" ou "Finalizar" ativadas, ou por outro motivo qualquer, basta clicar no botão "Parar". As tarefas do tópico 5 até o 11 só podem ser realizadas com esta função ativada, ou seja, quando o software não estiver executando cálculos.

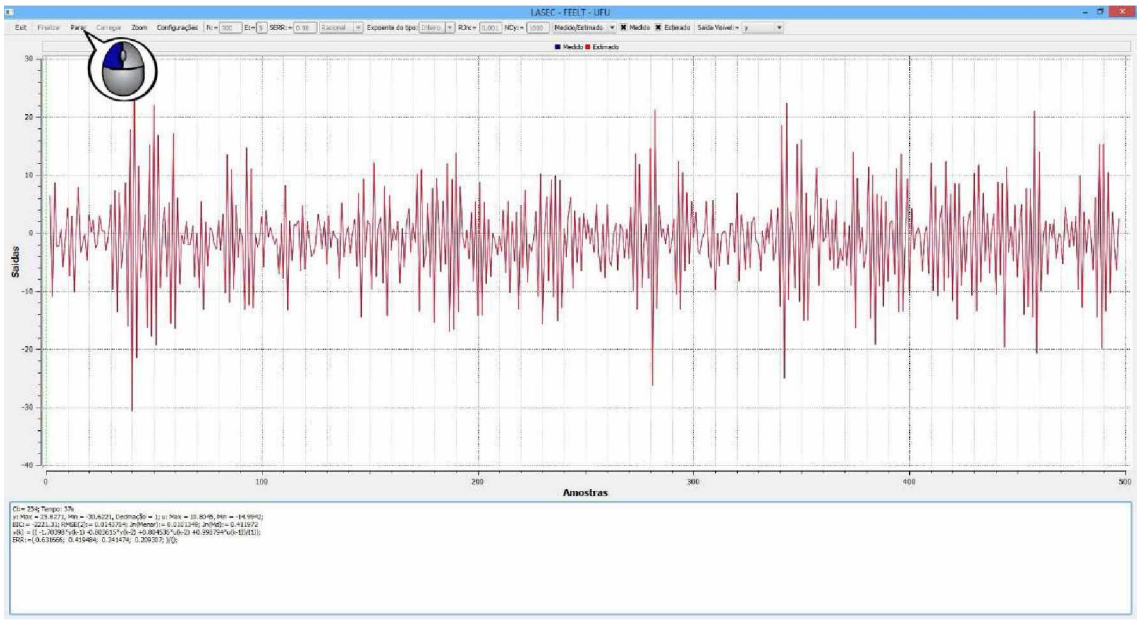


Figura A.11 – Parando os cálculos

A.5 - SELEÇÃO DE SAÍDA VISÍVEL

Se houverem mais de uma variável de saída, provavelmente cada uma terá um resultado diferente. Devido a este fato é possível escolher qual saída deve estar visível no gráfico. Para completar tal ação basta selecionar a saída na caixa de seleção localizada no canto superior esquerdo "Saída visível".

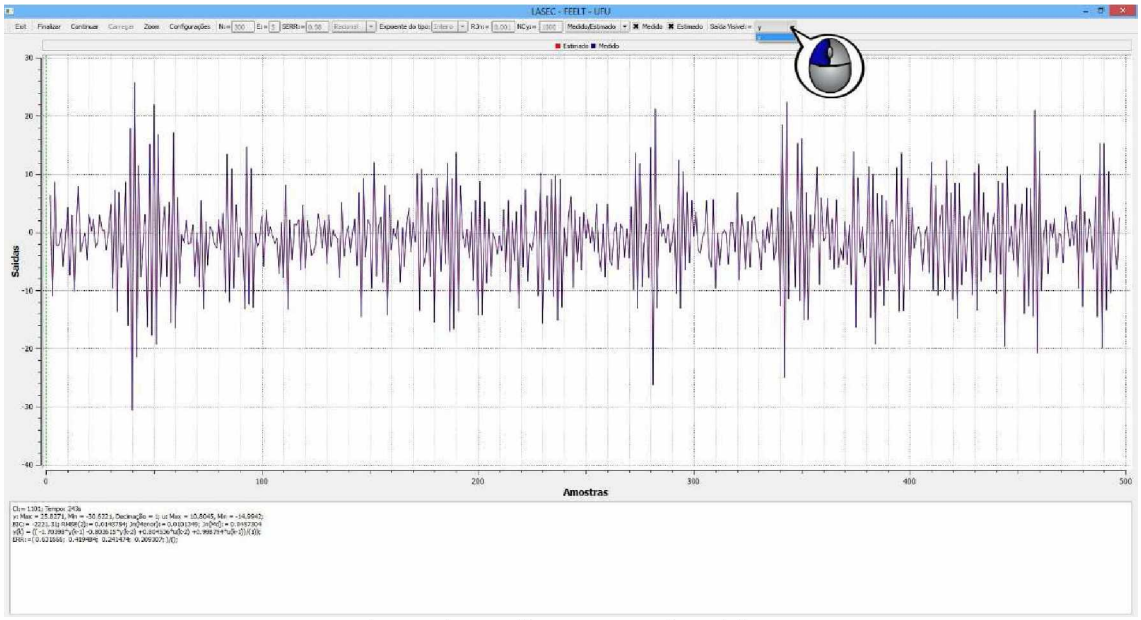


Figura A.12 – Seleção de saída visível

A.6 - FUNÇÃO ZOOM

Esta é uma função muito útil caso haja necessidade de uma análise minuciosa de determinado local da curva. Para isto basta clicar no ícone "Zoom", situado na barra de tarefas do programa, e arrastar o mouse de um ponto até outro no local que deseja-se ampliar o campo de visão.

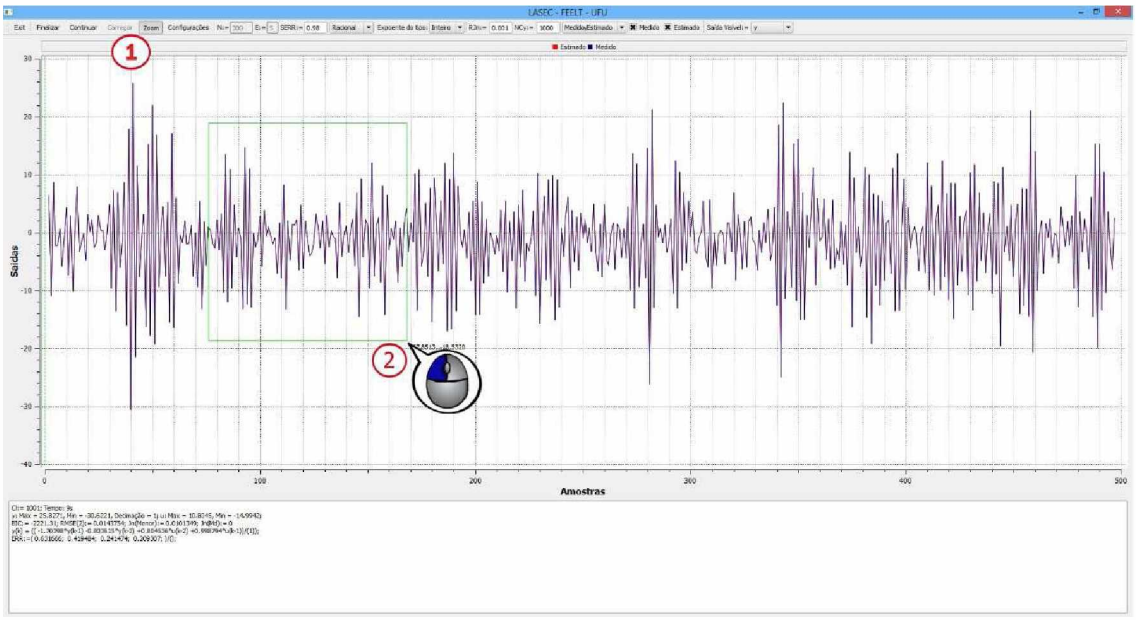


Figura A.13 – Aplicando Zoom

Para minimizar o zoom, deve-se clicar novamente no botão "Zoom" e a imagem voltará ao tamanho original. Na Figura A.14, é mostrado a curva com a ferramenta aplicada, percebe-se que há muito mais detalhamento visual no gráfico.

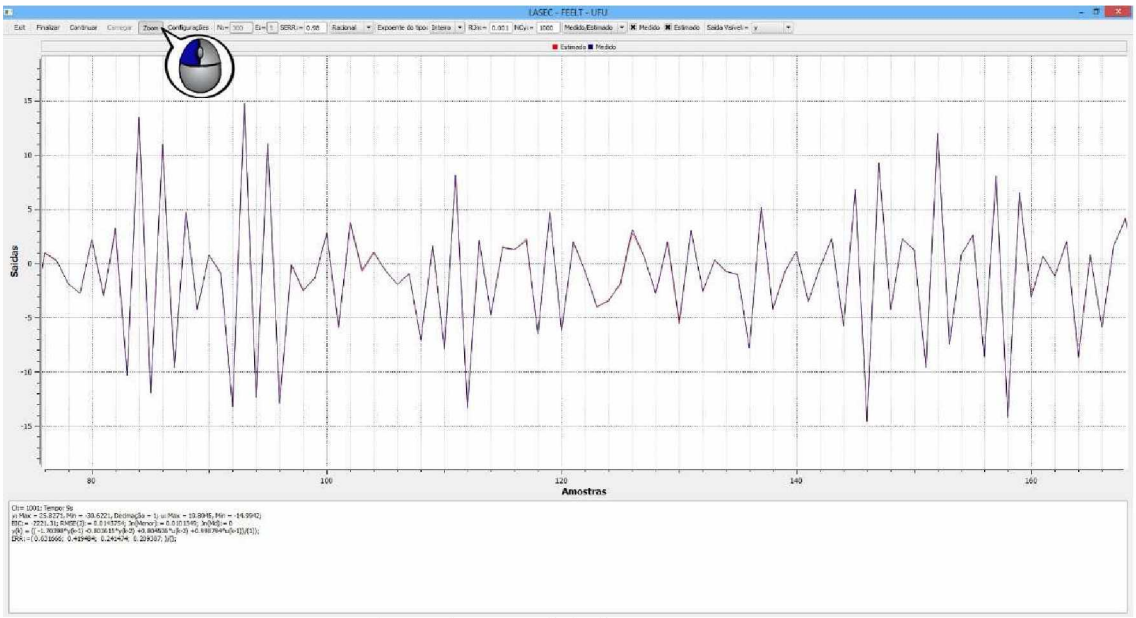


Figura A.14 – Minimizando o Zoom

A.7 - ESTRUTURA NARMAX POLINOMIAL OU RACIONAL

É possível modificar o tipo de estrutura utilizada pelo software entre Polinomial ou Racional. É importante ressaltar que caso seja escolhido a opção Racional não significa que um modelo polinomial não possa ser encontrado já que um modelo polinomial esta contido no domínio que um modelo racional pode abranger. Para efetuar esta seleção deve-se clicar na caixa de seleção que contém as opções "Racional" ou "Polinomial" e escolher uma delas.

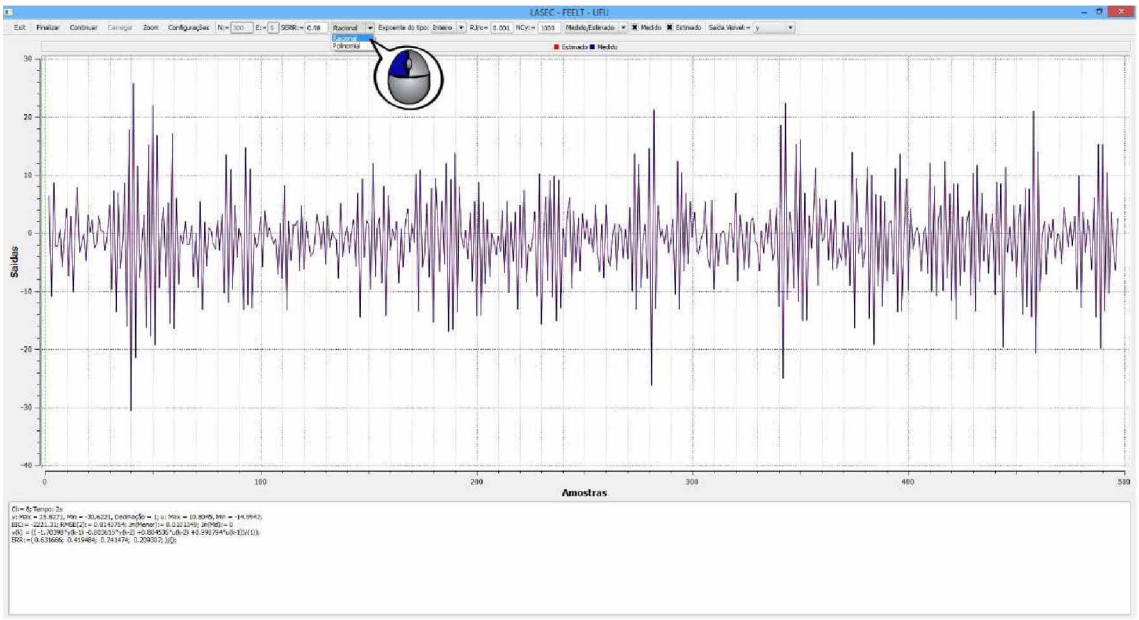


Figura A.15 – Determinando a forma do resultado

A.8 - TIPO DO EXPOENTE

O tipo de expoente utilizado pelo software pode ser selecionado entre Natural, Inteiro ou Real, basta selecionar uma das três opções na caixa de seleção "Expoente do tipo".

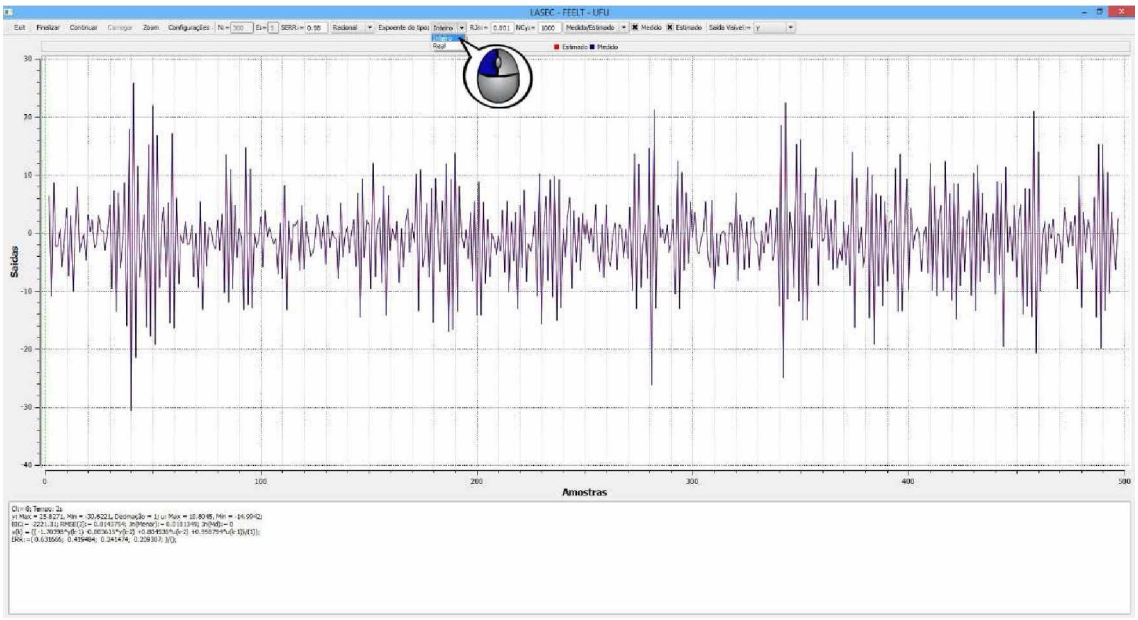


Figura A.16 – Modificando o tipo do expoente

A.9 - TAXA DO ERRO QUADRÁTICO MÉDIO (RJN)

Uma das seleções necessárias como critério de parada é o valor mínimo de progressão na taxa de redução do erro quadrático em um dado numero de iteração. E este valor é ajustado pela variável RJn apresentada na Figura A.17.

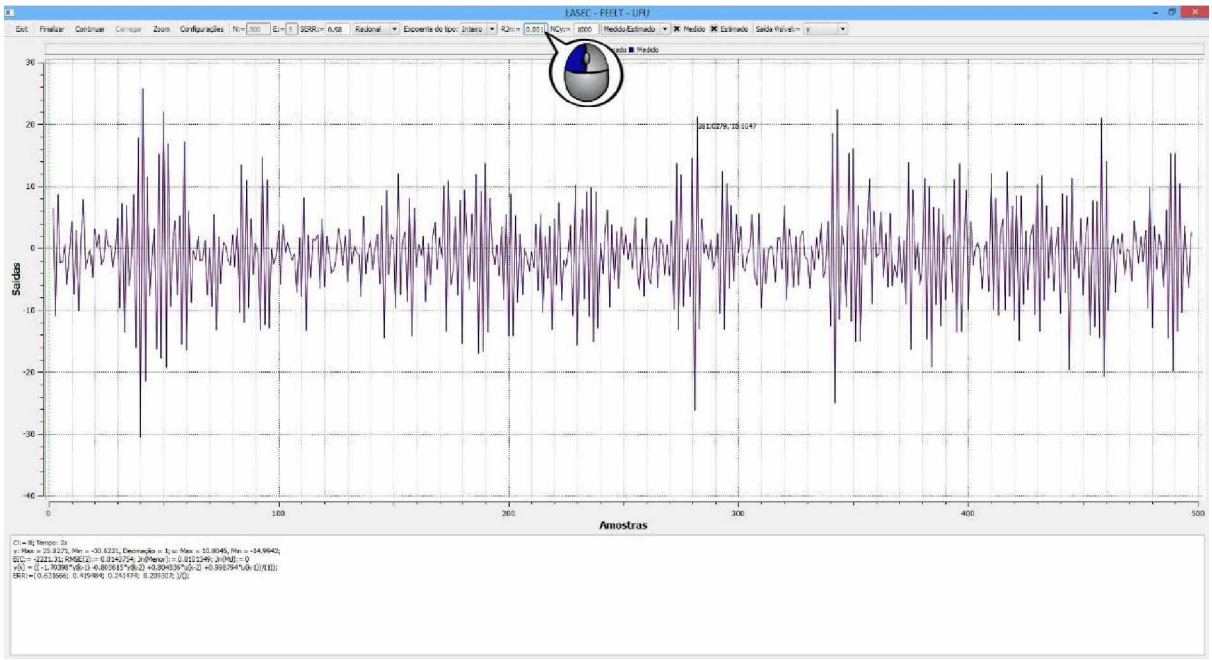


Figura A.17 – Modificando o valor do RJn

A.10 - NÚMERO DE CICLOS SEM ALTERAÇÃO CONSIDERAVEL (NCY)

O número de iteração do critério de parada é modificado pelo valor de NCy basta seguir as mesmas instruções dadas para o RJn do passo acima e assim como mostrado na Figura A.18.

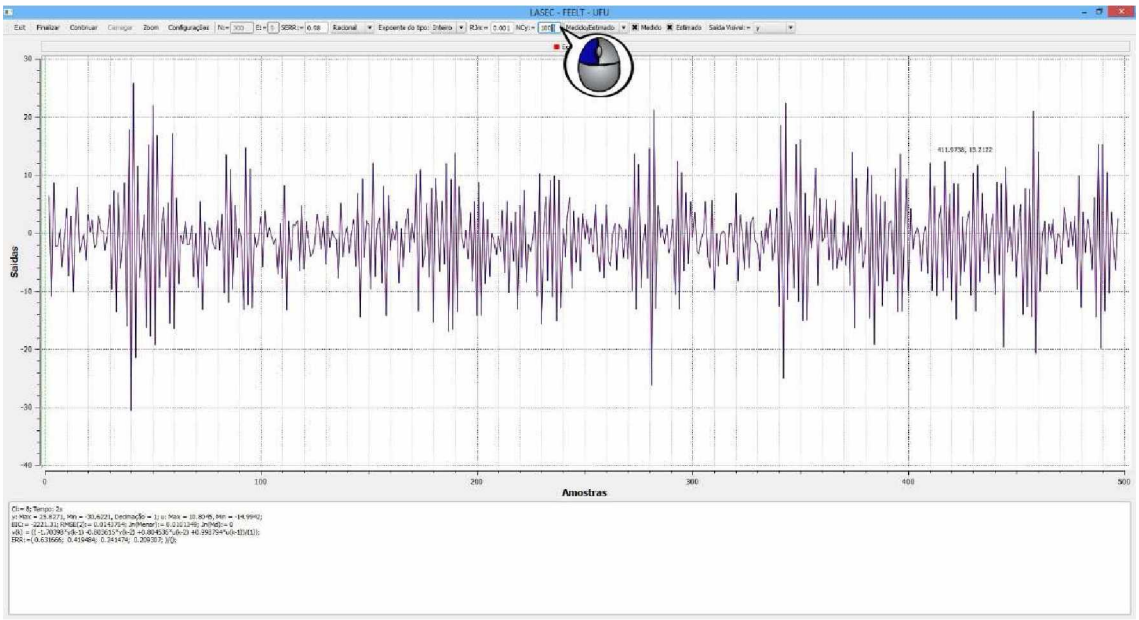


Figura A.18 – Modificando o valor de NCy

A.11 - SOMATÓRIO DE ERR DESEJADO (SERR)

Para um dado conjunto de regressores em teste o aplicativo calcula o valor de ERR para cada um e seleciona os melhores ate que a somatória deste índice dos melhores atinja o valor da variável SERR que vai de [0,1]. Inicialmente este valor é ajustado para 0.98 e caso não seja o desejado, é possível trocar seu valor clicando na caixa de texto "SERR:" e postando um novo valor.

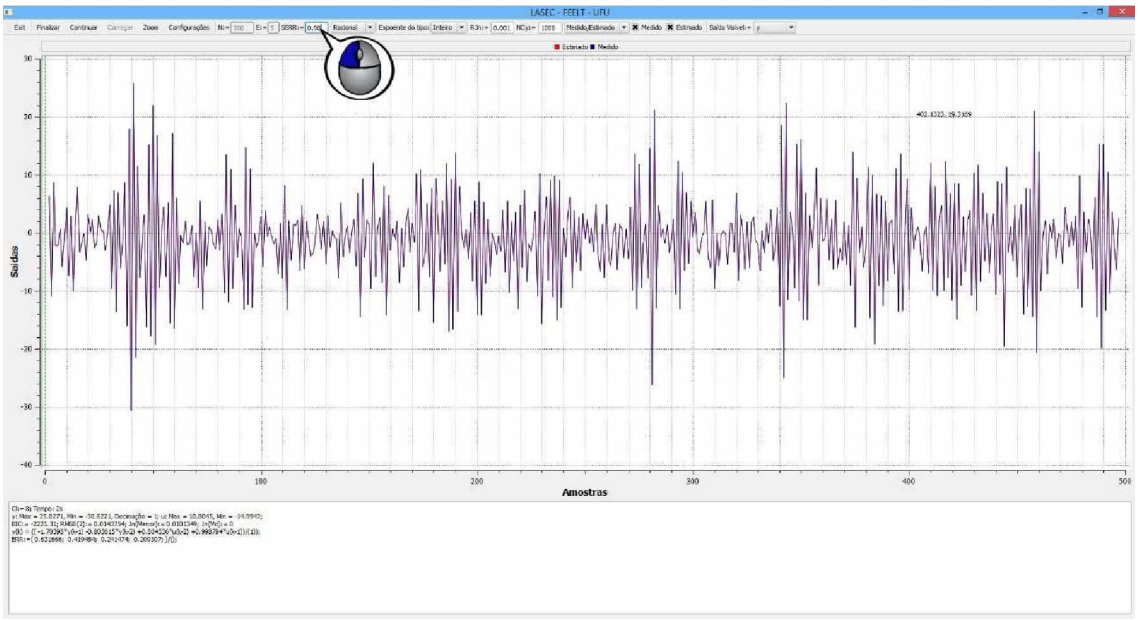


Figura A.19 – Determinando um novo valor para SERR

A.12 - CONTINUANDO AS OPERAÇÕES

Caso queira retornar às operações que foram de certa forma pausadas utilizando a função parar, deve-se clicar no botão "Continuar". Na Figura A.20, pode-se notar um par ordenado de números localizado no gráfico. Estes representam o local que o mouse se encontra em relação ao eixo y e x.

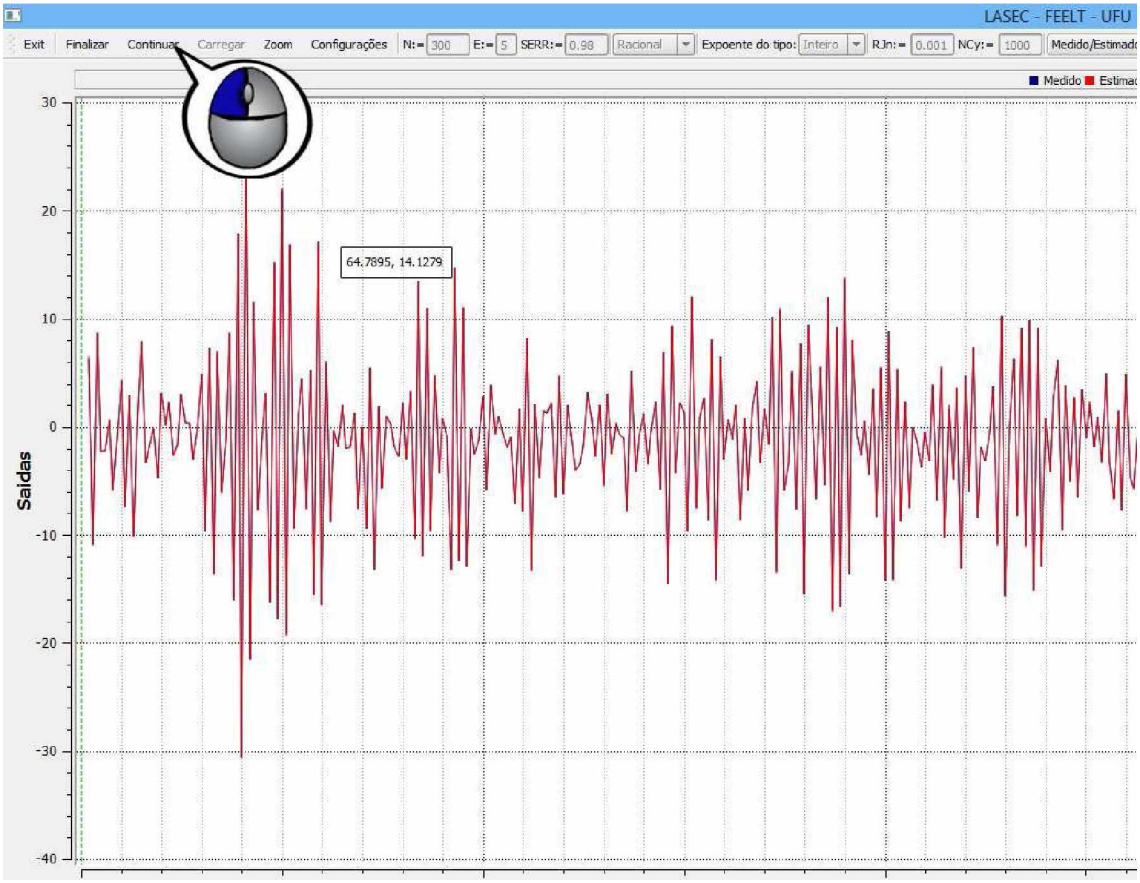


Figura A.20 – Continuando os cálculos

A.13 - FINALIZANDO OS CÁLCULOS

Após encontrar o resultado esperado é necessário finalizar as tarefas, tanto para sair do programa quanto se precisar trocar os valores de "N" e "E" que são o tamanho da população e número do elitismo respectivamente. O número do elitismo é usado quando deseja-se reiniciar o software mas mantendo uma quantidade de melhores

cromossomos que é dada por este índice. Desta forma, deve-se clicar no botão "Finalizar", localizado na barra de tarefas do software ao lado no botão "Continuar" para altera-los.

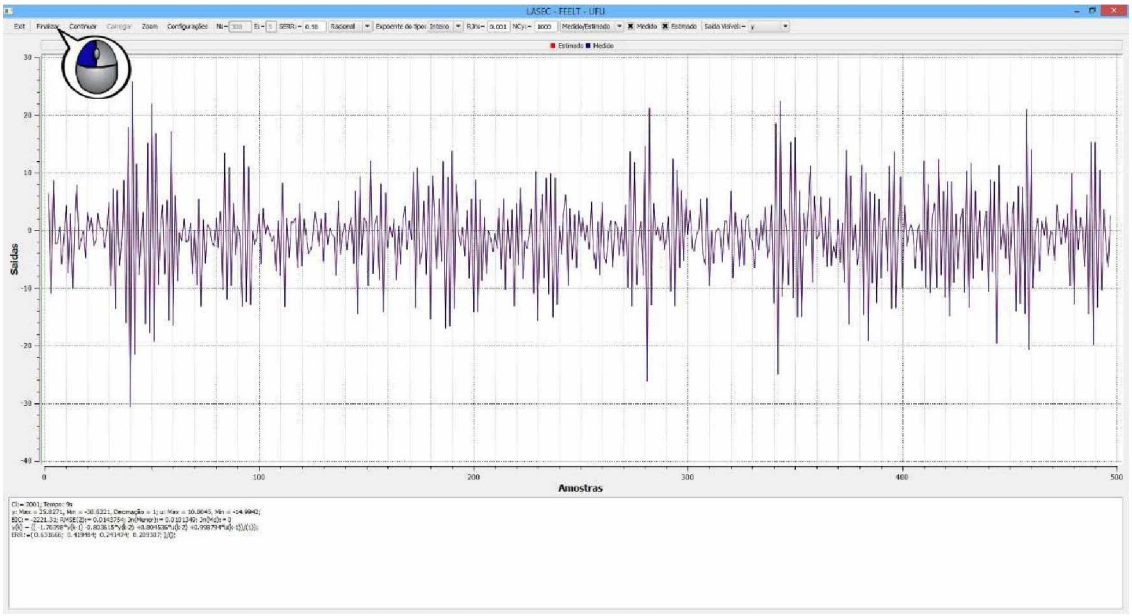


Figura A.21 – Finalizando os cálculos e tarefas

A.14 - TAMANHO DA POPULAÇÃO (N)

O "N" apenas pode ser modificado quando o botão finalizar for pressionado. Desta forma, basta trocar o valor localizado dentro da caixa de texto.

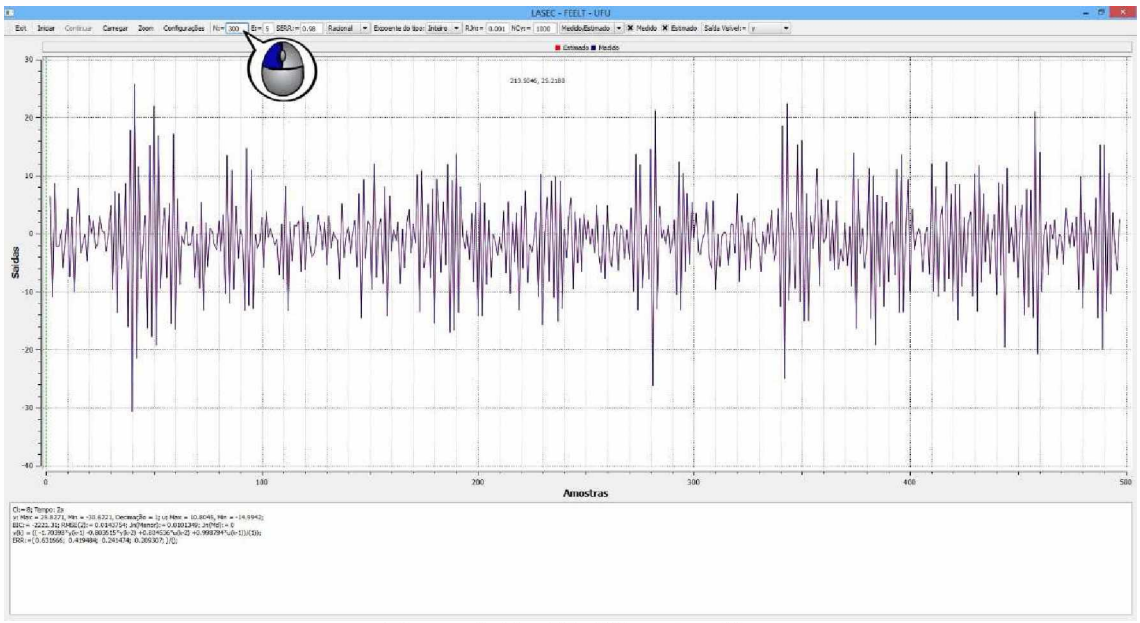


Figura A.22 – Modificando o N

A.15 - ELITISMO (E)

Análogo ao N deve-se apenas trocar o valor na caixa de texto.

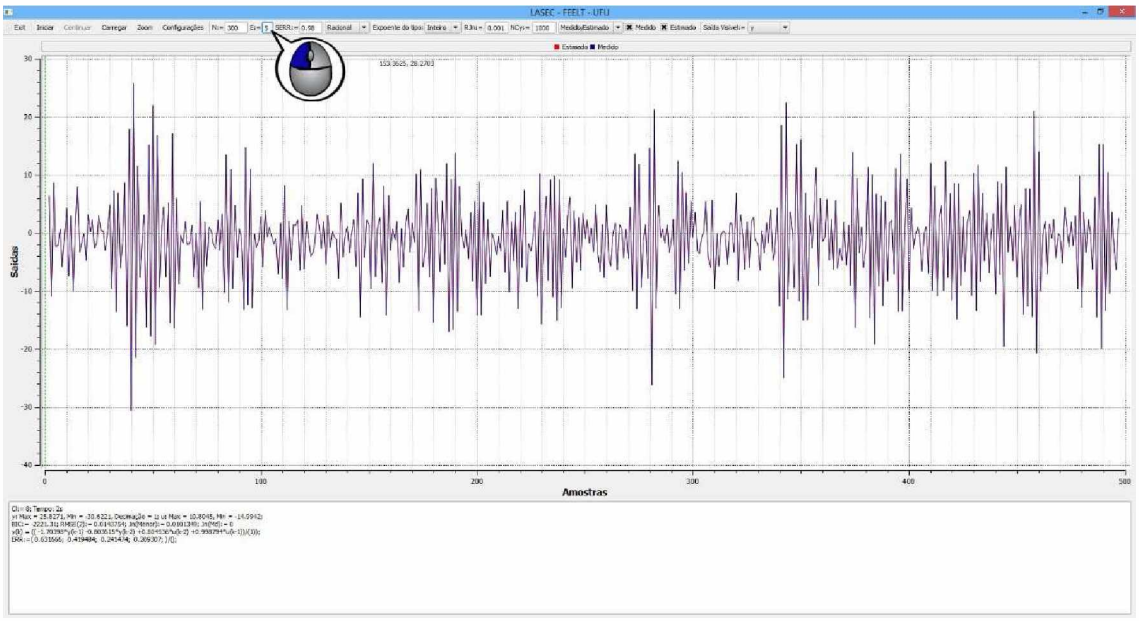


Figura A.23 – trocando o valor de E

A.16 - INICIANDO NOVAMENTE OS CÁLCULOS

Ao finalizar o software depois de encontrar o resultado esperado, há ainda a possibilidade de reiniciar o sistema clicando novamente no botão "Iniciar". Ao clicar neste campo uma caixa de diálogo aparece na tela para que caso for desejado, há como limpar os pontos encontrados anteriormente ou manter esses pontos do Elitismo conforme explicado anteriormente. Caso a escolha seja manter, o programa irá rodar novamente mantendo apenas os selecionados pelo elitismo e os demais serão criados de forma aleatória, no contrário todos os cromossomos serão criados de forma aleatória.



Figura A.24 – Reiniciando o sistema

A.17 - SAINDO DO SOFTWARE

Para sair do programa é necessário que o mesmo seja finalizado antes. Feito isto, basta clicar em "Exit", primeiro campo da direita para esquerda na barra de tarefas e a tela do software irá se fechar.

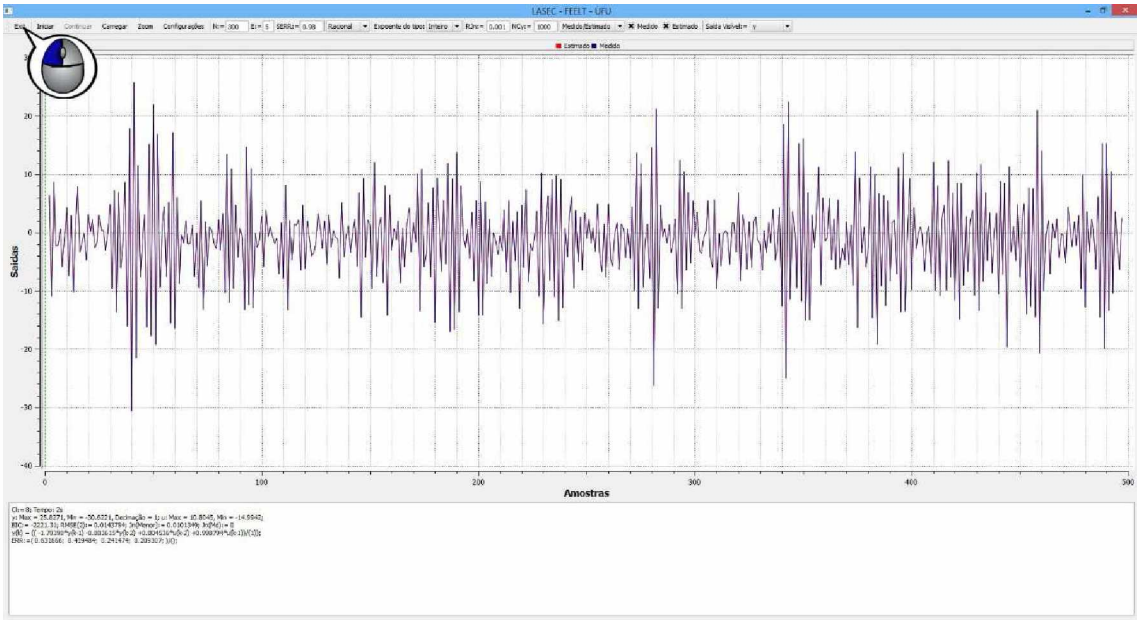


Figura A.25 – Saindo do Software

A.18 - MENU DE CONFIGURAÇÃO

No menu de configuração é possível salvar os dados que foram carregados no software, caso seja necessário salvar em um local específico, por exemplo, além de haver a possibilidade de salvar uma quantidade específica de cromossomos. Pode-se ainda concatenar ou carregar arquivos nesta aba também.

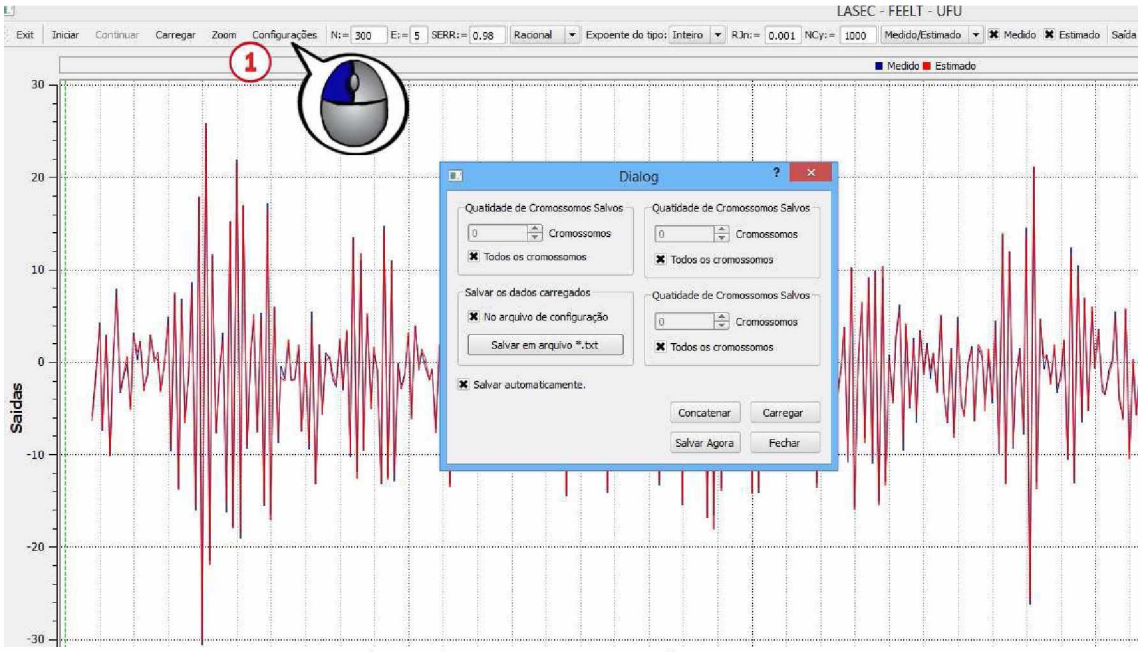


Figura A.26 – Menu de configurações

Para salvar o arquivo atual deve-se navegar até o menu de "Configurações" e no campo "Salvar dados carregados" e clicar no botão "Salvar". Irá se abrir uma janela, nela deve-se digitar um nome para o arquivo e o local que ele será salvo, logo após estes passos, clique no botão "Salvar".

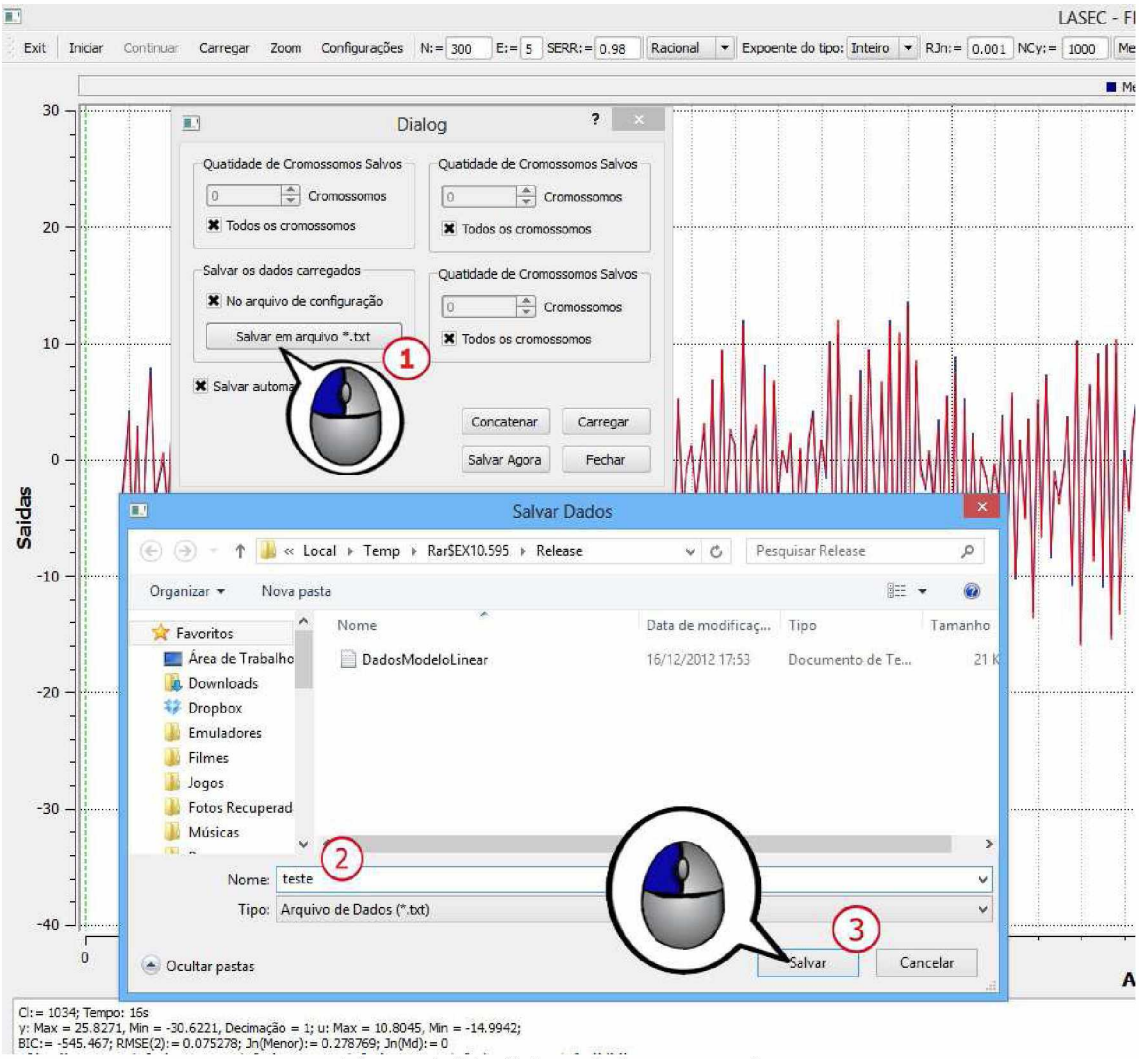


Figura A.27 – Salvar dados carregados

Há a possibilidade de salvar um número específico de cromossomos. Para isto, deve-se seguir no menu de configurações o campo "Quantidade de cromossomos salvos" e clicar na seta do lado da caixa de texto para aumentar ou diminuir a quantidade desejada. Depois disto clique em "Salvar agora", então uma janela irá se abrir. Nomeie o arquivo e clique em salvar. É importante lembrar que este arquivo pode ser salvo em formato "*.xml" ou "*.xbel".

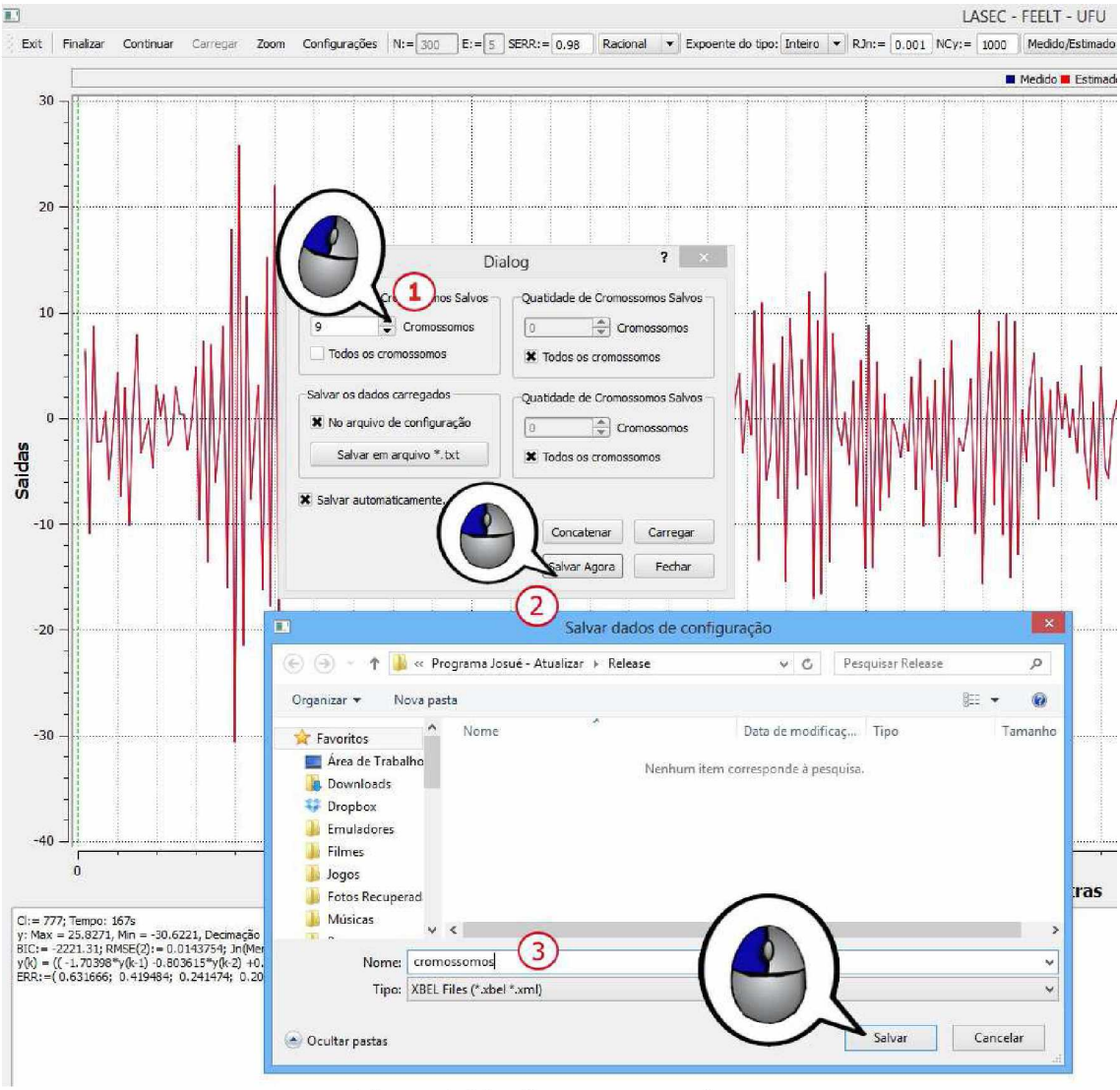


Figura A.28 – Cromossomos salvos

A ferramenta de concatenar arquivos está disponível tanto no menu "Carregar" como em "Configurações". Para utilizar esta ferramenta no menu de configuração, clique no botão "Concatenar". Irá surgir na tela uma janela do Explorer na qual deve-se selecionar o arquivo em formato "*.txt" que deseja-se concatenar ao software.

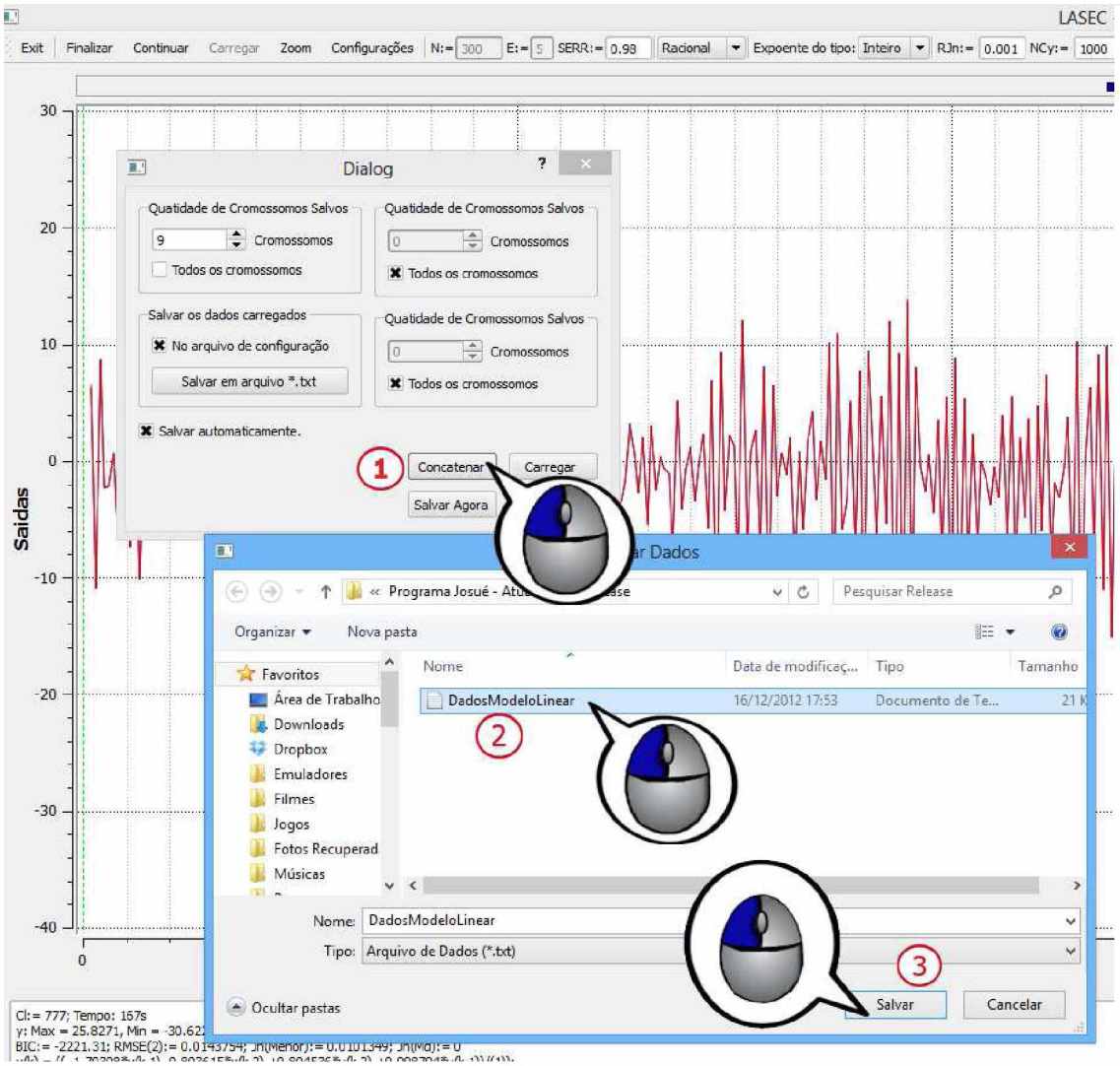


Figura A.29 – Concatenando arquivos

APÊNDICE B

TÉCNICAS DE PROCESSAMENTO PARALELO UTILIZADAS

Aumentar o clock do processador era uma técnica utilizada para elevar a quantidade de processamento em um determinado espaço de tempo. Porém, esta técnica alcançou um patamar onde o custo do resfriamento do processador tornou-se financeiramente inviável (BENINI, 2005). Outra limitação dos processadores de apenas um núcleo (singlecore) é a grande diferença entre a velocidade da memória e a do processador, aliada à estreita banda de dados, que faz com que aproximadamente 75 % do uso do microprocessador seja gasto na espera por resultados dos acessos à memória (Nikolas Aschermann, 2007).

Na maioria dos processadores de mais de um núcleo, cada núcleo pode possuir uma memória cache própria e, em alguns casos, realizam acesso direto e independente à memória principal, permitindo que cada processador realize os cálculos para os quais foi requisitado, concorrentemente com o outro processador, elevando, desta maneira, o desempenho global do sistema. No entanto, o bom aproveitamento dos vários núcleos em processadores que possuem vários núcleos (multi-core processors), depende do modo com que o software aplicativo foi projetado e implementado (Foster, 2009).

Processadores "multi-core" possuem um bom desempenho, performance, disponibilidade e segurança a um menor custo (ALVES, 2007). Por outro lado, esta tecnologia possui algumas desvantagens, porque aumenta a utilização dos recursos de hardware e depende dos recursos disponíveis tanto do sistema operacional de apoio quanto do software aplicativo, pois a capacidade dos processadores multi-core depende

da utilização de vários segmentos dentro das aplicações para aumentar o desempenho da aplicação (FREITAS, 2006).

Os AGs são algoritmos usados em várias aplicações com o intuito de aperfeiçoar parâmetros, no entanto, o processo de melhoramento da solução é normalmente lento e exige elevado esforço computacional. Dependendo da complexidade do problema, uma solução considerada boa pode demorar dias, semanas ou até meses (Celso Camilo, 2007).

Por isso, neste trabalho, os operadores genéticos, isto é, o operador mutação, cruzamento, seleção, dentre outros, são executados em paralelo desde que o processador seja "multi-core" e o sistema operacional possibilite a utilização dos núcleos.

Neste sentido, este capítulo apresenta às técnicas utilizadas para que o algoritmo de identificação de sistema proposto execute os operadores genéticos em paralelo, visando aproveitar toda capacidade computacional disponível no processo de solução da identificação do sistema.

B.1 - ESCOLHA DA LINGUAGEM DE PROGRAMAÇÃO

Para que fosse possível executar os operadores genéticos em paralelo, o primeiro desafio foi determinar a linguagem de programação que seria utilizada para programar o algoritmo de identificação de sistemas proposto.

Os critérios adotados para a escolha da linguagem de programação foram:

- Combinar características de linguagens de alto e baixo níveis;
- Possuir compilador freeware para PC e microcontroladores;
- Ser uma linguagem de programação multi-paradigma;
- Possuir framework multiplataforma para facilitar o desenvolvimento.

Com o objetivo de atender todos os critérios adotados, optou-se em utilizar o framework multiplataforma para desenvolvimento em C++, denominado Qt.

Optou-se pelo SDK Qt por ele ser um pacote multiplataforma, intuitivo e eficiente, ou seja, além de ser portável para vários sistemas operacionais, inclusive plataformas embarcadas, o código é sempre recompilado para a plataforma de trabalho, de forma a otimizar o código gerado, ao contrario de alguns que utilizam-se de maquinas virtuais e ou interpretadores para realiza esta compatibilidade com as plataformas.

Outro ponto é a grande variabilidade de ferramentas e módulos prontos, disponíveis neste pacote, que facilitam na codificação do sistema.

O Qt foi, originalmente, criado pela Trolltech, mas a Nokia adquiriu a Trolltech em 2008 e licenciou o QT sob a licença LGPL em 2009, o que eliminou as barreiras comerciais de licenciamento, e que impedia a adoção em larga escala do toolkit. Entre as empresas da atualidade que utilizam o Qt estão: Google, Amazon, Skype, Adobe e MathWorks (Qt.Nokia, 2012).

Além disso, ele possui vários recursos para o desenvolvimento de aplicativos que necessitam de processamento paralelo e facilita o desenvolvimento de aplicativos que necessitam ser executados em clusters, isto é, um aglomerado de computadores ligados em rede e que operam como se fossem uma única máquina de grande porte.

B.2 - A CLASSE MULTI-TRHEAD DESTRUCT DO QT

O SDK (Software Development Kit), conhecido como Qt, possui uma serie de recursos que facilitam a implementação de um sistema multitarefa (Qt.Nokia, 2012).

A classe QThread é um desses recursos cuja função é criar threads independentes. Cada thread fornece controle independente dentro do programa e pode

compartilhar dados com outras threads dentro do processo, porém, cada threads é executada separadamente como se fosse um programa separado que é executado em um sistema operacional multitarefa. Por padrão, uma QThread inicia a sua execução com o método denominado run( ), ao invés de iniciar no main ( ), que é a primeira função a ser executada por qualquer programa em C.

Para se criar uma thread é necessário criar uma classe e fazer de QThread uma subclasse e reimplementar o método run( ). Neste sentido, neste trabalho, criou-se uma classe denominada DEStruct, que possui como subclasse QThread e que contém o método run( ) reimplementado com o código que cada thread deverá conter. Desta maneira, cada thread que for criada, será alocada pelo sistema operacional em um núcleo diferente do processador.

Para que o aplicativo desenvolvido faça bom uso do recurso de hardware disponível do processador em que estiver sendo executado, ele identifica a quantidade de núcleos do computador usando o recurso QThread::idealThreadCount() e criar uma Thread para cada núcleo existente.

No entanto, quando se cria várias threads, é necessário que cada uma delas acesse uma determinada região de memória por vez. No aplicativo proposto, este controle foi realizado por meio das classes QSemaphore, QReadWriteLock, QWaitCondition e QMutex, fornecidas pelo SDK do Qt.

O controle e a identificação das threads criadas é realizada por meio de algumas variáveis estáticas, geradas para este fim, conforme mostrado no pseudocódigo da Figura B.1. Elas permitem identificar quantos objetos foram criados, isto é, quantas threads foram geradas e a sequência em que foram geradas.

```
class DEStruct : public QThread
{
    Q_OBJECT
public:
    static volatile qint16 DES_TH_size;
    qint32 DES_TH_id;

    DEStruct(){DES_TH_id=DES_TH_size; DES_TH_size++;};
    ~DEStruct();
};
```

Figura B.1 – Pseudocódigo dos atributos da classe DEStruct.

Além disso, para se obter o máximo dos recursos de hardware disponível, é necessário realizar um controle das tarefas que necessitam ser executadas, de modo a garantir que cada núcleo do processador não fique ocioso. Por isso, no aplicativo proposto, criou-se o método run( ) que será comentado no próximo item.

B.3 - O MÉTODO RUN.

No intuito de otimizar a alocação de recursos de hardware, criou-se uma variável global denominada DES_modeOper_TH na classe DEStruct. Esta variável é responsável por identificar a tarefa que a Thread deve executar em um determinado momento. Sendo que, cada thread pode executa uma das seis tarefas listadas a seguir:

1. Termina a *thread* para fechar o programa;
2. Tarefa finalizou e a *thread* espera o próximo comando. As classes utilizadas são QSemaphore, QWaitCondition e QMutex;
3. Thread para onde estiver e espera para continuar ou finalizar, conforme comando do usuário;
4. Executando o algoritmo DE, em cada *thread* criada, simultaneamente;
5. Carrega e condiciona os dados em cada *thread* criada, ao mesmo tempo;
6. Normaliza os dados em cada *thread* criada, simultaneamente.

O pseudocódigo do método run( ), que pertence a classe DEStruct, utilizado para implementar as seis tarefas apresentadas, é mostrado na Figura 4.2.

As chamadas DES_AlgDiffEvol( ), DES_Carregar( ) e DES_Normalizar( ), que são realizadas nas tarefas 3, 4 e 5, são responsáveis por identificar o sistema, carregar os dados do sistema que será identificado a partir de um arquivo "txt" e normalizar os dados, respectivamente. O arquivo "txt" mencionado contém os valores das entradas e o valor das respectivas saídas do sistema que se pretende identificar.

O Método DES_Carregar utiliza a primeira threads, criada com o objetivo de identificar o sistema, para calcular o número de valores armazenados no arquivo "txt" e dividir o valor encontrado pelo número total de threads criadas, sendo que o número total corresponde ao número de núcleos presente no processador.

```
void DEStruct::run()
{
   while(DES_modeOper_TH)
   {
      switch(DES_modeOper_TH)
      {
          case 0://Termina a Thread para fechar o programa
              break;
          case 1://Tarefa finalizou e a Thread espera o próximo
comando.
              mutex.lock();
              if(DES_waitThread.tryAcquire()) DES_sync.wait(&mutex);
              else
              {
                  DES_waitThread.release(DES_TH_size-1);
                  DES_sync.wait(&mutex);
              }
              mutex.unlock();
              break;
          case 2://Thread para onde estiver e espera para continuar
ou finalizar
              break;
          case 3://Executando o DE em Multi-Thread
              DES_AlgDiffEvol();
              break;
          case 4://Carregando e condicionando os dados em Multi-
Thread
              DES_Carregar();
              break;
          case 5://Normalizando os dados em Multi-Thread
              DES_Normalizar();
              break;
      }
   }
}
```

Figura B.2 – Pseudocódigo do método run ( ).

Então, cada thread irá abrir o arquivo "txt" e efetuar a leitura da fração de pontos que lhe compete carregar e, após carregar os dados, o algoritmo realiza o cálculo da taxa de decimação, de modo a evitar que os regressores tendam a se tornar linearmente dependentes (redundantes), durante o processo de estimação dos parâmetros e, também, impede o falseamento ("alising") do sinal (Aguirre, 2007). Em seguida, cada thread irá salvar os dados, de acordo com a taxa de decimação calculada, em uma matriz diferente.

A última thread que terminar de armazenar os pontos na matriz, irá verificar se é necessário normalizar os dados lidos. Para verificar se é necessário normalizar, o algoritmo da thread lê os dados da primeira linha do arquivo *.txt e, caso o nome das variáveis não esteja acompanhado dos valores de máximo e mínimo, que são utilizados durante o processo de normalização, o método DES_Normalizar será executado e cada thread criada irá normalizar a faixa de dados que lhe foi designada.

B.4 - O MÉTODO DES_ALGDIFFEVOL.

Criar uma thread para cada núcleo do processador não garante a utilização total dos recursos de hardware de cada núcleo. Para se obter o total aproveitamento, é necessário que haja um gerenciamento eficiente das tarefas que cada thread está realizando, de modo que cada núcleo receba tarefas de modo ininterrupto, ou seja, a medida que uma thread termina de executar suas tarefas, é necessário iniciar outra thread para que o processador não fique ocioso.

Deste modo, desenvolveu-se uma estratégia de Gerenciamento de Tarefas para a Execução de Tarefas em Modo Contínuo (GTETMC), para manter todos os núcleos do

processador operando de modo constante, com o objetivo de aproveitar ao máximo os recursos de hardware disponível.

A Figura 4.3 mostra o diagrama esquemático dos vetores criados para implementar a técnica GTETMC proposta. Foram criados três conjuntos de dois vetores denominados de Pipelines, cuja função é gerenciar todas as tarefas criadas.

O algoritmo de cada thread pega o primeiro token disponível no Vetor de Tokens 1 e em seguida aplica os algoritmos de Mutação, Cruzamento, Seleção e Cálculo da Aptidão no cromossomo. Quando a thread termina de tratar o cromossomo, ela aloca o token que pegou no Vetor de Tokens 2 e procura outro token disponível no Vetor de Tokens 1 para poder tratar outro cromossomo.

Deste modo, todas as threads disponíveis irão pegar um token disponível, tratar o cromossomo, alocar o token que pegou no Vetor de Tokens seguinte e assim sucessivamente até que todos os cromossomos recebam o mesmo tratamento.

A thread que terminar de tratar o último cromossomo de um determinado Vetor de Tokens, irá realizar o elitismo do Vetor de Cromossomos e alocar o index da posição do cromossomo no Vetor de Elitismo correspondente ao Pipeline que estiver analisando, enquanto as outras threads continuarão a tarefa de tratar os cromossomos utilizando o Vetor de Tokens do próximo Pipeline.

O processo é cíclico, pois, após as threads utilizarem o Pipeline 03, elas voltarão a utilizar o Pipeline 01 e assim sucessivamente até atingirem o valor de jn (função custo básica dos mínimos quadrados), informado pelo usuário.

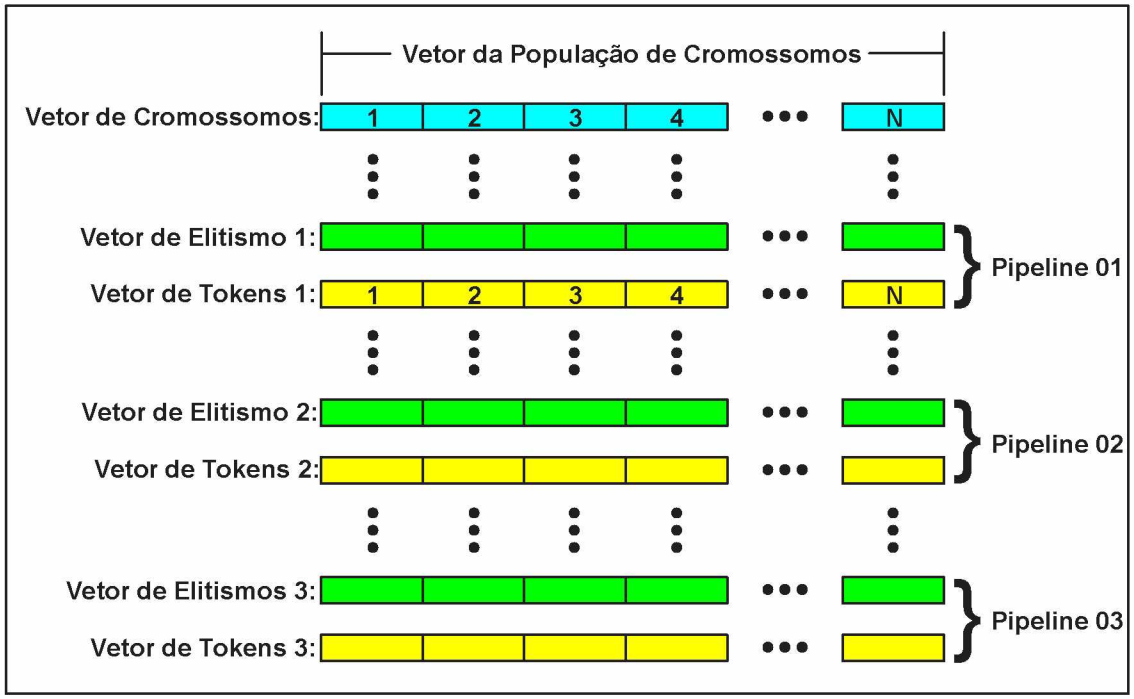


Figura B.3 – Diagrama esquemático dos pipelines criados para o Gerenciamento de Tarefas para a Execução de Tarefas em Modo Contínuo.

Vale salientar que o número de pipelines foi definido experimentalmente de modo a atender um processador intel i7 2a geração e 2GB de memória DDR3. Os testes experimentais realizados mostraram que somente dois pipelines são insuficientes para manter todos os núcleos do processador trabalhando de modo ininterrupto.

B.5 - RESULTADOS OBITIDOS

Foi avaliado a utilização de recursos do sistema proposto em 4 tipo de plataformas diferentes que são:

1. Plataforma com 2 núcleos reais conforme Figura B.4.
2. Plataforma com 2 núcleos reais e 2 núcleos simulados vide Figura B.5.
3. Plataforma com 4 núcleos reais apresentado na Figura B.6.
4. Plataforma com 8 núcleos reais apresentado na Figura 2.2

Em todos os casos, utilizando-se de 3 pipelines, o aplicativo desenvolvido, utilizou-se que quase 100% de processador das plataforma avaliadas, confirmando assim sua abrangência e grande capacidade de extrair o máximo da plataforma onde ele estava operando.

É possível que em maquinas com um número de núcleos maiores que os estudados a quantidade de pipelines tenha que ser aumentada também.

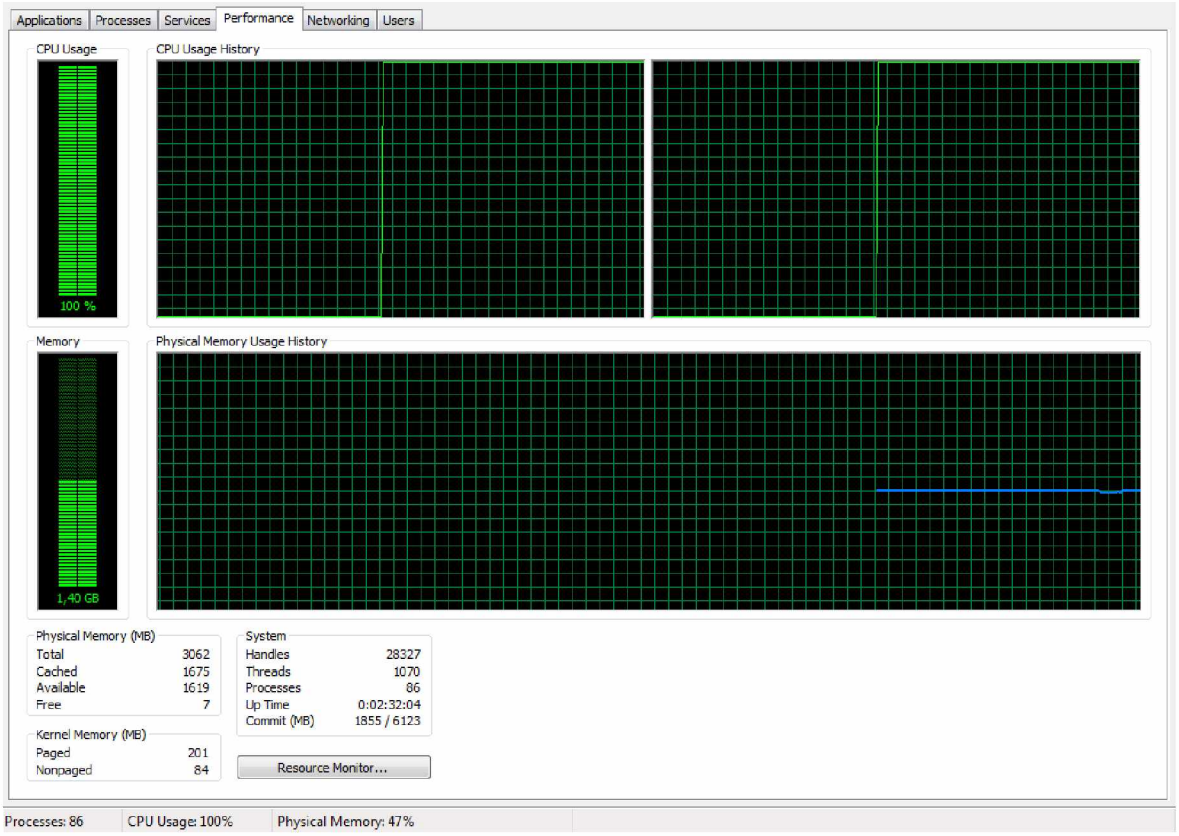


Figura B.4 –Software rodando em plataforma de 2 núcleos reais.

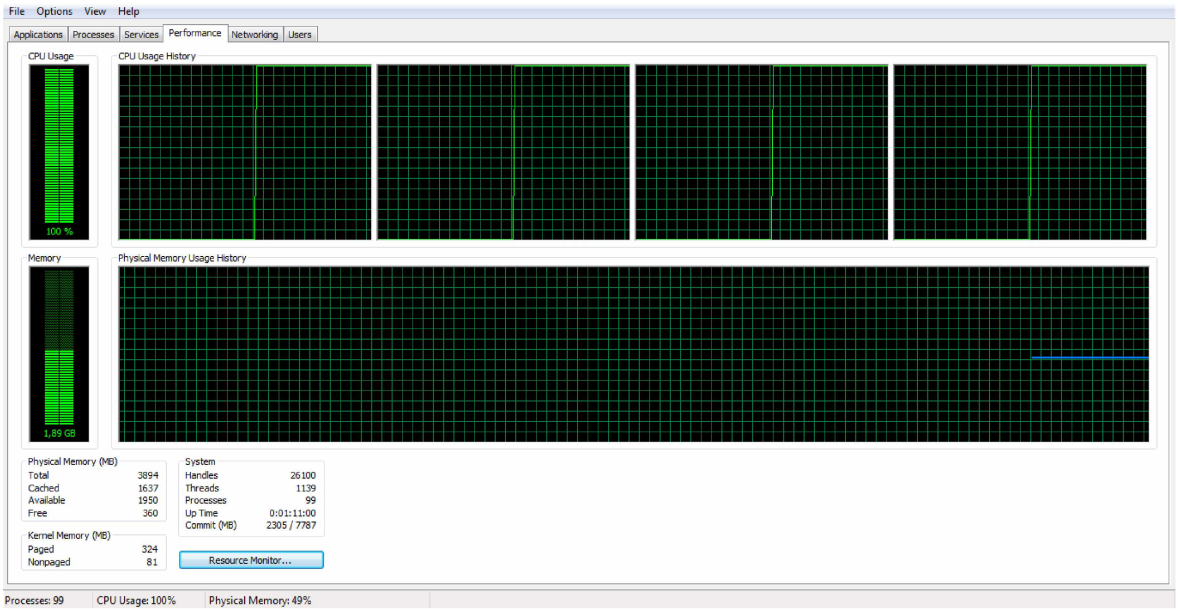


Figura B.5 – Software rodando em plataforma de 2 núcleos reais e 2 simulados.

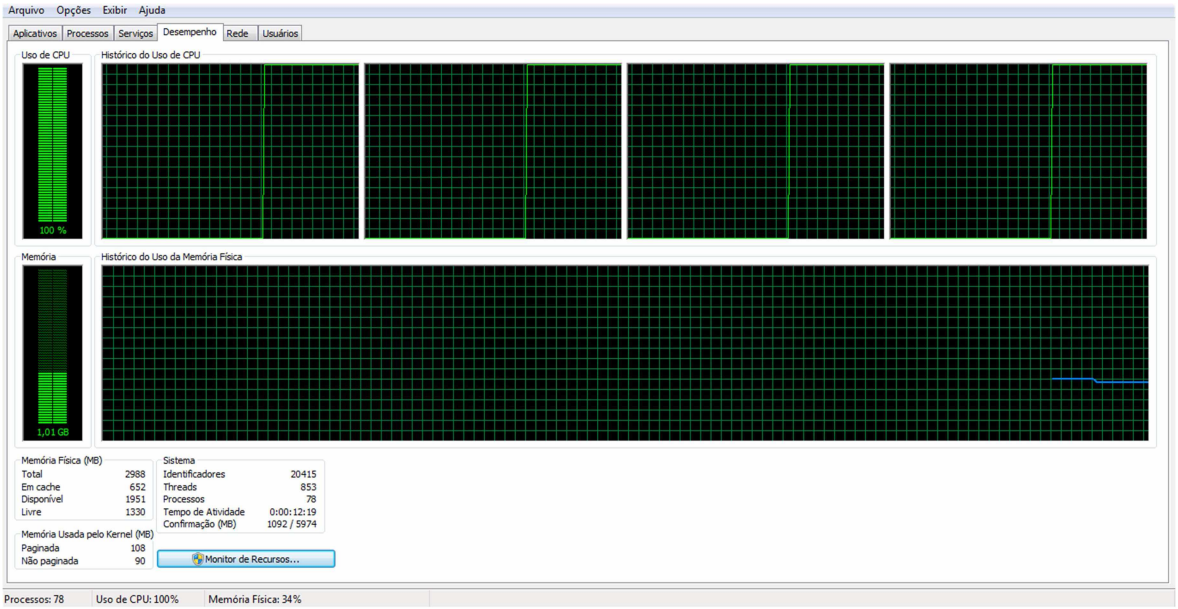


Figura B.6 – Software rodando em plataforma de 4 núcleos reais.

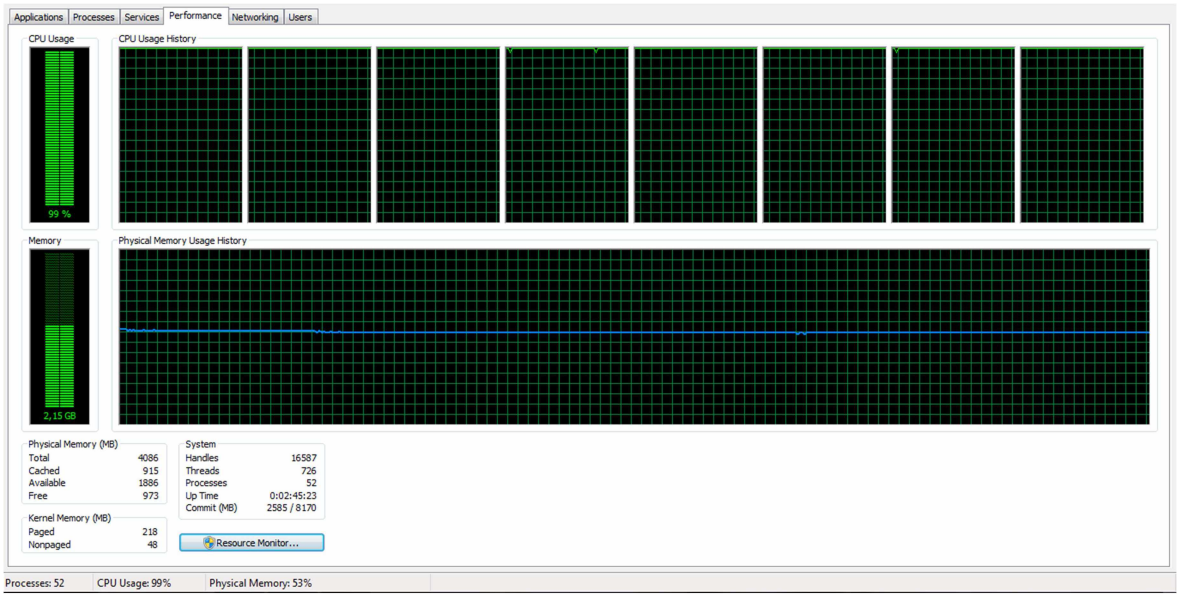


Figura B.7 – Software rodando em plataforma de 8 núcleos reais.

B.6 - CONCLUSÃO

Em geral, o processo de identificação de sistemas exige muito esforço computacional e, consequentemente, muito tempo. Neste sentido, a estratégia de Gerenciamento de Tarefas para a Execução de Tarefas em Modo Contínuo, proposta neste trabalho, fornece condições para um bom aproveitamento dos recursos de hardware disponível.

Além disso, esta estratégia pode ser empregada para aperfeiçoar outros algoritmos de identificação de sistemas em uso na atualidade.

Resultados em 4 tipos de plataformas foram apresentados confirmando o aspecto adaptativo do aplicativo na busca o melhor desempenho.

Para um trabalho futuro deixamos a proposta de que seja desenvolvido um dispositivo que avalie e aplique uma quantidade de pipelines necessária para o sistema de acordo com a necessidade da plataforma onde se esta em operação.

Outra proposta futura e adaptar esta metodologia para sistema do tipo cluster onde vários computadores com vários núcleos estarem disponíveis para o aplicativo de identificação.

ANEXO A

ALGORITMO DA TAXA DE REDUÇÃO DO ERRO (ERR)

SELEÇÃO DA ESTRUTURA

É fundamental para a representação global da dinâmica de um sistema a seleção adequada da estrutura (Aguirre, 1994). O processo de seleção pode ser realizado em duas partes:

- quais grupos de termos devem pertencer ao modelo e,
- quais termos, dentro destes grupos selecionados, farão parte do modelo.

Para uma adequada representação do sistema deve-se levar em conta o compromisso entre a escolha adequada dos grupos de termos (evitarão efeitos dinâmicos espúrios) e o número destes termos, que deve ser o menor possível para evitar problemas de instabilidade numérica (Aguirre e Billings, 1994). O problema que surge é justamente definir quantos termos e quais devem ser selecionados dentre os possíveis candidatos.

Existem dois grandes grupos de técnicas para seleção de termos, denominados construtivo e eliminativo (Kadtke et al., 1993; Aguirre, 1994).

- Escolhe-se um determinado modelo e a cada passo (step) um termo é inserido de acordo com sua importância na representação do sistema.
- Escolhe-se um "modelo de teste" e os termos menos significativos são descartados a cada passo, é preciso estimar os parâmetros iniciais do "modelo de teste", eliminando os termos menos significativos.

Dentre as técnicas utilizadas na determinação da estrutura será enfatizado nesta seção, o procedimento chamado ERR "Taxa de Redução de Erro" (Billings et al., 1989). O ERR consiste em associar a cada um dos termos candidatos, um índice correspondente à contribuição do mesmo na variância dos dados de saída. Desta forma, pode-se classificar os termos candidatos e gerar uma lista em ordem decrescente de acordo com a contribuição de cada um. O ERR é obtido usando as propriedades da fatoração ortogonal de matriz e será descrito mais adiante.

De posse do percentual de contribuição de cada termo deve-se definir o número de termos que será inserido no modelo. As estruturas de modelos NARMAX vêm sendo bastante estudadas, e vários artigos foram publicados com diferentes abordagens para o tema, o estudo dos efeitos da sobre parametrização (Aguirre e Billings, 1995) e de grupos de termos de pontos fixos de modelos polinomiais (Aguirre e Mendes, 1996).

ALGORITMO ORTOGONAL COM REGRESSÃO-DIRETA

Um algoritmo que combina de forma eficiente a seleção da estrutura ortogonal e a estimativa de parâmetros foi desenvolvido (Korenberg, 1985) e estendido para a identificação de sistemas SISO estocásticos (Korenberg et al. 1 988), posteriormente aplicados a sistemas MIMO (Billings et al., 1989).

Utilizando a representação matricial pode se escrever:

$$Y = P\Theta + \xi \qquad \text{(AN.1)}$$

Onde P é a matriz de regressores, Θ é a matriz que contém parâmetros a serem estimados e a matriz ξ contém os erros, a diferença entre o valor medido e o valor calculado. O vetor de parâmetros Θ é calculado de forma que a norma euclidiana

$\|Y - P\Theta\|_2$ seja minimizada, o que pode ser resolvido pelo método dos mínimos quadrados.

O método dos Mínimos Quadrados Ortogonal (OLS - "Orthogonal Least Squares") parte de vetores de base p_i e encontra um vetor solução $\hat{\Theta}$, que satisfaz a condição de $P\Theta$ projetar os valores de y no espaço dado pelos vetores base. O método OLS (Billings et al., 1988) envolve a transformação do conjunto de vetores p_i em um conjunto de vetores de base ortogonais, realizando cálculos para determinar a contribuição individual de cada vetor. A matriz de regressores P é decomposta como:

$$P = WA \tag{AN.2}$$

A é uma matriz triangular superior m x m com 1 na diagonal principal,

$$A = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1m} \\ 0 & 1 & \alpha_{23} & \cdots & \alpha_{2m} \\ 0 & 0 & 1 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \vdots & 1 \end{bmatrix} \tag{AN.3}$$

e W é uma matriz n x m com w_i colunas ortogonais sobre os dados. Então:

$$W^T W = D \tag{AN.4}$$

e D é diagonal com elementos d_i:

$$d_i = w_i^T w_i = \sum_{t=1}^{N} w_i(t) w_i(t), \quad 1 \le i \le m. \tag{AN.5}$$

O conjunto de vetores de base ortogonais w_i representa o mesmo espaço representado pelo conjunto p_i, podendo-se rescrever a equação (AN.1) como:

$$Y = W_g + E \tag{AN.6}$$

A solução OLS de $\hat{g}$ é dada por:

$$\hat{g} = \frac{w_i^T Y}{w_i^T w_i}, \quad 1 \le i \le m \tag{AN.7}$$

Os valores de $\hat{g}$ e $\hat{\Theta}$ satisfazem o sistema triangular:

$$A\hat{\Theta} = \hat{g} \tag{AN.8}$$

A partir da equação (AN.8) se pode obter os parâmetros do modelo original a partir dos parâmetros estimados de $\hat{g}$. Os métodos Gram-Shimidt Clássico e Gram-Shimidt Modificado podem ser usados para obter (AN.2) e então resolver para os mínimos quadrados calculando $\hat{\Theta}$ (Golub e Van Loan, 1989).

A variância da saída pode ser expressa por:

$$\frac{1}{N}Y^TY = \frac{1}{N}\sum_{i=1}^{M} g_i^2 w_i^T w_i + \frac{1}{N}E^TE \tag{AN.9}$$

Observe que $\sum_i g_i^2 w_i^T w_i / N$ é a parte da variância da saída calculada pelo i-ésimo regressor e que E^TE/N é a variância da saída $y(t)$ não calculada. Então $g_i^2 w_i^T w_i / N$ é a porção da variância de saída calculada por w_i. A taxa de redução de erro (*"error reduction ratio"*- ERR), devido à inclusão de w_i no modelo é definida como:

$$[ERR]_i = \frac{g_i^2 w_i^T w_i}{Y^TY}, \quad 1 \le i \le m \tag{AN.10}$$

A relação (AN.10) proporciona uma maneira simples e eficaz, para verificar a importância de cada regressor em calcular a variância da saída.

ANEXO B

MODELO RACIONAL LINEARIZADO NOS PARÂMETROS

DESCRIÇÃO ENTRADA-SAÍDA

Um modelo Entrada-Saída racional estocástico é definido pela razão de dois polinômios envolvendo o passado das entradas, saídas e o ruído (Billings e Chen, 1989):

$$y(t)=\frac{a\left(y(t-1),\ldots,y(t-r),u(t-1),\ldots,u(t-r),e(t-1),\ldots,e(t-r)\right)}{b\left(y(t-1),\ldots,y(t-r),u(t-1),\ldots,u(t-r),e(t-1),\ldots,e(t-r)\right)}+e(t) \quad \text{(BN.1)}$$

Onde $u(t)$ e $y(t)$ representam a entrada e a saída nos instantes discretos t (t=1, 2,...), respectivamente, e r é a ordem do modelo (supondo que todos os termos possuem o mesmo número de atraso) e $e(t)$ é um ruído independente não observável igualmente distribuído com média zero e variância finita σ_e^2.

Para simplificar a utilização da equação (BN.1) definem-se os polinômios do numerador e denominador, respectivamente como:

$$a(t)=\sum_{j=1}^{num} p_{nj}(t)\theta_{nj}, \quad \text{(BN.2)}$$

$$b(t)=\sum_{j=1}^{den} p_{dj}(t)\theta_{dj} \quad \text{(BN.3)}$$

Onde $p_{nj}(t)$ e $p_{dj}(t)$ são os coeficientes de $y(t-1),\ldots,y(t-r),u(t-1),\ldots,u(t-r),e(t-1),$ $\ldots,e(t-r)$ e o número total de parâmetros a ser estimado é $num+den$.

EXPRESSÃO LINEAR NOS PARÂMETROS

Infelizmente identificar um sistema a partir da equação (BN.1) é muito difícil, uma vez que esta é não linear nos parâmetros. Visando simplificar o problema pode-se multiplicar a equação (BN.1) por $b(t)$ em ambos os seus lados e a mover todos os termos para o lado direito com exceção de $y(t)p_{d1}(t)\theta_{d1}$ obtendo, como em (Zhu e Billings, 1991):

$$\begin{aligned} Y(t) &= a(t) - y(t)\sum_{j=2}^{den} p_{dj}(t)\theta_{dj} + b(t)e(t) \\ &= \sum_{j=1}^{num} p_{nj}(t)\theta_{nj} - \sum_{j=2}^{den} y(t)p_{dj}(t)\theta_{dj} + \zeta(t) \end{aligned} \tag{BN.4}$$

Onde

$$\begin{aligned} Y(t) &= y(t)p_{d1}\big|_{\theta_{d1}=1} \\ &= p_{d1}(t)\frac{a(t)}{b(t)} + p_{d1}(t)e(t) \end{aligned} \tag{BN.5}$$

E

$$\begin{aligned} \zeta(t) &= b(t)e(t) \\ &= \left(\sum_{j=2}^{den} p_{dj}(t)\theta_{dj}\right)e(t) \end{aligned} \tag{BN.6}$$

Onde $e(t)$ é um ruído branco como definido na equação (BN.1), e pelo fato de $e(t)$ ser independente de $b(t)$ e ter média nula, tem-se:

$$E\left[\zeta(t)\right] = E\left[b(t)\right]E\left[e(t)\right] = 0 \tag{BN.7}$$

A equação (BN.4) mostra que todos os termos $y(t)p_{dj}(t)$ incluem implicitamente o termo de ruído $e(t)$, através de $y(t)$, que é altamente correlacionado com $\zeta(t)$ o que resultará em polarização dos parâmetros mesmo se $e(t)$ for um ruído branco com média zero. O problema aparece quando se multiplica a equação (BN.1) por $b(t)$ a fim de tornar o modelo linear nos parâmetros.

A equação (BN.4) pode se expressa como:

$$\begin{aligned} Y(t) &= \sum_{j=1}^{num} p_{nj}(t)\theta_{nj} - \sum_{j=2}^{den} y(t)\, p_{dj}(t)\theta_{dj} + b(t)e(t) \\ &= \sum_{j=1}^{num} p_{nj}(t)\theta_{nj} - \sum_{j=2}^{den} \frac{a(t)}{b(t)} p_{dj}(t)\theta_{dj} + p_{d1}(t)e(t) \end{aligned} \tag{BN.8}$$

Em notação vetorial tem-se:

$$\begin{aligned} Y(t) &= \phi(t)\theta + \zeta(t) \\ &= \hat{\phi}(t)\theta + p_{d1}(t)e(t) \end{aligned} \tag{BN.9}$$

Onde

$$\begin{aligned} \phi(t) &= \left[\phi_n(t) \quad \phi_d(t)\right] \\ &= \left[p_{n1}(t) \quad \cdots \quad p_{nnum}(t) \quad -p_{d2}(t)y(t) \quad \cdots \quad -p_{dden}(t)y(t)\right] \\ &= \left[p_{n1}(t) \quad \cdots \quad p_{nnum}(t) \quad -p_{d2}(t)\left(\frac{a(t)}{b(t)} + e(t)\right) \quad \cdots \quad -p_{dden}(t)\left(\frac{a(t)}{b(t)} + e(t)\right)\right] \end{aligned} \tag{BN.10}$$

$$\begin{aligned} \theta^T &= \left[\theta_n \quad \theta_d\right] \\ &= \left[\theta_n \quad \cdots \quad \theta_{nnum} \quad \theta_{d2} \quad \cdots \quad \theta_{dden}\right] \end{aligned} \tag{BN.11}$$

E

$$\begin{aligned} \hat{\phi}(t) &= \left[\phi_n(t) \quad \hat{\phi}_d(t)\right] \\ &= \left[p_{n1}(t) \quad \cdots \quad p_{nnum}(t) \quad -p_{d2}(t)\left(\frac{a(t)}{b(t)} + e(t)\right) \quad \cdots \quad -p_{dden}(t)\left(\frac{a(t)}{b(t)} + e(t)\right)\right] \end{aligned} \tag{BN.12}$$

A matriz $\hat{\phi}(t)$ não pode ser obtida diretamente uma vez que $a(t)/b(t)$ não pode ser medida, enquanto que a matriz pode ser obtida totalmente a partir dos valores medidos de $y(t)$ e $u(t)$.

ANEXO C

ESTIMADOR DE MÍNIMOS QUADRADOS PARA O MODELO RACIONAL

ESTIMAÇÃO DE PARÂMETROS

O desenvolvimento de uma rotina de estimação de parâmetros utilizando mínimos quadrados (Billings e Zhu, 1991) é a razão primordial para se transformar o modelo racional em uma expressão linear nos parâmetros. Geralmente esta rotina é desenvolvida tendo sido considerada que a estrutura do modelo é conhecida a priori.

ESTIMADOR USANDO MÉTODO DOS MÍNIMOS QUADRADOS

Aplicando diretamente a equação dos mínimos quadrados tem-se:

$$\hat{\Theta} = \left[\phi^T \phi\right]^{-1} \phi^T Y \tag{CN.1}$$

Onde

$$\phi^T = \left[P(1)^T \quad \cdots \quad P(N)^T\right] =$$

$$= \begin{bmatrix} p_{n1}(1) & \cdots & p_{n1}(N) \\ \vdots & & \vdots \\ p_{nnum}(1) & \cdots & p_{nnum}(N) \\ -p_{d2}(1)\left(\dfrac{a(1)}{b(1)} + e(1)\right) & \cdots & p_{d2}(N)\left(\dfrac{a(N)}{b(N)} + e(N)\right) \\ \vdots & & \vdots \\ -p_{dden}(1)\left(\dfrac{a(1)}{b(1)} + e(1)\right) & \cdots & p_{dden}(N)\left(\dfrac{a(N)}{b(N)} + e(N)\right) \end{bmatrix} \tag{CN.2}$$

E

$$Y = \left[Y(1) \quad \cdots \quad Y(N)\right]^T \tag{CN.3}$$

Onde N indica o número de amostras e conforme (BN.1) ϕ pode incluir termos de ruído.

POLARIZAÇÃO DE $\hat{\Theta}$

Pode-se estimar os parâmetros de um modelo linear nos parâmetros do tipo $y(t) = p^T\Theta + e(t)$ a partir das equações (CN.4).

$$\begin{aligned} y &= P\Theta + e \\ \hat{\Theta} &= Ay \end{aligned} \tag{CN.4}$$

Onde P é a matriz dos regressores e A é uma matriz cujos elementos dependem dos regressores.

$$\begin{aligned} E[Ay] - \Theta = 0, \\ &= E\left[A(P\Theta + e)\right] - \Theta \\ &= E[AP - I]\Theta + E[Ae] \\ &= (E[AP] - I)\Theta + E[Ae] \end{aligned} \tag{CN.5}$$

Na equação (CN.5) a matriz Θ foi considerada determinística, nesta equação verifica-se que a polarização será nula se:

- $E[AP] = I$;
- Não houver correlação entre o ruído e os elementos de A;
- O ruído possuir média nula.

Analisando a equação para estimação de mínimos quadrados (CN.1) e a propriedade do limite da probabilidade (Wilks, 1962) para duas matrizes A e B, ambas funções das mesmas variáveis aleatórias, tem-se:

$$P\lim(A \quad B) = P\lim A \; P\lim B \tag{CN.6}$$

Considerando que a equação (CN.7) existe e aplicando a propriedade do limite da probabilidade obtém-se a equação (CN.8).

$$A \equiv P\lim\left[\frac{1}{N}\phi^T\phi\right]e$$
$$B \equiv P\lim\left[\frac{1}{N}\phi^T Y\right] \tag{CN.7}$$

$$P\lim\left[\hat{\Theta}\right] = P\lim\left[\left[\phi^T\phi\right]^{-1}\phi^T Y\right]$$
$$= P\lim\left[\phi^T\phi\right]^{-1} P\lim\left[\phi^T Y\right] \tag{CN.8}$$

Supondo-se que as sequências de entrada e de saída são fixas e que o comprimento dos dados N é suficientemente grande, tem-se:

$$P\lim\left[\frac{1}{N}\phi^T\phi\right] \approx \frac{1}{N}\phi^T\phi$$
$$P\lim\left[\frac{1}{N}\phi^T Y\right] \approx \frac{1}{N}\phi^T Y \tag{CN.9}$$

Onde:

$$\phi^T\phi = \begin{bmatrix} \sum_{k=1}^{N} p_{n1}^2(t) & \cdots & \sum_{k=1}^{N} p_{n1}(t)\,p_{nnum}(t) & \cdots & -\sum_{k=1}^{N} p_{n1}(t)\,p_{d2}(t)\frac{a(t)}{b(t)} & \cdots & -\sum_{k=1}^{N} p_{n1}(t)\,p_{dden}(t)\frac{a(t)}{b(t)} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \sum_{k=1}^{N} p_{nnum}(t)\,p_{n1}(t) & \cdots & \sum_{k=1}^{N} p_{nnum}^2(t) & \cdots & -\sum_{k=1}^{N} p_{nnum}(t)\,p_{d2}(t)\frac{a(t)}{b(t)} & \cdots & -\sum_{k=1}^{N} p_{nnum}(t)\,p_{dden}(t)\frac{a(t)}{b(t)} \\ -\sum_{k=1}^{N} p_{d2}(t)\,p_{n1}(t)\frac{a(t)}{b(t)} & \cdots & -\sum_{k=1}^{N} p_{d2}(t)\,p_{nnum}(t)\frac{a(t)}{b(t)} & \cdots & -\sum_{k=1}^{N} p_{d2}^2(t)\left(\frac{a(t)}{b(t)}\right)^2 & \cdots & -\sum_{k=1}^{N} p_{d2}(t)\,p_{dden}(t)\left(\frac{a(t)}{b(t)}\right)^2 \\ \vdots & & \vdots & & \vdots & & \vdots \\ -\sum_{k=1}^{N} p_{dden}(t)\,p_{n1}(t)\frac{a(t)}{b(t)} & \cdots & -\sum_{k=1}^{N} p_{dden}(t)\,p_{nnum}(t)\frac{a(t)}{b(t)} & \cdots & -\sum_{k=1}^{N} p_{dden}(t)\,p_{d2}(t)\left(\frac{a(t)}{b(t)}\right)^2 & \cdots & -\sum_{k=1}^{N} p_{dden}^2(t)\left(\frac{a(t)}{b(t)}\right)^2 \end{bmatrix} \tag{CN.10}$$

$$+\begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & \sigma_e^2\sum_{t=1}^{N} p_{d2}^2(t) & \cdots & \sigma_e^2\sum_{t=1}^{N} p_{d2}(t)\,p_{den}(t) \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & \sigma_e^2\sum_{t=1}^{N} p_{den}(t)\,p_{d2}(t) & \cdots & \sigma_e^2\sum_{t=1}^{N} p_{dden}^2(t) \end{bmatrix}$$

e

$$\phi^T Y = \begin{bmatrix} \sum_{t=1}^{N} p_{n1}(t)\, p_{d1}(t)\left(\frac{a(t)}{b(t)} + e(t)\right) \\ \vdots \\ \sum_{t=1}^{N} p_{nmum}(t)\, p_{d1}(t)\left(\frac{a(t)}{b(t)} + e(t)\right) \\ \sum_{t=1}^{N} p_{d2}(t)\, p_{d1}(t)\left(\frac{a(t)}{b(t)} + e(t)\right)^2 \\ \vdots \\ \sum_{t=1}^{N} p_{dden}(t)\, p_{d1}(t)\left(\frac{a(t)}{b(t)} + e(t)\right)^2 \end{bmatrix} = \begin{bmatrix} \sum_{t=1}^{N} p_{n1}(t)\, p_{d1}(t)\left(\frac{a(t)}{b(t)}\right) \\ \vdots \\ \sum_{t=1}^{N} p_{nmum}(t)\, p_{d1}(t)\left(\frac{a(t)}{b(t)}\right) \\ \sum_{t=1}^{N} p_{d2}(t)\, p_{d1}(t)\left(\frac{a(t)}{b(t)}\right)^2 \\ \vdots \\ \sum_{t=1}^{N} p_{dden}(t)\, p_{d1}(t)\left(\frac{a(t)}{b(t)}\right)^2 \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\sigma_e^2 \sum_{t=1}^{N} p_{d2}(t)\, p_{d1}(t) \\ \vdots \\ -\sigma_e^2 \sum_{t=1}^{N} p_{dden}(t)\, p_{d1}(t) \end{bmatrix} \quad \text{(CN.11)}$$

Para $N \to \infty$ a estimativa de $\hat{\Theta}$ converge em probabilidade para $\left[\phi^T \phi\right]^{-1} \phi^T Y$.

Rescrevendo a equação (CN.11) usando notação matricial, tem-se:

$$\begin{aligned} \phi^T \phi &= \left[\phi^T \phi\right]_{t-1} + \sigma_e^2 \Psi \\ \phi^T Y &= \left[\phi^T Y\right]_{t-1} + \sigma_e^2 \psi \end{aligned} \quad \text{(CN.12)}$$

Onde:

$$\Psi = \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & \sum_{t=1}^{N} p_{d2}^2(t) & \cdots & \sum_{t=1}^{N} p_{d2}(t)\, p_{den}(t) \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & \sum_{t=1}^{N} p_{den}(t)\, p_{d2}(t) & \cdots & \sum_{t=1}^{N} p_{dden}^2(t) \end{bmatrix} \quad \text{(CN.13)}$$

$$\psi = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\sum_{t=1}^{N} p_{d2}(t)\, p_{d1}(t) \\ \vdots \\ -\sum_{t=1}^{N} p_{dden}(t)\, p_{d1}(t) \end{bmatrix} \quad \text{(CN.14)}$$

Todos os termos envolvendo $e(t)$ aparecem em $\sigma_e^2\Psi$ e $\sigma_e^2\,\psi$, que são os termos de ruído. O subscrito $(t-1)$ indica que apenas termos de ruído até $(t-1)$ estão presentes (desde que $e(t-j),\ j\leq 1$) (Billings e Zhu, 1991).

O estimador de parâmetros dado em (CN.1) pode ser reescrito como:

$$\begin{aligned}\hat{\Theta} &= \left[\phi^T\phi\right]^{-1}\phi^T Y \\ &= \left[\left[\phi^T\phi\right]_{t-1}^{-1} + \sigma_e^2\Psi\right]^{-1}\left[\left[\phi^T Y\right]_{t-1} + \sigma_e^2\psi\right]\end{aligned} \tag{CN.15}$$

Comparando a equação (CN.15) com a (CN.5) verifica-se que $\left[\phi^T\phi\right]^{-1}\phi^T$ não pode estar correlacionado com o ruído para se evitar a polarização do estimador.

A equação (CN.15) mostra que linearização os parâmetros do modelo racional cria dois termos adicionais $\sigma_e^2\Psi$ e $\sigma_e^2\,\psi$, provando a correlação entre $\left[\phi^T\phi\right]^{-1}\phi^T$ e o ruído. Infelizmente essa correlação causa polarização na estimação dos parâmetros, mesmo para uma sequência $e(t)$ branca. O problema surge porque os termos $y(t)\,p_{dj}(t)$ em (BN.4) contêm implicitamente $e(t)$ (Billings e Zhu, 1991).

ESTIMADOR DE MÍNIMOS QUADRADOS PARA O MODELO RACIONAL – RME

O algoritmo de mínimos quadrados estendido para modelos com equações lineares diferenciais tem sido exaustivamente usado ao longo dos anos (Goodwin e Payne 1977; Norton 1986; Ljung 1987). Ponto comum em todos os algoritmos é que a correlação do vetor de resíduos é reduzida pela incorporação do modelo de ruído na matriz dos regressores.

Este procedimento é um método iterativo onde os parâmetros dos termos de processo e dos termos de ruído são estimados até a polarização ser reduzida a níveis

aceitáveis. Na Figura CN.1 tem-se o algoritmo sugerido em (Billings e Zhu, 1991), para implementação do método acima.

1º passo. Usar um algoritmo de mínimos quadrados ordinários (OLS) para computar:

$$\hat{\Theta} = \left[\phi^T \phi\right]^{-1} \phi^T Y$$

Esta estimação fornece os valores iniciais dos parâmetros para os cálculos seguintes.

2º passo. Calcular a sequência de resíduos

$$\xi(t) = y(t) - \frac{a(\ldots, \hat{\Theta}(i-1))}{b\left(\ldots, \hat{\Theta}(i-1)\right)}$$

Estimar a variância dos resíduos $\hat{\sigma}_\xi^2$ como

$$\hat{\sigma}_\xi^2(i) = \frac{1}{N - md} \sum_{t=md+1}^{N} \left(y(t) - \frac{a(\ldots, \hat{\Theta}(i-1))}{b\left(\ldots, \hat{\Theta}(i-1)\right)} \right)^2$$

Onde i é o número da iteração, N o número de amostras e *md* é o máximo atraso presente nos termos.

3º passo. Utilizando as equações (CN.10), (CN.11), (CN13) e (CN14), atualize as matrizes $\phi^T\phi$ e Ψ, vetores $\phi^T Y$ e ψ usando a sequência de resíduos calculada no 2º passo. Note que em tais matrizes os resíduos podem aparecer dentro dos polinômios $a(t)$ e $b(t)$.

4º passo. Calcule os novos parâmetros como

$$\hat{\Theta}(i) = \left[\phi^T \phi - \sigma_\xi^2(i)\Psi\right]^{-1} \left[\phi^T Y - \sigma_\xi^2(i)\psi\right]$$

5ª passo. Retorne ao 2º passo e repita até os parâmetros estimados e $\hat{\sigma}_\xi^2$ convergir em um valor constante.

Figura CN.1 - Algoritmo de Mínimos Quadrados Estendido proposto em (Billings e Zhu, 1991).

ANEXO D

ESTIMADOR ESTENDIDO DOS MÍNIMOS QUADRADOS

Quando se trabalha com os algoritmos de mínimos quadrados, observa-se que existe um ponto comum em todos os algoritmos, em que a correlação do vetor de resíduos é reduzida através da incorporação de um modelo de ruído na matriz de regressores. Tal procedimento é desenvolvido de forma interativa, onde os parâmetros dos termos e os parâmetros dos ruídos são estimados até que sua polarização seja reduzida em níveis aceitáveis.

Em (Aguirre L. A., 2007) tem-se que uma das formas de se evitar a polarização, utilizada pelo estimador estendido de mínimos quadrados, EMQ, é fazer a transformação da equação matricial $y = \Psi\theta + e$ na equação (AN.1):

$$y^* = \Psi^* \theta^* + e^* \tag{DN.1}$$

Onde e* um ruído branco, em que $E[A^* e^*] = 0$ e $A^* \Psi^* = I$.

Assim, para aplicações em estruturas matemáticas lineares, tais como o ARX, o desenvolvimento para a remoção da polarização é feito da seguinte forma: seja o modelo ARX descrito pela Equação (DN.2):

$$\begin{aligned} y(k) &= ay(k-1) + bu(k-1) + e(k) \\ e(k) &= cv(k-1) + v(k) \end{aligned} \tag{DN.2}$$

e $v(k)$ é considerado como um ruído branco. Assim, realizando as substituições na Equação (DN.2), tem-se a Equação (DN.3):

$$\begin{aligned} &y(k) = ay(k-1) + bu(k-1) + cv(k-1) + v(k), \\ &y(k-1) = ay(k-2) + bu(k-2) + cv(k-2) + v(k-1) \end{aligned} \tag{DN.3}$$

Agora substituindo em y(k) a equação de y(k-1) temos a equação (DN.4):

$$\begin{aligned} y(k) &= a[ay(k-2)+bu(k-2)+cv(k-2)+v(k-1)] \\ &+bu(k-1)+cv(k-1)+v(k) \end{aligned} \tag{DN.4}$$

Passando para a forma matricial obtemos as Equações (DN.5), (DN.6) e (DN.7):

$$\psi^T(k-1) = \left[\left(ay(k-2)+bu(k-2)+cv(k-2)+v(k-1)\right) \quad u(k-1)\right] \tag{DN.5}$$

$$\theta = \begin{bmatrix} a & b \end{bmatrix}^T \tag{DN.6}$$

$$e(k) = cv(k-1)+v(k) \tag{DN.7}$$

Através do EQM, se consegue eliminar a polarização. Seja a Equação (DN.8) abaixo:

$$\begin{aligned} y(k) &= a\left[ay(k-2)+bu(k-2)+cv(k-2)+v(k-1)\right] \\ &+bu(k-1)+cv(k-1)+v(k) \end{aligned} \tag{DN.8}$$

Podendo esta ser escrita na forma matricial, conforme a Equação (DN.9):

$$\begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+N-1) \end{bmatrix} = \begin{bmatrix} [y(k-1)] & u(k-1) & v(k-1) \\ [y(k)] & u(k) & v(k) \\ \vdots & \vdots & \vdots \\ [y(k+N-2] & u(k+N-2) & v(k+N-2) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} + \begin{bmatrix} v(k) \\ v(k+1) \\ \vdots \\ v(k+N-1) \end{bmatrix} \tag{DN.9}$$

isto é:

$$y^* = \Psi^* \theta^* + e^* \tag{DN.10}$$

sendo que:

$$y^* = y, e^* = \begin{bmatrix} v(k) & \text{K} & v(k+N-1) \end{bmatrix}^T \tag{DN.11}$$

$$\Psi^* = \begin{bmatrix} & \text{M} & v(k-1) \\ & \text{M} & v(k) \\ \Psi & \text{M} & v(k+1 \\ & \text{M} & \text{M} \\ & \text{M} & v(k+N-2) \end{bmatrix} \tag{DN.12}$$

$$\theta^* = \begin{bmatrix} \theta^T & \text{M} \; c \end{bmatrix}^T \tag{DN.13}$$

É importante observar que $e^*(k)$ é uma variável aleatória "branca", definida como:

$$r_{e^*e^*}(k) = 0, \forall k \neq 0 \tag{DN.14}$$

e consequentemente $E\left[\Psi e^*\right] = 0, E\left[A^* e^*\right] = 0, E\left[\Psi^* e^*\right] = 0$.

Assim, a estimativa é calculada conforme a Equação (DN.15):

$$\hat{\theta}^* = \left[\Psi^{*T}\Psi^*\right]^{-1}\Psi^{*T} y \tag{DN.15}$$

e não apresenta a polarização, definida como:

$$E\left[\hat{\theta}^*\right] = \left[\theta^T \quad c\right]^T \tag{DN.16}$$

O algoritmo EQM pode ser caracterizado pelos seguintes passos: (LJUNG, 1999):

1. Dada a equação de regressão $y(k) = \varphi^T(k-1)\theta + e(k)$ e os dados disponíveis, provenientes dos ensaios experimentais, monte a equação matricial $y = \Psi\theta + e$, definido no MQ, e determine a matriz com os parâmetros estimados $\hat{\theta}_{MQ} = \left[\Psi^T \quad \Psi\right]^{-1}\Psi^T y$;

2. A partir da saída gerada pelo modelo, calcule os resíduos $\xi = y - \Psi\,\hat{\theta}_{MQ}$;

3. Inicialize $i = 2$, onde i representa o número de interações;

4. Com o valor dos resíduos calculados, monte a matriz estendida de regressores, Ψ_i^*, e estime os parâmetros da equação desejada $\hat{\theta}^*_{EQMi} = \left[\Psi_i^* \quad \Psi_i^*\right]^{-1}\Psi_i^{*T} y$;

5. Determine novamente os valores dos resíduos $\xi = y - \Psi\,\hat{\theta}_{MQ}$;

6. Faça $i = i + 1$ e volte para o passo 4. Repita os passos até que a resposta convirja.