

RAFAEL BARTNIK GREBOGI

MÁQUINA DE APRENDIZAGEM EXTREMA COM OTIMIZAÇÃO POR ENXAME
DE PARTÍCULAS APLICADA À PREVISÃO DE SÉRIES TEMPORAIS

CURITIBA

2013

RAFAEL BARTNIK GREBOGI

MÁQUINA DE APRENDIZAGEM EXTREMA COM OTIMIZAÇÃO POR ENXAME
DE PARTÍCULAS APLICADA À PREVISÃO DE SÉRIES TEMPORAIS

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Engenharia Elétrica, pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Paraná.

Orientador: Prof. Dr. Leandro dos Santos Coelho

CURITIBA

2013

Termo de Aprovação

MÁQUINA DE APRENDIZAGEM EXTREMA COM OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS APLICADA À PREVISÃO DE SÉRIES TEMPORAIS

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica, pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Paraná, pela seguinte banca examinadora:

Prof. Dr. Eduardo Parente Ribeiro
Universidade Federal do Paraná

Prof. Dr. Gideon Villar Leandro
Universidade Federal do Paraná

Prof. Dr. Leandro dos Santos Coelho
Universidade Federal do Paraná

Prof^a. Dr^a. Maria Teresinha Arns Steiner
Pontifícia Universidade Católica do Paraná

Curitiba, 26 de fevereiro de 2013.

DEDICATÓRIA

Eu dedico este trabalho a duas pessoas que foram muito importantes em minha vida, as quais eu sempre guardarei as melhores e mais felizes memórias.

Vó Maria e Vó Eva, este trabalho é dedicado a vocês.

Sumário

LISTA DE FIGURAS.....	vi
LISTA DE TABELAS.....	xiii
LISTA DE SIGLAS.....	xv
RESUMO.....	xvi
ABSTRACT.....	xvii
1. INTRODUÇÃO	1
1.1. OBJETIVOS	2
1.2. JUSTIFICATIVA	3
1.3. ORGANIZAÇÃO DO TRABALHO	3
2. SÉRIES TEMPORAIS	5
2.1. SÉRIE TEMPORAL DE BOX & JENKINS	5
2.2. SÉRIE TEMPORAL DE MACKEY-GLASS.....	7
2.3. SISTEMA DE LORENZ	11
3. IDENTIFICAÇÃO DE SISTEMAS	14
3.1. REPRESENTAÇÕES DE MODELOS	15
3.1.1. MODELO DE RESPOSTA AO IMPULSO.....	16
3.1.2. MODELO ARX.....	17
3.1.3. MODELO NARX	18
3.1.4. MODELO ARMAX/NARMAX	18
3.1.5. MODELO DE <i>HAMMERSTEIN / WIENER</i>	19
3.2. REDES NEURAS ARTIFICIAIS	20
3.2.1. ALGORITMO DE APRENDIZAGEM EXTREMA	25
3.3. VALIDAÇÃO DE MODELOS.....	30
4. OTIMIZAÇÃO	37

4.1.	ALGORITMO DE ENXAME DE PARTÍCULAS	41
4.1.1.	ALGORITMO DE ENXAME DE PARTÍCULAS DINÂMICO	45
4.1.2.	ALGORITMO DE ENXAME DE PARTÍCULAS COM FUNÇÃO GAMA	46
5.	RESULTADOS	50
5.1.	BOX & JENKINS	54
5.1.1.	ESTUDO DE CASO ORIGINAL BOX & JENKINS.....	55
5.1.2.	ESTUDO DE CASO MODIFICADO BOX & JENKINS	63
5.2.	MACKEY-GLASS	67
5.2.1.	ESTUDO DE CASO ORIGINAL MACKEY-GLASS	67
5.2.2.	ESTUDO DE CASO MODIFICADO MACKEY-GLASS.....	73
5.3.	SISTEMA DE LORENZ	76
5.3.1.	ESTUDO DE CASO ORIGINAL SISTEMA DE LORENZ	77
5.3.2.	ESTUDO DE CASO MODIFICADO SISTEMA DE LORENZ.....	82
6.	CONCLUSÕES E SUGESTÕES PARA FUTUROS TRABALHOS	86
	REFERÊNCIAS	89
	ANEXO A	97
	ANEXO B	100

Lista de Figuras

Figura 1: Sinal de entrada (taxa de vazão) da fornalha.....	6
Figura 2: Sinal de saída (% CO ₂) da fornalha.	7
Figura 3: Trajetória de dois sistemas divergindo. Adaptada de [15].....	9
Figura 4: Série de Mackey-Glass com $\tau = 17$ e $x_0 = 0,24$	10
Figura 5: Série de Mackey-Glass com $\tau = 34$ e $x_0 = 0,68$	10
Figura 6: Gráfico tridimensional do sistema de equações diferenciais ordinárias de Lorenz.....	12
Figura 7: Série de Lorenz na coordenada x	13
Figura 8: Modelo de <i>Hammerstein</i>	20
Figura 9: Descrição das partes de um neurônio de um animal vertebrado. Adaptada de [1].	21
Figura 10: Modelo de neurônio de McCulloch e Pitts. Adaptada de [1].	22
Figura 11: Resposta da função SIG.	29
Figura 12: Ilustração mostrando o espaço de procura S e o espaço possível de procura F . Adaptada de [37].	39
Figura 13: Mínimos de uma função.	40
Figura 14: Movimento das partículas em direção à melhor solução.....	42
Figura 15: Topologia de estrela do PSO.	45
Figura 16: Exemplo de peso de inércia obtido através da equação (4.7) com os parâmetros $\omega_1 = 0,9$, $\omega_2 = 0,4$ e $\omega_e = 700$	46
Figura 17: Curvas da função gama incompleta original e ajustada.	48
Figura 18: Curva de convergência do algoritmo DPSO para a série de Box & Jenkins.	57
Figura 19: Previsão e sinal de erro para o conjunto de treinamento da série de Box & Jenkins utilizando a função de ativação TAF.....	58
Figura 20: Previsão e sinal de erro para o conjunto de validação da série de Box & Jenkins com a função de ativação TAF.	59
Figura 21: Previsão e sinal de erro da previsão do tipo N -passos-a-frente para a série de Box & Jenkins.	60
Figura 22: Previsão e sinal de erro utilizando o método de retropropagação para a série de Box & Jenkins.	62

Figura 23: Previsão e sinal de erro utilizando o algoritmo ELM original para a série de Box & Jenkins.	62
Figura 24: Previsão e sinal de erro para o conjunto de treinamento da série Box & Jenkins empregando as novas entradas.	65
Figura 25: Previsão e sinal de erro para o conjunto de validação da série Box & Jenkins empregando as novas entradas.....	65
Figura 26: Previsão do tipo N -passos-a-frente e sinal de erro utilizando o modelo com as novas entradas para a série Box & Jenkins.	66
Figura 27: Curvas de convergência média e do melhor experimento do algoritmo GPSO para a série de Mackey-Glass empregando a função TAF.	69
Figura 28: Previsão e sinal de erro utilizando a função TAF para o conjunto de treinamento da série de Mackey-Glass.	70
Figura 29: Previsão e sinal de erro empregando a função TAF para o conjunto de validação da série de Mackey-Glass.	71
Figura 30: Previsão do tipo N -passos-a-frente e sinal de erro utilizando a função TAF e o conjunto de dados de validação da série de Mackey-Glass.	71
Figura 31: Previsão e sinal de erro utilizando a função TAF para o conjunto de treinamento da série de Mackey-Glass com as novas entradas.	74
Figura 32: Previsão e sinal de erro empregando a função TAF para o conjunto de validação da série de Mackey-Glass com as novas entradas.	75
Figura 33: Previsão do tipo N -passos-a-frente e sinal de erro utilizando a função TAF para o conjunto de validação da série de Mackey-Glass com as novas entradas.....	75
Figura 34: Curvas de convergência média e da melhor solução obtida pelo algoritmo GPSO para a série de Lorenz.....	78
Figura 35: Previsão e sinal de erro utilizando a função TAF para o conjunto de dados de treinamento da série de Lorenz.	79
Figura 36: Previsão e sinal de erro utilizando a função TAF para o conjunto de dados de validação da série de Lorenz.....	80
Figura 37: Previsão do tipo N -passos-a-frente e sinal de erro empregando a função TAF para o conjunto de validação da série de Lorenz.....	81
Figura 38: Previsão e sinal de erro utilizando a função TAF para o conjunto de dados de treinamento da série de Lorenz.	84

Figura 39: Previsão e sinal de erro utilizando a função TAF para o conjunto de dados de validação da série de Lorenz.....	84
Figura 40: Previsão do tipo N -passos-a-frente e sinal de erro utilizando a função TAF para o conjunto de dados de validação para a série de Lorenz.....	85
Figura 41: Gráficos de autocorrelação, autocorrelação de alta ordem e correlação cruzada de alta ordem entre a saída com o resíduo e o resíduo, para a série de Box & Jenkins com as entradas originais.	100
Figura 42: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 1)$ da série de Box & Jenkins com as entradas originais.	101
Figura 43: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 2)$ da série de Box & Jenkins com as entradas originais.	101
Figura 44: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 3)$ da série de Box & Jenkins com as entradas originais.	102
Figura 45: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 1)$ da série de Box & Jenkins com as entradas originais.	102
Figura 46: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 2)$ da série de Box & Jenkins com as entradas originais.	103
Figura 47: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 3)$ da série de Box & Jenkins com as entradas originais.	103
Figura 48: Gráficos de autocorrelação, autocorrelação de alta ordem e correlação cruzada de alta ordem entre a saída com o resíduo e o resíduo, para a série de Box & Jenkins com as novas entradas.	104
Figura 49: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 1)$ da série de Box & Jenkins com as novas entradas.	104

Figura 50: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 2)$ da série de Box & Jenkins com as novas entradas.	105
Figura 51: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 3)$ da série de Box & Jenkins com as novas entradas.	105
Figura 52: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 1)x(k - 1)$ da série de Box & Jenkins com as novas entradas.	106
Figura 53: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 1)x(k - 2)$ da série de Box & Jenkins com as novas entradas.	106
Figura 54: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 1)$ da série de Box & Jenkins com as novas entradas.	107
Figura 55: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 2)$ da série de Box & Jenkins com as novas entradas.	107
Figura 56: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 3)$ da série de Box & Jenkins com as novas entradas.	108
Figura 57: Gráficos de autocorrelação, autocorrelação de alta ordem e correlação cruzada de alta ordem entre a saída com o resíduo e o resíduo, para a série de Mackey-Glass com as entradas originais.	108
Figura 58: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k)$ da série de Mackey-Glass com as entradas originais.	109
Figura 59: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 6)$ da série de Mackey-Glass com as entradas originais.	109
Figura 60: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 12)$ da série de Mackey-Glass com as entradas originais.	110

Figura 61: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 18)$ da série de Mackey-Glass com as entradas originais.....	110
Figura 62: Gráficos de autocorrelação, autocorrelação de alta ordem e correlação cruzada de alta ordem entre a saída com o resíduo e o resíduo, para a série de Mackey-Glass com as novas entradas adicionadas.....	111
Figura 63: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k)$ da série de Mackey-Glass com as novas entradas adicionadas.....	111
Figura 64: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 6)$ da série de Mackey-Glass com as novas entradas adicionadas.....	112
Figura 65: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 12)$ da série de Mackey-Glass com as novas entradas adicionadas.....	112
Figura 66: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 18)$ da série de Mackey-Glass com as novas entradas adicionadas.....	113
Figura 67: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 11)$ da série de Mackey-Glass com as novas entradas adicionadas.....	113
Figura 68: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $e(k - 5)$ da série de Mackey-Glass com as novas entradas adicionadas.....	114
Figura 69: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $e(k - 6)x(k - 6)$ da série de Mackey-Glass com as novas entradas adicionadas.....	114
Figura 70: Gráficos de autocorrelação, autocorrelação de alta ordem e correlação cruzada de alta ordem entre a saída com o resíduo e o resíduo, para a série de Lorenz com as entradas originais.....	115
Figura 71: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k)$ da série de Lorenz com as entradas originais.....	115

Figura 72: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 8)$ da série de Lorenz com as entradas originais.	116
Figura 73: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 16)$ da série de Lorenz com as entradas originais.	116
Figura 74: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 24)$ da série de Lorenz com as entradas originais.	117
Figura 75: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 32)$ da série de Lorenz com as entradas originais.	117
Figura 76: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 40)$ da série de Lorenz com as entradas originais.	118
Figura 77: Gráficos de autocorrelação, autocorrelação de alta ordem e correlação cruzada de alta ordem entre a saída com o resíduo e o resíduo, para a série de Lorenz com as novas entradas adicionadas.....	118
Figura 78: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k)$ da série de Lorenz com as novas entradas adicionadas.	119
Figura 79: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 8)$ da série de Lorenz com as novas entradas adicionadas.	119
Figura 80: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 16)$ da série de Lorenz com as novas entradas adicionadas.	120
Figura 81: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 24)$ da série de Lorenz com as novas entradas adicionadas.	120
Figura 82: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 32)$ da série de Lorenz com as novas entradas adicionadas.	121
Figura 83: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 40)$ da série de Lorenz com as novas entradas adicionadas.	121

Figura 84: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $e(k - 1)$ da série de Lorenz com as novas entradas adicionadas.	122
Figura 85: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $e(k - 8)$ da série de Lorenz com as novas entradas adicionadas.	122
Figura 86: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $e(k - 5)$ da série de Lorenz com as novas entradas adicionadas.	123
Figura 87: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $e(k - 1)e(k - 8)^3$ da série de Lorenz com as novas entradas adicionadas.	123
Figura 88: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 1)e(k - 8)$ da série de Lorenz com as novas entradas adicionadas.	124

Lista de Tabelas

Tabela 1: Parâmetros da série Mackey-Glass e respectivos expoentes de Lyapunov.....	9
Tabela 2: Expoentes de Lyapunov para cada dimensão da série de Lorenz.	11
Tabela 3: Tipos de função de ativação.....	23
Tabela 4: Testes de correlação a serem aplicados.	35
Tabela 5: Teste de K para diferentes números de neurônios ocultos.	53
Tabela 6: Número de neurônios lineares e não-lineares por camada de neurônios ocultos.	54
Tabela 7: Resultados de previsão da série Box & Jenkins com a função SIG.	55
Tabela 8: Resultado para a série Box & Jenkins utilizando a função de ativação TAF.	56
Tabela 9: Dados estatísticos para simulação com a série de Box & Jenkins e função TAF com as entradas originais.	57
Tabela 10: Dados estatísticos referentes à previsão do tipo N -passos-a-frente para a série de Box & Jenkins.....	60
Tabela 11: Comparação entre o algoritmo ELM original, método de retropropagação e ELM otimizado pelo algoritmo PSO, para a série de Box & Jenkins.	63
Tabela 12: Dados estatísticos referentes aos experimentos com as novas entradas para a série de Box & Jenkins utilizando a função TAF.	64
Tabela 13: Dados estatísticos referentes à previsão N -passos-a-frente utilizando o modelo com as novas entradas para a série de Box & Jenkins.	67
Tabela 14: Resultado da otimização para a série Mackey-Glass e função SIG.....	67
Tabela 15: Resultados das otimizações para a série de Mackey-Glass com a função TAF.....	68
Tabela 16: Dados estatísticos referentes à melhor simulação para a série de Mackey-Glass utilizando a função TAF.	69
Tabela 17: Dados estatísticos referentes à previsão do tipo N -passos-a-frente para a série de Mackey-Glass.....	72
Tabela 18: Dados estatísticos referentes à simulação com as novas entradas para a série de Mackey-Glass.....	73

Tabela 19: Dados estatísticos referentes à previsão do tipo N -passos-a-frente da série de Mackey-Glass utilizando as novas entradas.....	76
Tabela 20: Valores de <i>fitness</i> para a série de Lorenz com função SIG.....	77
Tabela 21: Valores de <i>fitness</i> dos experimentos com a série de Lorenz e função TAF.....	77
Tabela 22: Dados estatísticos referentes à melhor solução com a função TAF para a série de Lorenz.....	79
Tabela 23: Dados estatísticos referentes à previsão do tipo N -passos-a-frente.	81
Tabela 24: Dados estatísticos referentes ao melhor experimento com as novas entradas para a série de Lorenz.	83
Tabela 25: Dados estatísticos referentes à previsão do tipo N -passos-a-frente para a série de Lorenz com as novas entradas.....	85

Lista de Siglas

ARMAX	<i>Autoregressive Moving Average with Exogenous Inputs</i>
ARX	<i>Autoregressive with Exogenous inputs</i>
DPSO	<i>Dynamic Particle Swarm Optimization</i>
ELM	<i>Extreme Learning Machine</i>
EELM	<i>Effective Extreme Learning Machine</i>
E-ELM	<i>Evolutionary Extreme Learning Machine</i>
GPSO	<i>Gamma Particle Swarm Optimization</i>
MSE	<i>Mean Sum of Squares Error</i>
MSEW	<i>Mean Sum of Squares Error and Weight</i>
NARMA	<i>Nonlinear Autoregressive Moving Average</i>
NARMAX	<i>Nonlinear Autoregressive Moving Average with Exogenous Inputs</i>
PSO	<i>Particle Swarm Optimization</i>
SIG	Sigmóide
SLFN	<i>Single Layer Feedforward Neural Network</i>
TAF	<i>Tunable Activation Function</i>

Resumo

Identificação de sistemas é uma área interessada em obter modelos matemáticos de sistemas desconhecidos baseados em dados de leituras sequenciais do sistema. Diversas aplicações do mundo real não tem sua dinâmica completamente compreendida ou são complexas para serem modeladas, para estes casos, a identificação de sistemas é uma ferramenta eficiente para modelagem e previsão. Este trabalho aborda redes neurais artificiais, mais precisamente, redes neurais com uma única camada de neurônios ocultos, em inglês, *Single Layer Feedforward Neural Network* (SLFN), para previsão de séries temporais. Um algoritmo de aprendizagem proposto recentemente chamado de Máquina de Aprendizagem Extrema, em inglês, *Extreme Learning Machine* (ELM), é introduzido para a tarefa de aprendizagem da rede neural. O algoritmo ELM é baseado na matriz inversa generalizada de Moore-Penrose, que torna o problema um simples sistema linear. No núcleo do algoritmo ELM, duas funções de ativação diferentes serão testadas, sendo que uma delas é uma função de ativação variável. Para alcançar melhores resultados, um método estocástico de otimização do campo da inteligência de enxame chamado de Otimização por Enxame de Partículas, em inglês, *Particle Swarm Optimization* (PSO), é validado para otimizar os parâmetros do algoritmo ELM. O PSO consiste em modelar as ações de um bando de pássaros procurando por comida, onde cada pássaro é uma partícula, e cada partícula é uma possível solução para o problema. Neste trabalho é proposta uma nova variação do PSO empregando a função gama invertida. Neste contexto, três conjuntos de dados são usados para testar os algoritmos, um é a leitura de uma fornalha, e dois são obtidos de equações diferenciais com comportamento caótico. Os modelos obtidos através do algoritmo ELM são então validados através de testes de correlação. As previsões realizadas pelo algoritmo ELM são promissoras para todos os conjuntos de dados, revelando que a combinação do algoritmo PSO com o ELM é uma eficiente forma de identificação de sistemas.

Palavras-chave: rede neural, máquina de aprendizagem extrema, inteligência de enxame, identificação de sistemas, função gama incompleta.

Abstract

System identification is a subject concerned about obtaining mathematical models of unknown systems based on sequential systems' data readings. Many real-world applications do not have their dynamics well understood yet or are complex for modeling, for these cases, system identification is an efficient tool for modeling and forecasting. This work approaches artificial neural networks, more precisely, the Single Layer Feedforward Neural Network (SLFN) for time series forecasting. A recently proposed learning algorithm called Extreme Learning Machine (ELM) is introduced for the neural network learning task. The ELM algorithm is based on the Moore-Penrose generalized inverse of a matrix, turning the problem into a simple linear system. In the core of ELM algorithm, two different activation functions will be evaluated, where one is a variable activation function. To reach better results, a stochastic optimization method of the swarm intelligence field called Particle Swarm Optimization (PSO) is validated to optimize ELM's parameters. PSO consists of modeling the actions of a flock of birds looking for food, where each bird is a particle and each particle is a possible solution for the problem. This work proposes a modified PSO based on the incomplete gamma function. In this context, three datasets are used for testing, one is from a gas furnace, and two are obtained from differential equations with chaotic behavior. After the optimization, model verification is realized by correlation tests. The ELM's forecasting results are promising for all datasets, revealing that PSO combined with ELM is an efficient way for the task of the system identification.

Keywords: neural network, extreme learning machine, swarm intelligence, system identification, incomplete gamma function.

1. INTRODUÇÃO

A necessidade do ser humano em tentar antever eventos remonta dos tempos mais longínquos, quando pessoas chamadas muitas vezes de “magos”, por supostamente serem dotados de poderes divinos, tentavam prever catástrofes ou boas safras para a agricultura, acontecimentos importantes, etc., baseando-se em observações astronômicas, eventos cotidianos, os quais eles julgavam serem relevantes, e também por experiências vivenciadas, sendo que na maioria das vezes não obtinham sucesso na previsão, ou quando obtinham, seria por pura coincidência. Esta necessidade perdura até os tempos atuais, onde cartomantes tentam predizer o futuro a partir da leitura de cartas, por exemplo.

Já no âmbito científico, nos anos de 1950 surgiam os primeiros esforços na tentativa de prever acontecimentos futuros, mas agora em sistemas, como por exemplo, na tentativa de prever o clima, em problemas de classificação e regressão, com algo em comum a todos, a necessidade da obtenção de um modelo.

Nesta época começou o desenvolvimento das redes neurais, com os primeiros trabalhos de Rosenblatt, McCulloch e Pitts, com o objetivo de desenvolver um modelo para problemas de classificação [1].

A utilização de redes neurais artificiais se tornou uma ferramenta cada vez mais utilizada com o aumento da capacidade de processamento dos computadores nas últimas décadas. Inicialmente desacreditada devido ao alto custo computacional que envolve o treinamento supervisionado, durante a década de 1980 ela começou a se destacar e continua exercendo um papel importante em problemas de classificação, regressão e agrupamento de dados (*clustering*). Seu sucesso é devido a sua flexibilidade para mapeamentos multidimensionais não-lineares de entrada e saída, em conjunto com sua capacidade de aproximação universal [1][2].

Entretanto, diversos desafios envolvem o processo de treinamento supervisionado de uma rede, como a utilização de conjuntos de dados observados limitados e normalmente ruidosos para o treinamento pode fazer com que a rede se torne especializada unicamente para aquele conjunto de dados, perdendo a

flexibilidade, ou seja, a capacidade de generalização do modelo, problema esse conhecido como sobre-treinamento, em inglês, *overtraining*.

Atualmente existe uma necessidade crescente de previsão de processos nas mais diversas áreas, onde as redes neurais atuam com sucesso, desde negócios, criando modelos que atuem na cadeia de suprimentos, com o objetivo de evitar a falta de matéria-prima na indústria, abastecendo-a antecipadamente, na meteorologia, dando um prognóstico do clima, na previsão dos volumes de precipitações e vazões de rios, na biologia, prevendo reações de um organismo a certos medicamentos, como também no auxílio à tomada de decisões financeiras, como na compra e venda de ações [3]. Processos dinâmicos e com alto grau de não-linearidade também são abordados pelas redes neurais, como por exemplo, sistemas caóticos.

1.1.OBJETIVOS

Este trabalho têm como objetivos:

- utilizar uma forma mais eficiente e simples de treinamento de redes neurais com uma única camada de neurônios ocultos, chamado de Máquina de Aprendizagem Extrema. Esta eficiência é obtida através da diminuição do custo computacional despendido para sintonia das ponderações da rede a partir do uso da matriz inversa generalizada de Moore-Penrose;
- otimizar as ponderações e parâmetros através de um método estocástico de otimização, chamado de Otimização por Enxame de Partículas. Ele tem a finalidade de melhorar o desempenho da rede neural com uma única camada de neurônios ocultos, e do algoritmo de aprendizagem;
- implementar uma variação do método de otimização, com o objetivo de melhorar o desempenho do método original;
- identificar e prever três séries temporais, sendo que uma delas é um problema de identificação obtido do mundo real, e as duas séries

restantes geradas a partir de equações diferenciais com comportamento caótico.

- implementar uma variação da função de ativação da rede neural, com a finalidade de aprimorar seu desempenho.

1.2. JUSTIFICATIVA

A justificativa deste trabalho se baseia na baixa eficiência de métodos de treinamento de redes neurais já consagrados, como o método de retropropagação que tem um alto custo computacional. Em aplicações no mundo real, o tempo despendido durante o treinamento da rede pode ser um fator crítico.

Portanto, a implementação de um método mais eficiente de treinamento, que produza soluções de melhor qualidade necessitando de menos processamento é uma vantagem do método chamado de máquina de aprendizagem extrema. Juntamente com os métodos de otimização estocástica clássico e o proposto pelo autor, onde ambos são aplicados à busca de parâmetros ótimos de funcionamento da rede. A obtenção de modelos mais flexíveis pode evitar problemas de excesso de treinamento e consequente perda de generalização, melhorando o desempenho final do modelo encontrado pela rede neural.

1.3. ORGANIZAÇÃO DO TRABALHO

O restante do trabalho está organizado da seguinte forma. No Capítulo 2 são descritos os estudos de caso, explicando inicialmente o conceito de séries temporais e posteriormente introduzindo cada um dos estudos de caso. O Capítulo 3 aborda a identificação de sistemas, em especial, a representação de modelos, a aplicação de redes neurais, o conceito da máquina de aprendizagem extrema, e alguns testes de validação de modelos. O Capítulo 4 é sobre otimização das soluções de problemas, onde o algoritmo de otimização é descrito e sua variação é proposta. O Capítulo 5

apresenta os resultados obtidos através de simulações para cada estudo de caso. O Capítulo 6 traz as conclusões deste trabalho e indicações para futuros estudos.

2. SÉRIES TEMPORAIS

A definição de série temporal pode ser baseada em [4], onde é definida como um conjunto de variáveis estocásticas (probabilísticas) equiespaçadas e ordenadas no tempo. Outra definição pode ser dada a partir de [5], que define uma série temporal como observações de uma variável de interesse realizadas numa sequência cronológica, que quando analisadas, podem ajudar a prever comportamentos futuros a partir de características passadas da série, sendo que [6] citando [7] afirma que a característica mais importante de uma série temporal é que observações vizinhas são dependentes, onde o maior interesse é analisar esta dependência para se entender a dinâmica da série.

Segundo [8], podem-se observar outras características importantes e pertinentes das séries temporais, tais como a relevância da qualidade e confiabilidade dos dados utilizados, como sendo um pré-requisito para uma boa investigação.

A partir de [6], podem-se classificar os principais objetivos de uma análise de séries temporais como descrição, modelagem, previsão e controle. A descrição envolve a aplicação de cálculos estatísticos com o objetivo de extrair características da série, a modelagem consiste na identificação do processo gerador da série e estimação dos parâmetros do modelo que representa o processo, a previsão consiste na estimação de valores futuros da série temporal, e o controle seria o monitoramento dos valores da série, com o objetivo de detectar alterações na característica da mesma.

A seguir serão apresentadas as séries temporais que serão utilizadas para validar os métodos de identificação e otimização.

2.1. SÉRIE TEMPORAL DE BOX & JENKINS

Um problema de identificação de sistemas conhecido é a série de leituras de uma fornalha a gás, publicada em 1976 pelos autores Box e Jenkins [9]. A série

consiste em 296 pares de amostras de entrada e saída de uma fornalha, com tempo de amostragem de nove segundos. A entrada $x(k)$ é o fluxo de gás metano (pés³/min), e a saída $y(k)$ é a concentração de gás carbônico (CO₂), sendo k o índice da amostra. A série foi retirada do banco de dados da Universidade de York, disponível em [10].

A Figura 1 a seguir mostra os dados de entrada da série, já a Figura 2 apresenta os dados de saída da série.

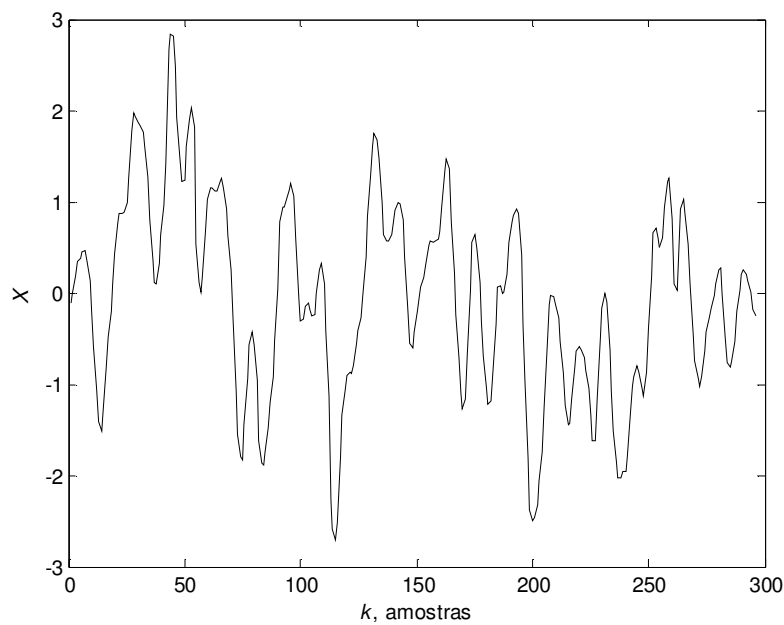


Figura 1: Sinal de entrada (pés³/min de metano) da fornalha.

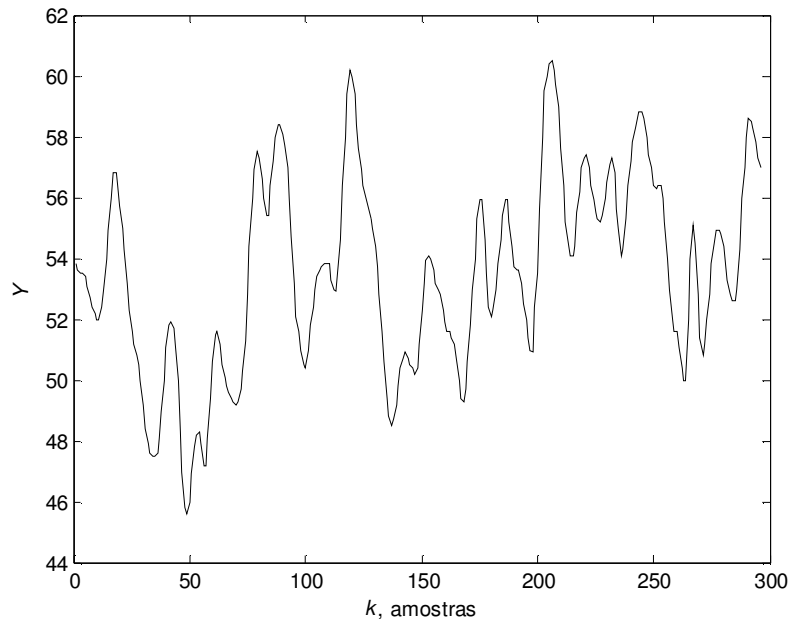


Figura 2: Sinal de saída (% CO₂) da fornalha.

As entradas selecionadas para o algoritmo de identificação foram escolhidas de acordo com [11], e são $y(k-1)$, $y(k-2)$ e $y(k-3)$, $x(k-1)$, $x(k-2)$ e $x(k-3)$, a previsão será $\hat{y}(k)$. Todas as amostras da série foram normalizadas no intervalo $[0,1]$, sendo que as primeiras 180 amostras foram selecionadas para o treinamento do algoritmo, e as 116 amostras restantes para validação do algoritmo.

2.2. SÉRIE TEMPORAL DE MACKEY-GLASS

A série temporal Mackey-Glass é gerada a partir de uma equação diferencial com comportamento caótico e é dada pela equação (2.1),

$$\frac{dx(t)}{dt} = \frac{0,2x(t-\tau)}{1+x^{10}(t-\tau)} - 0,1x(t), \quad (2.1)$$

onde τ é o atraso [12][13]. Os atributos escolhidos para o algoritmo de identificação são $x(k)$, $x(k-6)$, $x(k-12)$ e $x(k-18)$ e a saída estimada é $\hat{x}(k+6)$, conforme [12][13] sugerem. Todos os dados foram normalizados na faixa $[0,1]$.

Para este caso, foi utilizado o método de Euler para a resolução da equação diferencial e geração das 2500 amostras utilizadas. As 500 amostras iniciais serão descartadas a fim de evitar o início transitório.

Um sistema caótico pode ser definido como um sistema sensível a pequenas variações nas condições iniciais e/ou parâmetros [14][15]. Sendo assim, se faz necessário o uso de uma ferramenta que permita quantificar o caos do dado sistema. Alligood *et al.* [14] e Kinsner [15] demonstram a aplicação do cálculo do expoente de Lyapunov, que indica se um sistema é caótico, quasi-periódico ou periódico.

O conceito do expoente de Lyapunov é quantificar a separação exponencial entre duas trajetórias vizinhas e, a partir disto, pode-se obter a taxa média de divergência entre os pontos vizinhos, no caso de um sistema discreto, por passo.

O expoente de Lyapunov pode ser calculado discretamente a partir da equação (2.2).

$$\lambda = \frac{1}{\Delta n} \left(\ln \frac{d_n}{d_0} \right), \quad (2.2)$$

onde, Δn é o número de passos, d_0 é a distância inicial entre os pontos e d_n é a distância entre os pontos após Δn passos.

A Figura 3 mostra como as trajetórias que estavam inicialmente próximas, com uma distância tendendo a zero, divergem e após Δn passos estão a uma distância d_n que tenderá ao infinito se o sistema for caótico.

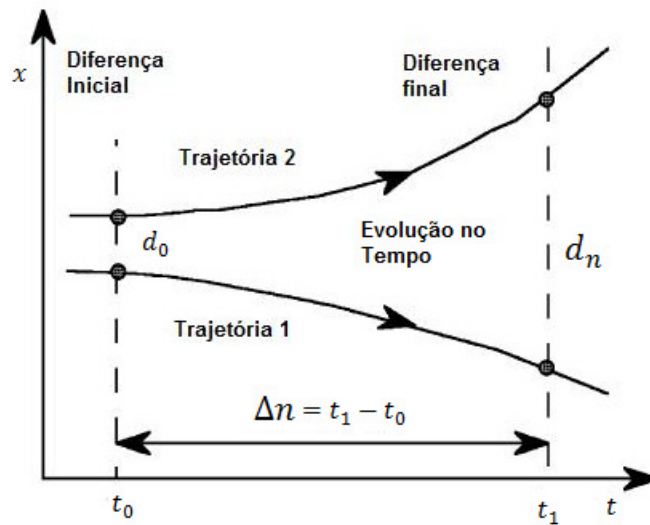


Figura 3: Trajetória de dois sistemas divergindo. Adaptada de [15].

A interpretação dos valores do expoente de Lyapunov é que sistemas caóticos irão gerar valores positivos do expoente, sistemas quasi-periódicos terão valores iguais a zero, e sistemas periódicos terão valores negativos do expoente de Lyapunov.

Com o expoente de Lyapunov, é possível determinar qual condição inicial e valor do parâmetro τ que tornam o sistema mais caótico. Para determinar estas condições, foram calculados os expoentes para as condições iniciais variando entre $-1 \leq x_0 \leq 1$ com passo de 0,01, e o parâmetro τ variando entre $17 \leq \tau \leq 40$, pois, segundo [13], a série torna-se caótica quando $\tau > 16,8$. A distância inicial d_0 escolhida é 0,001 e foram calculados 3000 passos da série.

Trabalhos anteriores [12][13][16] escolheram o parâmetro $\tau = 17$ para calcular os valores da série, mas como os cálculos do expoente de Lyapunov mostram, esta escolha não apresenta a situação mais caótica. A Tabela 1 mostra o expoente de Lyapunov λ e a condição inicial x_0 para $\tau = 17$ e para $\tau = 34$, que possui o maior expoente das faixas de valores de τ e de x_0 citadas anteriormente.

Tabela 1: Parâmetros da série Mackey-Glass e respectivos expoentes de Lyapunov.

τ	x_0	λ
17	0,24	$3,349 * 10^{-3}$
34	0,68	$3,533 * 10^{-3}$

A partir dos valores de λ , foram escolhidos os parâmetros $\tau = 34$ e $x_0 = 0,68$ para calcular a série temporal. A Figura 4 mostra a série com $\tau = 17$ e $x_0 = 0,24$, já a Figura 5 mostra a série com $\tau = 34$ e $x_0 = 0,68$.

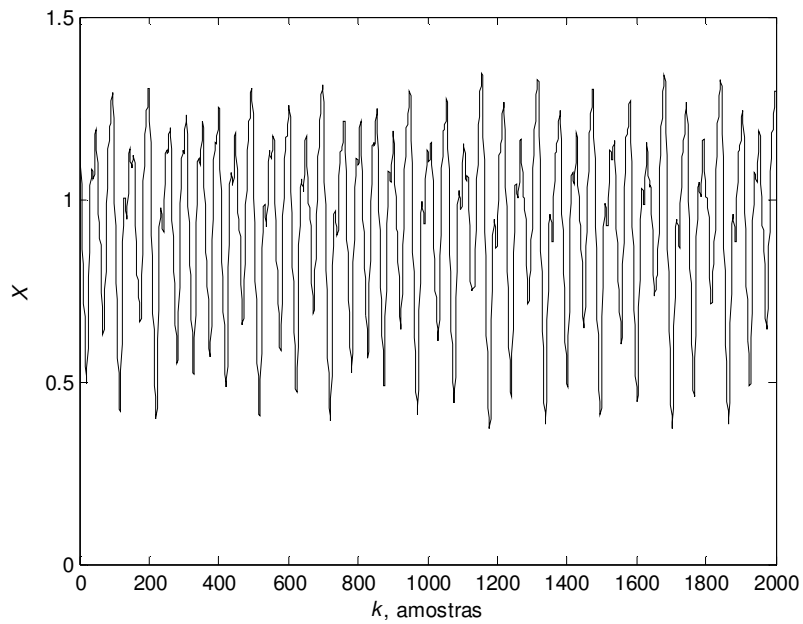


Figura 4: Série de Mackey-Glass com $\tau = 17$ e $x_0 = 0,24$.

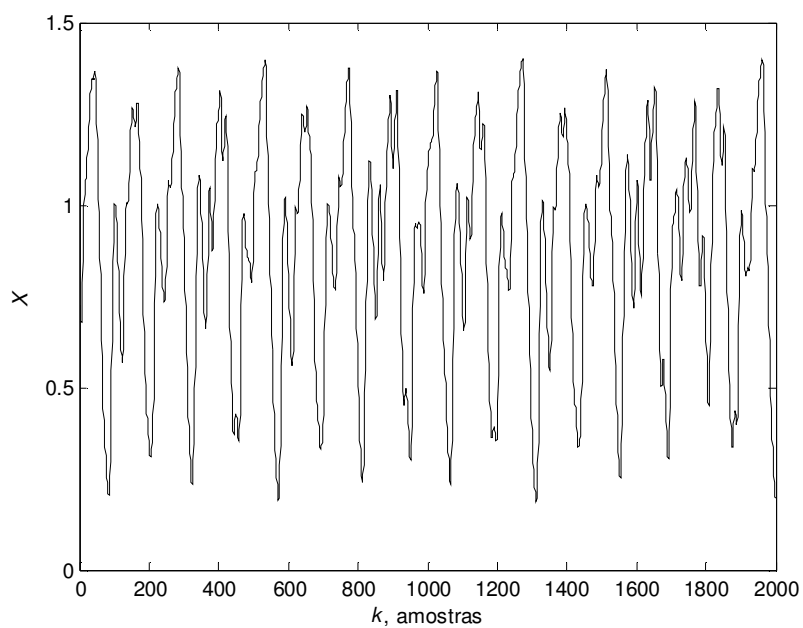


Figura 5: Série de Mackey-Glass com $\tau = 34$ e $x_0 = 0,68$.

2.3. SISTEMA DE LORENZ

O sistema de equações diferenciais de Lorenz foi desenvolvido com o objetivo de modelar o movimento de convecção do calor em um fluido como água ou ar [13].

Ele consistia inicialmente de sete equações diferenciais, mas após algumas simplificações que são descritas em [13], Lorenz obteve três equações diferenciais ordinárias,

$$\begin{cases} \frac{dx}{dt} = -\sigma x(t) + \sigma y(t), \\ \frac{dy}{dt} = rx(t) - y(t) - x(t)z(t), \\ \frac{dz}{dt} = x(t)y(t) - cz(t), \end{cases} \quad (2.3)$$

onde σ , r e c são parâmetros do sistema.

Lorenz notou que sempre que utilizava $\sigma = 10$ e $b = 8/3$, e o parâmetro r , também conhecido como número de Rayleigh (ou Reynolds), maior que aproximadamente $r \approx 24,74$, o sistema se comportava caoticamente, mostrando também uma sensibilidade às condições iniciais escolhidas [14]. Sendo assim, os valores dos parâmetros para as simulações foram escolhidos de acordo com [13] e [14], e são $\sigma = 10$, $b = \frac{8}{3}$ e $r = 28$. As condições iniciais escolhidas foram $x(0) = 12$, $y(0) = 2$ e $z(0) = 9$. O método de Euler foi utilizado para resolução das equações diferenciais.

Para estas condições, os valores dos expoentes de Lyapunov podem ser observados na Tabela 2.

Tabela 2: Expoentes de Lyapunov para cada dimensão da série de Lorenz.

λ_x	$3,082 * 10^{-3}$
λ_y	$3,074 * 10^{-3}$
λ_z	$3,062 * 10^{-3}$

A partir das condições iniciais e dos parâmetros, obtêm-se a série temporal com 3000 amostras mostrada na Figura 6, onde estão representadas as três equações nas três dimensões respectivas.

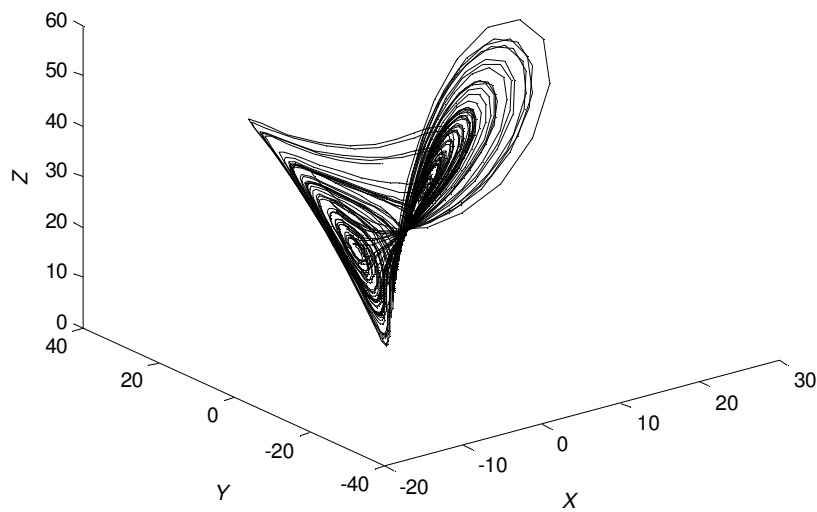


Figura 6: Gráfico tridimensional do sistema de equações diferenciais ordinárias de Lorenz.

Para o algoritmo, foram selecionados somente os dados da coordenada x , conforme o resultado dos expoentes de Lyapunov apresentados na Tabela 2, a coordenada x é mais caótica que as demais. A Figura 7 apresenta a série temporal formada pela coordena x , sendo que as primeiras 1500 amostras são utilizadas para treinamento e o restante utilizado para a validação do algoritmo.

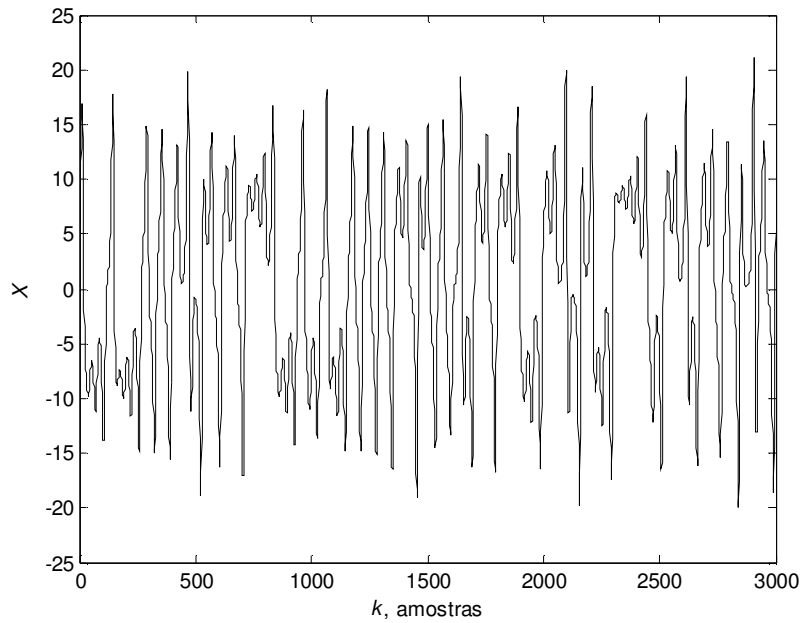


Figura 7: Série de Lorenz na coordenada x .

Os atributos escolhidos para a entrada do algoritmo de identificação foram baseados em [12] e são $x(k)$, $x(k - 8)$, $x(k - 16)$, $x(k - 24)$, $x(k - 32)$ e $x(k - 40)$. O algoritmo irá prever $\hat{x}(k + 1)$.

3. IDENTIFICAÇÃO DE SISTEMAS

A modelagem de sistemas faz parte da natureza humana, mesmo que este processo seja inconsciente. Diversas ações são guiadas por modelos mentais de atividades cotidianas, como por exemplo, no processo de aprender a dirigir um carro, que nada mais é que a construção de um modelo mental do carro [17].

A existência deste modelo mental pode ser facilmente notado quando se dirige um carro diferente, pois logo no início percebe-se que o controle da embreagem ficou ruim, freia-se bruscamente, etc. Após algum tempo o cérebro adaptou o modelo mental ao novo sistema (carro) e não mais cometerá os mesmo erros.

Esta necessidade de descrever o mundo ao nosso redor acompanha-nos desde os princípios da ciência, quando Isaac Newton formulou as leis de movimento e gravitação universal e Charles Coulomb, as leis da repulsão e atração das cargas elétricas [18].

A aplicação de equações diferenciais para reproduzir comportamentos dinâmicos foi uma das primeiras soluções encontradas para modelar matematicamente fenômenos naturais, como os descritos anteriormente [18].

Na indústria existe a necessidade de se modelar diversos processos, como os processos químicos, por exemplo, com o objetivo de criar um controle baseado em modelo.

A modelagem de sistemas a partir de equações diferenciais exige o desenvolvimento de um estudo do próprio processo físico envolvido, lhe dando uma característica mais conceitual, de modo que este tipo modelagem pode ser chamado de modelagem 'caixa-branca'. O uso de séries temporais extraídas de sistemas pode resultar num tipo de modelagem conhecido como 'caixa-preta' [19], devido a falta de conhecimento *a priori* do sistema envolvido em alguns casos.

Quando se opta por um modelo 'caixa-preta', é necessário estimar seus parâmetros através de métodos de aprendizagem, os quais podem ser classificados de acordo com o seu funcionamento:

- Aprendizagem em Batelada: quando todo o conjunto de dados deve estar disponível antes do início do processo de estimação, pois os

parâmetros serão calculados em um único conjunto de operações e permanecerão imutáveis [20];

- Aprendizagem Adaptativa: os dados são utilizados à medida que são disponibilizados, estimando-se constantemente os parâmetros. Pode ser chamado também de treinamento continuado [20].

Em situações de aprendizagem adaptativa onde os dados previstos anteriormente são realimentados no modelo, configura-se o caso de aprendizagem recursiva.

Com a popularização dos computadores nas últimas décadas, foram produzidas várias propostas de estruturas de equações matemáticas com o objetivo de identificar e modelar sistemas. Equações fundamentalmente baseadas em equações a diferenças, onde a dinâmica do sistema é reproduzida utilizando valores anteriores da saída. Existem diversas representações matemáticas, algumas destas serão apresentadas com maiores detalhes a seguir.

3.1. REPRESENTAÇÕES DE MODELOS

Existem diversas representações matemáticas utilizadas para modelar sistemas lineares e não-lineares, como por exemplo, representações utilizando funções de transferência contínua ou discreta, e espaço de estados. Dentro destas representações, podem-se citar algumas especialmente adequadas à identificação de sistemas em tempo discreto a partir de algoritmos para a estimação de parâmetros [17].

A equação (3.1) mostra o modelo geral dado por:

$$\begin{aligned} A(q)y(k) &= \frac{B(q)}{F(q)}x(k) + \frac{C(q)}{D(q)}v(k), \\ y(k) &= \frac{B(q)}{F(q)A(q)}x(k) + \frac{C(q)}{D(q)A(q)}v(k), \\ y(k) &= H(q)x(k) + G(q)v(k), \end{aligned} \tag{3.1}$$

onde q^{-1} é o operador atraso, $v(k)$ é o ruído branco, $y(k)$ é a saída prevista, $x(k)$ é o sinal de entrada, e $A(q)$, $B(q)$, $C(q)$, $D(q)$ e $F(q)$ são os polinômios definidos a seguir,

$$\begin{aligned}
 A(q) &= 1 - a_1 q^{-1} - \dots - a_{n_y} q^{-n_y} ; \\
 B(q) &= b_1 q^{-1} + \dots + b_{n_u} q^{-n_u} ; \\
 C(q) &= 1 + c_1 q^{-1} + \dots + c_{n_v} q^{-n_v} ; \\
 D(q) &= 1 + d_1 q^{-1} + \dots + d_{n_d} q^{-n_d} ; \\
 F(q) &= 1 + f_1 q^{-1} + \dots + f_{n_f} q^{-n_f} .
 \end{aligned} \tag{3.2}$$

As funções $H(q)$ e $G(q)$ são normalmente referidas como as funções de transferência do processo e do ruído, respectivamente.

A partir das definições anteriores se podem descrever algumas representações de modelos.

3.1.1. MODELO DE RESPOSTA AO IMPULSO

O modelo de resposta ao impulso é o somatório de convolução entre a resposta ao impulso de um sistema, $h(k)$, e um sinal de entrada $x(k)$, resulta na saída do sistema com ruído $e(k)$, conforme mostra a equação (3.3),

$$y(k) = \sum_{j=0}^{\infty} h(j)x(k-j) + e(k) , \tag{3.3}$$

sendo que a equação (3.3) é conhecida como resposta ao impulso infinita, ou IIR, do inglês, *Infinite Impulse Response*. O modelo de resposta ao impulso finita, ou FIR, do inglês, *Finite Impulse Response*, é obtido a partir da equação (3.1), quando $A(q) = C(q) = D(q) = F(q) = 1$ e $B(q) \neq 1$. Se $H(q)$ for assintoticamente estável, então

existe $M < \infty$, de forma que $h(k) = 0, \forall k > M$ [17]. Assim, “truncando” a equação (3.3) obtêm-se o modelo FIR,

$$y(k) = \sum_{j=0}^M h(j)x(k-j) + e(k) . \quad (3.4)$$

Se $B(q)$ é um polinômio arbitrário de ordem M , ou seja, $n_x = M$, a equação (3.4) pode ser reescrita da seguinte forma,

$$y(k) = B(q)x(k) + v(k) , \quad (3.5)$$

sendo que deve ser observado o seguinte aspecto, como $C(q)/D(q) = 1$, então o ruído $e(k)$ acrescentado na saída é branco, portanto, $e(k) = v(k)$.

3.1.2. MODELO ARX

O modelo auto-regressivo com entradas exógenas ARX, do inglês, *AutoRegressive with eXogenous inputs*, pode ser derivado do modelo geral (3.1) quando $C(q) = D(q) = F(q) = 1$. Sendo que $A(q)$ e $B(q)$ são polinômios arbitrários, resultando em

$$\begin{aligned} A(q)y(k) &= B(q)x(k) + v(k) , \\ y(k) &= \frac{B(q)}{A(q)}x(k) + \frac{1}{A(q)}v(k) . \end{aligned} \quad (3.6)$$

Ao contrário do modelo FIR, o ruído que é adicionado a saída, $e(k) = v(k)/A(q)$, não é branco [17].

Para as séries temporais, as quais não possuem entradas exógenas, ou seja, $x(k) = 0$, o modelo pode ser chamado Auto-Regressivo, ou AR.

3.1.3. MODELO NARX

O modelo não-linear auto-regressivo com entrada exógena, do inglês, *Nonlinear AutoRegressive with eXogenous input*, é um modelo discreto que explica o valor de $y(k)$ em função de valores anteriores dos sinais de entrada e saída, como mostra a equação (3.7),

$$y(k) = F[y(k-1), \dots, y(k-n_y), x(k-\tau_d), \dots, x(k-n_x)], \quad (3.7)$$

onde τ_d é o atraso inicial de x ; n_y e n_x são os maiores atrasos de y e x , respectivamente. De acordo com [17], é comum adicionar termos de ruído no modelo com objetivo de se evitar polarização de parâmetros, passando o modelo para a classe dos modelos NARMAX, explicado adiante.

3.1.4. MODELO ARMAX/NARMAX

A classe dos modelos auto-regressivos com média móvel e entradas exógenas, ARMAX, do inglês, *AutoRegressive Moving Average models with eXogenous inputs*, é uma forma comum de identificação de modelos lineares [21]. Sua estrutura relaciona uma sequência de entradas com uma sequência de saídas, e pode ser obtida através do modelo geral dado pela equação (3.1), onde $D(q) = F(q) = 1$ e $A(q)$, $B(q)$ e $C(q)$ são polinômios arbitrários. Assim têm-se,

$$\begin{aligned} A(q)y(k) &= B(q)x(k) + C(q)v(k) , \\ y(k) &= \frac{B(q)}{A(q)}x(k) + \frac{C(q)}{A(q)}v(k) , \\ y(k) &= H(q)x(k) + e(k) , \end{aligned} \quad (3.8)$$

sendo que (3.8) pode ser reescrita na forma de equações a diferenças linear com coeficientes constantes, conforme (3.9):

$$y(k) = \sum_{j=1}^{n_y} a_j y(k-j) + \sum_{j=0}^{n_x} b_j x(k-j) + \sum_{j=0}^{n_e} c_j e(k-j) , \quad (3.9)$$

onde n_y , n_x e n_e são os maiores atrasos de y , x e o maior atraso no modelo de ruído, respectivamente.

Quando o sistema a ser identificado se trata de uma série temporal, não possuindo entradas exógenas, ou seja, $x(k) = 0$, o modelo é chamado de ARMA.

O modelo NARMAX, do inglês, *Nonlinear AutoRegressive Moving Average models with exogenous inputs*, é utilizado para sistemas não-lineares e pode ser descrito da seguinte forma,

$$y(k) = F \left(y(k-1), y(k-2), \dots, y(k-n_y), x(k), x(k-1), \dots, x(k-n_x), e(k-1), e(k-2), \dots, e(k-n_e) \right) + e(k) , \quad (3.10)$$

onde $F(\cdot)$ é uma função não-linear de $(n_y + n_x + n_e + 1)$ variáveis.

A variação para séries temporais do modelo NARMAX, as quais não possuem entradas exógenas, conforme citado anteriormente, pode ser chamado de NARMA.

3.1.5. MODELO DE HAMMERSTEIN / WIENER

Uma forma simples de modelar não-linearidades é o modelo em blocos, que é feita com conexões em cascata e/ou paralelo de uma não-linearidade estática e uma linearidade dinâmica. O modelo mais conhecido desta família é o modelo de *Hammerstein*, que é composto por um modelo não-linear estático, seguido em cascata por um modelo linear dinâmico. Se a ordem for invertida, temos o modelo de *Wiener* [17][21].

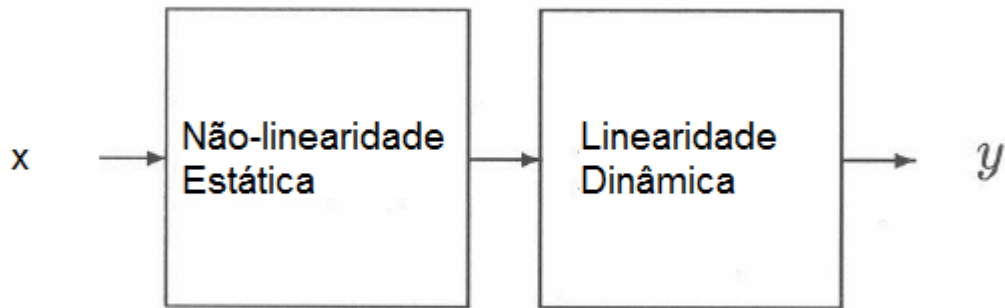


Figura 8: Modelo de *Hammerstein*.

A partir da Figura 8, o modelo de *Hammerstein* pode ser definido em tempo discreto como em (3.11):

$$y(k) = a_1y(k-1) + \dots + a_{n_y}y(k-n_y) + f[x(k-1) + \dots + x(k-n_x)], \quad (3.11)$$

onde apenas $y(k)$ e $x(k)$ são conhecidos em problemas de identificação, enquanto $f(\cdot)$ é uma função desconhecida [17].

Quando mais blocos são adicionados em cascata, têm-se o modelo sanduíche, o qual é composto por um modelo linear dinâmico, um modelo não-linear estático, e um modelo linear dinâmico, ou, *LNL* (Linear – Não-linear – Linear).

Modelos compostos por m blocos *Hammerstein* em paralelo são conhecidos como modelo de Uryson, ou, *PNL_m* (Paralelo – Não-linear – Linear) [21].

3.2. REDES NEURAIS ARTIFICIAIS

O cérebro humano é considerado um sistema altamente complexo, não-linear e de processamento paralelo [1]. O sistema nervoso humano contém mais de 100 bilhões de células chamadas de neurônios, que fazem o processamento de milhões de *bits* de informação enviados pelos diferentes órgãos sensoriais a cada minuto, para determinar respostas adequadas realizadas pelo corpo [22].

As conexões realizadas entre os neurônios são chamadas de sinapses, sendo que o número de sinapses realizadas por cada neurônio irá variar de acordo com a

função do neurônio, podendo chegar a até 200 mil sinapses com fibras enviando sinais para o neurônio [22].

A Figura 9 apresenta o formato básico de um neurônio, onde o sinal entra principalmente pelos dendritos, mas também pelo corpo celular [22], o sinal de saída viaja pelo axônio, que possui diversos 'galhos' separados, e deixa o neurônio em direção a outras partes do sistema nervoso ou partes do corpo.

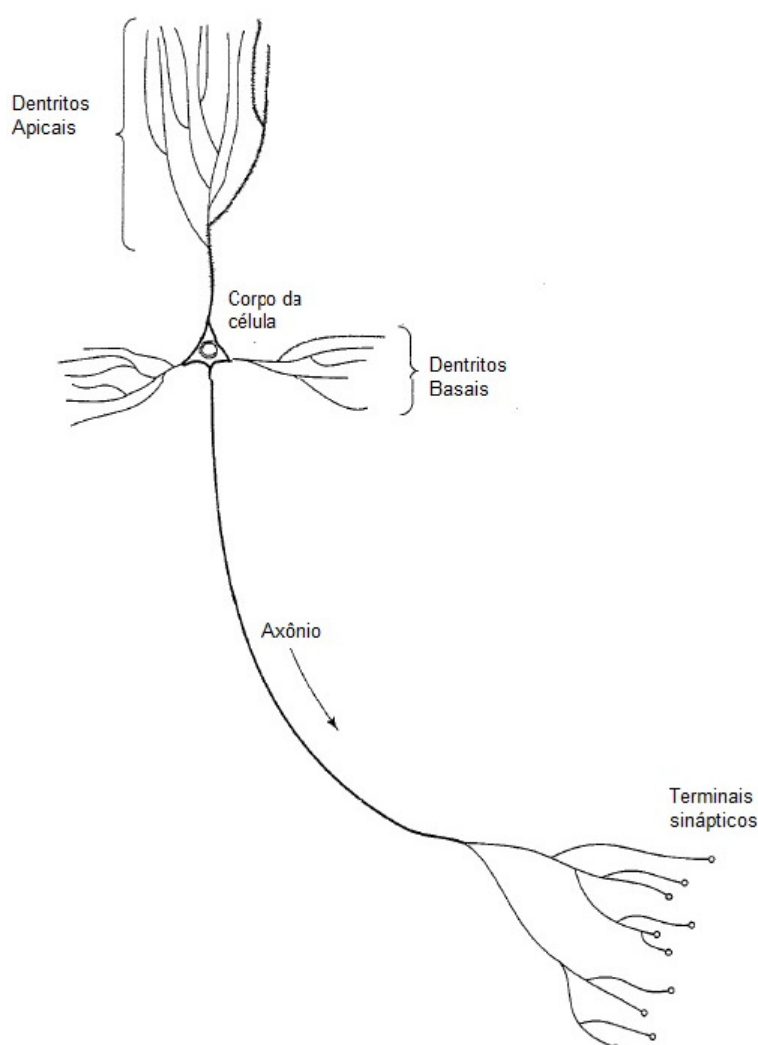


Figura 9: Descrição das partes de um neurônio de um animal vertebrado. Adaptada de [1].

Em 1943 McCulloch e Pitts publicaram um trabalho modelando matematicamente as operações biológicas de um neurônio, trabalho este que motivou Frank Rosenblatt a criar a primeira geração de redes neurais artificiais [23]. A Figura 10 mostra o modelo de neurônio de McCulloch e Pitts que é o mais amplamente utilizado [23].

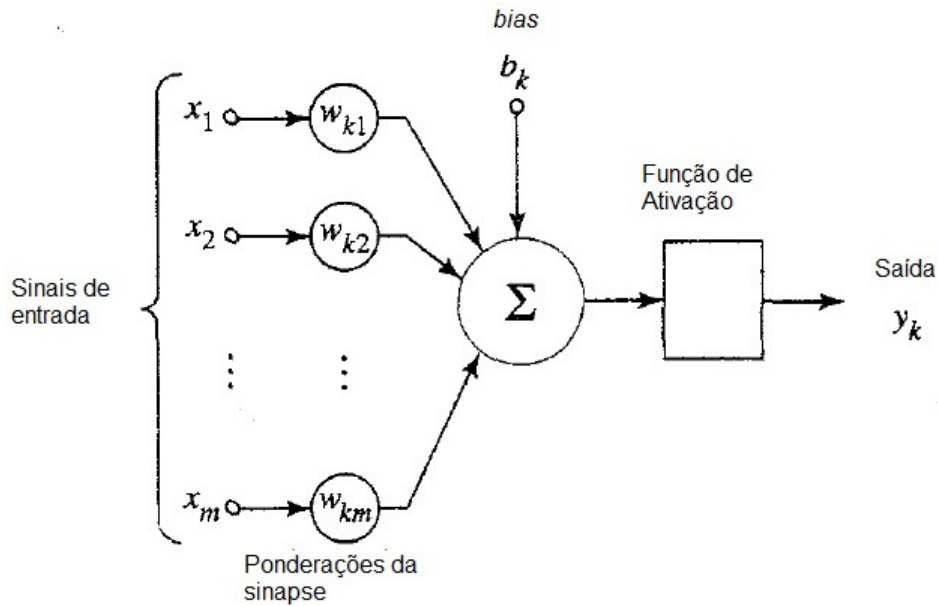


Figura 10: Modelo de neurônio de McCulloch e Pitts. Adaptado de [1].

O neurônio pode ser dividido em três partes: a função de rede, que determina como as entradas x da rede serão ponderadas e somadas. Essa função pode ser uma combinação linear ponderada, como é mostrado na Figura 10 e pode ser definido pela equação (3.12),

$$u = \sum_{i=1}^N w_i x_i + b_i, \quad (3.12)$$

onde N é o número de entradas, x_i é a i -ésima entrada, w_i é o peso de ponderação da i -ésima entrada, b_i é a polarização, em inglês, *bias*, e u é a saída da combinação linear. Neste trabalho a polarização b_i será chamada unicamente de *bias*.

A segunda parte do neurônio é a função de ativação,

$$y = g(u), \quad (3.13)$$

onde y é o sinal de saída do neurônio e $g(\cdot)$ é a função de ativação.

E a terceira e última parte do neurônio é uma ponderação de saída, ligando a camada de neurônios até a camada de saída, sendo o modelo final representado pela equação (3.14),

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_j \cdot x_j + b_i) = y_j, \quad (3.14)$$

$$j = 1, \dots, N,$$

onde β_i é a ponderação de saída do i -ésimo neurônio, \tilde{N} é o número de neurônios e $w_i \cdot x_j$ denota o produto interno entre w_i e x_j .

Diferentes funções podem ser utilizadas como uma função de ativação, lineares ou não-lineares. Algumas funções de ativação normalmente utilizadas podem ser observadas na Tabela 3.

Tabela 3: Tipos de função de ativação.

Tipo	Função de ativação	Fórmula
Linear	Linear	$g(u) = au + b$
Não-linear	Sigmóide	$g(u) = \frac{1}{1 + e^{-u}}$
Não-linear	Limiar	$g(u) = \begin{cases} 1, & u > 0 \\ -1, & u < 0 \end{cases}$

Numa rede neural artificial, múltiplos neurônios são interconectados em camadas para formar uma rede com a intenção de facilitar a computação distribuída [1][23]. Esta topologia pode ser composta de uma simples camada de neurônios, ou de múltiplas camadas de neurônios.

Na forma mais simples, a rede possui uma camada composta por nós de entrada, que propagam o sinal para uma camada composta por neurônios, ou, nós computacionais.

Como os dados seguem somente em um sentido, da entrada para a saída, esta é uma rede do tipo *feedforward*, se opondo à rede do tipo *feedback*, quando ocorre realimentação de dados da saída para a entrada.

A designação de 'camada simples' (*single layer*) se refere somente a camada de neurônios (nós computacionais), não incluindo os neurônios de entrada, pois

nenhum processamento é realizado neste [1]. Estas redes também são chamadas de *SLFN*, do inglês, *Single Layer Feedforward Neural Networks*.

Dependendo da função de ativação escolhida, redes do tipo *SLFN* podem ser interpretadas como modelos do tipo Uryson, descrito anteriormente. Cada neurônio da camada oculta pode ser ativado por uma função de ativação não-linear $g_i(\cdot)$, que pode ser considerado um modelo não-linear estático. A camada de saída é considerada um modelo linear dinâmico. Portanto, uma rede neural composta por \tilde{N} neurônios na camada oculta pode ser denominada como um modelo de Uryson $PNL_{\tilde{N}}$ [21].

A forma mais complexa é composta por múltiplas camadas de neurônios ocultos, onde o termo ‘oculto’ se refere ao fato de esta parte da rede neural não ser vista diretamente tanto pela entrada, quanto pela saída da rede. A adição de camadas ocultas possibilita mais graus de liberdade para a rede, empregando um número menor de neurônios [1].

Na identificação de sistemas, considera-se um mapeamento não-linear descrito pela relação,

$$y_R = f(x), \quad (3.15)$$

onde y_R é a saída e x é o vetor de entrada. A função $f(\cdot)$ é assumida ser desconhecida. Sendo assim, pode-se compor um conjunto η com os exemplos de dados conhecidos da função,

$$\eta = \{(x_i, y_{R_i})\}_{i=1}^N, \quad (3.16)$$

onde i é a amostra atual e N é o número total de amostras.

Conforme [1], o principal requerimento de uma rede neural para aproximar uma função desconhecida $f(\cdot)$ é que a função $F(\cdot)$ que descreverá as entradas-saídas mapeadas, função essa realizada pela rede neural, esteja perto o suficiente, num ponto de vista Euclidiano em todas as amostras, como pode ser descrito pela equação (3.17),

$$\|F(x) - f(x)\| < \varepsilon \quad \text{para todo } x \quad (3.17)$$

onde ε é o erro de aproximação e é um número positivo pequeno. Desde que o número de amostras N seja grande o suficiente e a rede esteja equipada com parâmetros livres suficientes, o erro de aproximação pode ser suficientemente pequeno para a tarefa [1].

A partir da diferença entre a saída real da função desconhecida e saída estimada pela rede, se obtém o vetor de sinal de erro e_i , sendo este vetor utilizado para ajustar os parâmetros livres, a fim de minimizar a diferença entre a saída real e a estimada.

Existem diversas formas de ajustar os parâmetros livres de uma rede neural, dentre as quais, têm-se:

- Aprendizagem *online* utilizando retropropagação;
- Máquina de aprendizagem extrema.

Neste trabalho apenas o método de aprendizagem extrema será abordado, em [1] e [23] encontram-se maiores detalhes sobre a aprendizagem utilizando retropropagação.

3.2.1. ALGORITMO DE APRENDIZAGEM EXTREMA

O algoritmo de aprendizagem extrema, em inglês, *Extreme Learning Machine* (ELM), foi proposto por Guang-Bin Huang, Qin-Yu Zhu e Chee-Kheong Siew em 2004 [24]. Os autores tinham como motivação criar um algoritmo de aprendizagem de redes neurais mais rápido que o método mais popular de aprendizagem, o método de retropropagação, pois o tempo dispensado na aprendizagem tem sido um dos gargalos de diversas aplicações nas últimas décadas.

O algoritmo ELM apresenta diversas vantagens em relação aos outros algoritmos, uma delas foi citada anteriormente, a velocidade de processamento pode ser milhares de vezes mais rápido que o algoritmo de retropropagação, além de obter um melhor desempenho na generalização. O algoritmo também tende a alcançar não somente o menor erro de treinamento, mas também a menor norma das ponderações [25].

Uma das características provadas pelos autores é que as ponderações de entrada e os *bias* da camada oculta da SLFN podem ser assinalados aleatoriamente se a função de ativação for infinitamente diferenciável. Ou seja, após escolher aleatoriamente as ponderações de entrada e os *bias*, a SLFN pode ser considerada um simples sistema linear e as ponderações de saída podem ser analiticamente calculadas através de uma operação de inversa generalizada.

A inversa generalizada de Moore-Penrose foi introduzida inicialmente em 1920 por Moore, sendo que algumas extensões foram propostas por outros autores, mas até 1955 nenhum estudo sistemático havia sido realizado, quando Penrose [26], sem conhecimento do trabalho anterior de Moore, redefiniu de uma forma diferente o método [27].

As definições de matriz inversa podem ser obtidas em [25][26][27][28]. Existem diferentes métodos para obter-se a inversa generalizada, sendo que eles resultam em matrizes inversas com as seguintes características:

- Inversa para uma solução de norma mínima;
- Inversa para uma solução de mínimos quadrados;
- Inversa para uma solução de mínimos quadrados com norma mínima.

Para maiores detalhes sobre o cálculo da matriz inversa generalizada, ver [24][25][26][27][28][29].

O algoritmo ELM faz uso da matriz inversa para uma solução de mínimos quadrados com norma mínima para o cálculo das ponderações de saída β .

Fazendo uso do modelo de neurônio descrito pela equação (3.14), pode-se escrever as N equações de forma compacta como,

$$H\beta = Y_R , \quad (3.18)$$

onde,

$$\begin{aligned}
H(w_1, \dots, w_N, b_1, \dots, b_{\tilde{N}}, x_1, \dots, x_N) \\
= \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}}, \quad (3.19)
\end{aligned}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad \text{e} \quad Y_R = \begin{bmatrix} y_{R1}^T \\ \vdots \\ y_{RN}^T \end{bmatrix}_{N \times m}, \quad (3.20)$$

onde $y_{Ri} = [y_{Ri1}, y_{Ri2}, \dots, y_{Rim}]^T \in R^m$, é a saída real, e H é chamada de matriz de saída da camada oculta de neurônios, onde i -ésima coluna de H é o i -ésimo nó de saída oculto com respeito as entradas x_1, x_2, \dots, x_N [24][25].

Se a função de ativação for infinitamente diferenciável, é possível provar que o número de neurônios necessários será $\tilde{N} \leq N$, para isso formula-se os seguintes teoremas a partir de [25] sobre a SLFN em questão,

Teorema (3.1): Dada uma SLFN normal com N neurônios ocultos e função de ativação $g: R \rightarrow R$, a qual é infinitamente diferenciável em qualquer intervalo, para as N amostras distintas (x_i, y_{Ri}) , onde $x_i \in R^n$ e $y_{Ri} \in R^m$, para qualquer w_i e b_i aleatoriamente escolhidos de um intervalo qualquer de R^n e R , respectivamente, de acordo com qualquer distribuição de probabilidade contínua, então, com probabilidade “1”, a matriz de saída da camada oculta H da SLFN é inversível e $\|H\beta - Y_R\| = 0$.

A prova do Teorema (3.1) pode ser encontrada em [24][25].

Teorema (3.2): Dado qualquer valor pequeno e positivo $\varepsilon > 0$ e função de ativação $g: R \rightarrow R$, a qual é infinitamente diferenciável em qualquer intervalo, então existe $\tilde{N} \leq N$ para N amostras arbitrariamente distintas (x_i, y_{Ri}) , onde $x_i \in R^n$ e $y_{Ri} \in R^m$, para qualquer w_i e b_i aleatoriamente escolhidos de um intervalo qualquer de R^n e R , respectivamente, de acordo com qualquer distribuição de probabilidade contínua, então, com probabilidade “1”, $\|H_{N \times \tilde{N}} \beta_{\tilde{N} \times m} - Y_{R_{N \times m}}\| < \varepsilon$.

A prova do Teorema (3.2) é clara, caso contrário simplesmente se faz $\tilde{N} = N$, o que faz $\|H_{N \times \tilde{N}} \beta_{\tilde{N} \times m} - Y_{R_{N \times m}}\| < \varepsilon$ de acordo com o Teorema (3.1). Se o número de neurônios ocultos for igual ao número de amostras distintas, a matriz H é quadrada e inversível quando as ponderações de entrada w_i e os *bias* ocultos b_i são aleatoriamente escolhidos, e a SLFN pode aproximar as amostras de treinamento com erro zero. A capacidade universal de aproximação de uma SLFN com pesos aleatórios pode ser vista com detalhes em [30].

Como na maioria dos casos, o número de neurônios ocultos é muito menor que o número de amostras distintas selecionadas para o treinamento, a matriz H normalmente não é quadrada e pode não existir uma solução para (3.18), então, pode se afirmar a seguinte definição baseado em [25][27][29]:

Definição (3.1): uma matriz G de ordem $n \times m$ é uma inversa generalizada de Moore-Penrose da matriz A de ordem $m \times n$, se

$$\begin{aligned}AGA &= A, \\GAG &= G, \\(AG)^T &= AG, \\(GA)^T &= GA.\end{aligned}\tag{3.21}$$

Para conveniência, a matriz inversa generalizada de Moore-Penrose da matriz A sera denotada por A^\dagger .

A partir da definição (3.1) e do método de obtenção da inversa por uma solução de mínimos quadrados com norma mínima, citado anteriormente, tem-se a seguinte equação,

$$\hat{\beta} = H^\dagger Y_R,\tag{3.22}$$

onde a solução especial $\hat{\beta}$ é uma das soluções de mínimos quadrados de um sistema linear geral dado pela equação (3.18), significando que o menor erro de treinamento pode ser alcançado por esta solução, como pode ser demonstrado a seguir, através de (3.23).

$$\|H\hat{\beta} - Y_R\| = \|HH^\dagger Y_R - Y_R\| = \min_{\beta} \|H\beta - Y_R\|. \quad (3.23)$$

Assim, pode-se resumir o algoritmo ELM no Pseudocódigo 1 que se encontra no Anexo A.

Como foi afirmado anteriormente, teoricamente, este algoritmo funciona para qualquer função de ativação infinitamente diferenciável, como por exemplo, a função sigmóide, seno, cosseno e exponencial.

Sendo assim, a função de ativação escolhida neste trabalho será a função sigmóide, a partir deste momento, denominada SIG, que pode ser definida como,

$$g(u) = \frac{1}{1 + e^{-uc}}, \quad (3.24)$$

onde c é uma constante positiva.

A Figura 11 mostra a função SIG para $c = 10$, com $-1 \leq x \leq 1$.

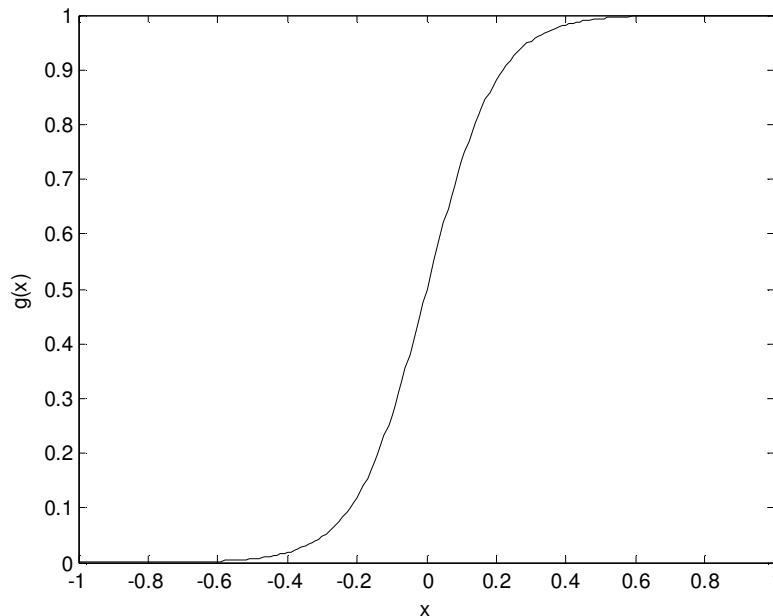


Figura 11: Resposta da função SIG.

O trabalho de Li, Li e Rong [31] sugere a utilização de mais de uma função de ativação, a fim de captar mais características do sistema analisado, onde o número

de neurônios com uma função, e outra, dependeria do problema. Com isso a função linear será combinada com a função sigmóide nas simulações, onde o número de neurônios destinados à função SIG $nSig$, deverá respeitar a restrição (3.25), $1 < nSig < \tilde{N}$. A função de ativação linear é dada por,

$$z_{lin}(u) = uk_{lin}, \quad (3.25)$$

onde k_{lin} é uma constante positiva.

A utilização da função de ativação sigmóide convencional com parâmetros fixos de ângulo de inclinação e posição provoca uma restrição no mapeamento não-linear da rede, além de diminuir a precisão do treinamento. Assim, [32] sugere a utilização de uma função de ativação variável como uma forma de corrigir estas falhas. A função de ativação variável, a partir deste momento denominada TAF, do inglês *Tunable Activation Function*, pode ser definida pela seguinte equação,

$$z_{TAF}(u) = \frac{\zeta}{1 + e^{-\frac{u-\delta}{\mu}}} + \alpha, \quad (3.26)$$

onde ζ é o fator de mapeamento, por ele alterar o comprimento e o alcance do mapeamento, α é o fator de deslocamento vertical, δ é o parâmetro de posicionamento horizontal, e μ é o fator do ângulo de inclinação, ele determina o formato de $z_{TAF}(u)$.

Uma situação especial de $z_{var}(u)$ é quando $\alpha = \delta = 0$ e $\zeta = \mu = 1$, resultando na função SIG dada pela equação (3.24).

3.3. VALIDAÇÃO DE MODELOS

A validação de modelos é um passo importante no processo de identificação de sistemas, sendo que ele normalmente é realizado nas etapas finais do procedimento para checar se o modelo identificado é satisfatório [33][34].

Diversos métodos foram desenvolvidos para a validação de modelos lineares e não-lineares, sendo que os testes utilizando as funções de correlação tornaram-se populares devido a vantagem em diagnosticar diretamente se um modelo identificado é adequado ou não. Métodos baseados em comparação de modelos tem a desvantagem da necessidade de testar todos os conjuntos de modelos possíveis antes de se obter um diagnóstico [34].

O método baseado em correlação se baseia no fato de que em um sistema identificado de forma ideal, o resíduo $\xi(k)$ deve ser reduzido a uma sequência de dados não-correlacionados definidos como $e(k)$, com média zero e variância finita. Portanto, testes de validação de modelos baseados em correlação normalmente são validados, se

$$\xi(k) \approx e(k), \quad (3.27)$$

onde o resíduo ξ é definido como em (3.28)

$$\xi = Y_R - \hat{Y}. \quad (3.28)$$

A Função de AutoCorrelação, ou FAC, de um sinal $x(k)$ pode ser definida por (3.29) [17],

$$r_{x'x'}(\tau, k) = E[(x(k) - \bar{x})(x^*(k + \tau) - \bar{x})], \quad (3.29)$$

sendo que, se $x(k)$ for real, $x^*(k + \tau) = x(k + \tau)$, onde τ é a unidade de atraso, e $E[\cdot]$ é a esperança matemática.

Normalmente os dados utilizados são coletados, ou seja, são sinais reais e com média nula [17]. A partir dessa informação, pode-se reescrever a equação (3.29) como (3.30).

$$r_{xx}(\tau, k) = E[x(k)x^*(k + \tau)], \quad (3.30)$$

onde, $r_{x'x'}(\tau, k) = r_{xx}(\tau, k)$ quando \bar{x} é igual a zero [33].

Sabendo que k é um valor inteiro e N tem um valor suficientemente elevado, a função de autocorrelação baseada em (3.29) pode ser estimada a partir de séries discretas no tempo,

$$r_{x'x'}(k) = \lim_{N \rightarrow \infty} \frac{1}{2N + 1} \sum_{i=-N}^N (x(i) - \bar{x})(x(i+k) - \bar{x}), \quad (3.31)$$

sendo que a equação (3.31) pode ser reescrita a seguir para o caso prático de séries temporais de duração finita, baseado na seguinte definição.

Definição (3.3): Se um processo for estacionário, então sua função de autocorrelação será simétrica, ou seja, $r_{x'x'}(\tau) = r_{x'x'}(-\tau)$ [17].

A partir da definição anterior, obtém-se a nova FAC, $\hat{r}_{x'x'}(k)$,

$$\hat{r}_{x'x'}(k) = \frac{1}{N} \sum_{i=1}^{N-k} (x(i) - \bar{x})(x(i+k) - \bar{x}), \quad k = 0, 1, \dots, N - 1. \quad (3.32)$$

Outra forma de calcular a autocorrelação de um sinal é a partir da função de covariância. Isto é possível devido à forma como os dados são coletados, fato citado anteriormente. Em casos como este, a função de autocorrelação coincide com função de autocovariância, como mostra as definições a seguir [17].

Definição (3.4): A matriz de covariância de dimensão $n \times n$ é definida em (3.33).

$$cov[x] = E[(x - E[x])(x - E[x])^T] = E[xx^T] - E[x]E[x^T], \quad (3.33)$$

onde $E[x]$ pode ser substituído por \bar{x} , devido a propriedade que determina a média assintótica como sendo

$$\overline{x_a(N)} = \lim_{N \rightarrow \infty} E[x(N)]. \quad (3.34)$$

Definição (3.5): A função de autocovariância de $x(k)$ é definida em (3.35).

$$c_{xx}(\tau, k) = Cov[x(k), x(k + \tau)] = E[x(k)]E[x^*(k + \tau)]. \quad (3.35)$$

O estudo da correlação entre dois sinais diferentes com média nula, $x(k)$ e $y(k)$, resulta Função de Correlação Cruzada, ou FCC, e pode ser definida como,

$$r_{xy}(\tau, k) = E[x(k)y^*(k + \tau)], \quad (3.36)$$

sendo que se o processo for considerado real, $y^*(k) = y(k)$. A equação (3.36) pode ser estimada a partir de séries discretas no tempo como (3.37),

$$r_{xy}(k) = \lim_{N \rightarrow \infty} \frac{1}{2N + 1} \sum_{i=-N}^N x(i)y(i + k). \quad (3.37)$$

No caso de séries temporais de duração finita, a equação (3.37) pode ser definida da seguinte forma,

$$\hat{r}_{xy}(k) = \begin{cases} \frac{1}{N} \sum_{i=1}^{N-k} x(i)y(i + k), & k = 0, 1, \dots, N - 1 \\ \frac{1}{N} \sum_{i=1}^{N+k} y(i)x(i - k), & k = 0, -1, \dots, -N + 1 \end{cases}, \quad (3.38)$$

sendo que é importante ressaltar que se algum sinal não tiver média nula, a respectiva média deve ser subtraída antes de utilizar o estimador (3.38) [17].

Assim como a FAC, a FCC pode ser obtida através do uso da função covariância cruzada, definida a seguir.

Definição (3.6): A função de covariância cruzada de $x(k)$ e $y(k)$ é definida em (3.39).

$$c_{xy}(\tau, k) = Cov[x(k), y(k + \tau)] = E[x(k)]E[y^*(k + \tau)]. \quad (3.39)$$

A partir da definição (3.6) pode-se determinar o coeficiente de correlação entre as variáveis x e y como (3.40),

$$\rho(x, y) = \frac{Cov[x, y]}{(\sigma_x \sigma_y)}, \quad (3.40)$$

onde σ_x e σ_y o desvio padrão dos sinais x e y , respectivamente [17][35].

Da mesma forma obtêm-se o coeficiente de correlação em função do deslocamento τ , $\rho_{x,x}(\tau)$, para as séries $x(k)$ e $x(k + \tau)$.

Aplicar a FAC e FCC ao conceito de séries temporais têm o objetivo de identificar termos atrasados lineares e não-lineares não modelados. Os trabalhos de [33] e [34] apresentam a utilização de testes para identificação de termos não-lineares, como por exemplo, um ruído não-linear atrasado $e(k - 1)e^3(k - 4)$.

Em [34] são propostos dois novos testes baseados na relação $y\xi(k) = y(k)\xi(k)$. Estes testes permitem análise da correlação cruzada entre a saída e o resíduo e também entre a saída, resíduos e entrada, aumento o poder dos testes de correlação cruzada. Também será empregada a relação $x\xi(k) = x(k)\xi(k)$ com o mesmo objetivo.

Para os testes não-lineares de alta ordem serão utilizados sinais de entrada e resíduo elevados ao quadrado nos testes de autocorrelação e correlação cruzada, assim têm-se os seguintes testes lineares e não-lineares a serem realizados mostrados na Tabela 4 a seguir.

Tabela 4: Testes de correlação a serem aplicados.

$\hat{r}_{x,\xi}(\tau)$	Correlação cruzada entre entrada e resíduo	Linear
$\hat{r}_{x^2,\xi^2}(\tau)$	Correlação cruzada entre entrada e resíduo	Não-linear
$\hat{r}_{x^2,\xi}(\tau)$	Correlação cruzada entre entrada e resíduo	Não-linear
$\hat{r}_{\xi,\xi}(\tau)$	Autocorrelação do resíduo	Linear
$\hat{r}_{\xi^2,\xi^2}(\tau)$	Autocorrelação do resíduo	Não-linear
$\hat{r}_{Y\xi,\xi^2}(\tau)$	Correlação cruzada entre saída, resíduo e resíduo	Não-linear
$\hat{r}_{Y\xi,x^2}(\tau)$	Correlação cruzada entre saída, resíduo e entrada	Não-linear
$\hat{r}_{x\xi,\xi}(\tau)$	Correlação cruzada entre entrada, resíduo e resíduo	Não-linear

O objetivo dos testes de correlação é que os resíduos sejam reduzidos a uma sequência não-correlacionada com todas as combinações lineares e não-lineares de entradas e saídas passadas [34]. Cada teste detecta termos diferentes, $\hat{r}_{x\xi}(\tau)$ detecta termos atrasados de $x(k)$, $\hat{r}_{\xi\xi}(\tau)$ detecta termos atrasados de $e(k)$, $\hat{r}_{x^2\xi^2}(\tau)$ detecta termos atrasados do tipo $x(k-1)x(k-2)$, $\hat{r}_{\xi^2\xi^2}(\tau)$ detecta termos do tipo $e(k-1)e^3(k-4)$, por exemplo. A detecção de termos atrasados da saída $y(k)$ deve ser feita a partir da observação dos testes $\hat{r}_{x\xi}(\tau)$ e $\hat{r}_{\xi\xi}(\tau)$, da mesma forma para termos do tipo $x(k-1)y(k-2)$, por exemplo, onde os testes $\hat{r}_{x^2\xi^2}(\tau)$ e $\hat{r}_{\xi^2\xi^2}(\tau)$ devem ser observados.

Idealmente, quando os resíduos possuem média zero e não são correlacionados, os testes de correlação devem ter o seguinte comportamento,

$$\begin{aligned}
 \hat{r}_{x,\xi}(\tau) &= 0, \forall \tau, \\
 \hat{r}_{x^2,\xi^2}(\tau) &= 0, \forall \tau, \\
 \hat{r}_{x^2,\xi}(\tau) &= 0, \forall \tau, \\
 \hat{r}_{\xi,\xi}(\tau) &= \begin{cases} 1, & \tau = 0 \\ 0, & \forall \tau \neq 0 \end{cases}, \\
 \hat{r}_{\xi^2,\xi^2}(\tau) &= \begin{cases} 1, & \tau = 0 \\ 0, & \forall \tau \neq 0 \end{cases}, \\
 \hat{r}_{Y\xi,\xi^2}(\tau) &= \begin{cases} k\hat{r}_{Y\xi,\xi^2}, & \tau = 0 \\ 0, & \forall \tau \neq 0 \end{cases},
 \end{aligned} \tag{3.41}$$

$$\hat{r}_{Y\xi, x^2}(\tau) = 0, \forall \tau ,$$

$$\hat{r}_{x\xi, \xi}(\tau) = 0, \forall \tau .$$

A constante $k_{\hat{r}_{Y\xi, \xi^2}}$ pode ser calculada pela equação (3.42).

$$k_{\hat{r}_{Y\xi, \xi^2}} = \frac{(\sum_{k=1}^N (\xi^2(k) - \overline{\xi^2})^2)^{1/2}}{(\sum_{k=1}^N (y(k)\xi(k) - \overline{Y\xi})^2)^{1/2}} . \quad (3.42)$$

Os resíduos analisados podem estar correlacionados com os dados de entrada, sendo assim, o objetivo dos testes é procurar picos positivos de correlação com atrasos negativos, no caso das correlações cruzadas. Segundo [17], valores positivos de correlação cruzada em um atraso positivo podem indicar um tempo morto do sistema.

4. OTIMIZAÇÃO

Algoritmos de otimização são métodos de procura ou busca, que têm por objetivo obter uma solução para um problema de otimização, onde, segundo [36], estes problemas podem ou não serem sujeitos a uma ou mais restrições.

Um problema de otimização possui alguns componentes básicos:

- função objetivo: representa a quantidade a ser otimizada, podendo ser maximizada ou minimizada. Alguns problemas podem não possuir uma função objetivo, assim, o objetivo passa a ser determinar uma solução que satisfaça as restrições do problema, no caso de problemas com presença de restrição. A função objetivo pode ser também chamada de, em inglês, *fitness*, ou aptidão. Este termo foi utilizado originalmente nos algoritmos de otimização baseados na Teoria da Evolução de Darwin, onde somente os mais aptos sobrevivem;
- conjunto de variáveis: elas afetam o valor da função objetivo;
- conjunto de restrições: elas restringem os valores das variáveis, podendo até excluir possíveis soluções candidatas.

As soluções para os problemas nem sempre são simples, sendo que elas podem consistir de:

- diferentes tipos de dados, tais como variáveis inteiras ou de ponto flutuante;
- diferentes áreas de procura, podendo estas serem restritas por restrições lineares ou não-lineares.

Os problemas também podem apresentar algumas dificuldades a mais para o algoritmo, como por exemplo:

- as características podem variar com o tempo, desqualificando soluções encontradas anteriormente e consideradas boas;
- os objetivos podem ser conflitantes.

A solução é designada a partir de valores de um domínio permitido para as variáveis, sendo que este domínio, ou pode ser restrito somente nos limites dos valores, onde o problema é dito 'problema irrestrito', em inglês, *unconstrained problem*, ou pode ser restrito por equações e/ou inequações lineares ou não-lineares, onde o problema é dito 'problema restrito', em inglês, *constrained problem*.

Um problema de otimização pode ser classificado a partir de algumas características:

- número de variáveis que influenciam a função objetivo;
- tipo das variáveis, contínuas ou inteiras;
- grau de não-linearidade da função objetivo;
- restrições utilizadas;
- número de critérios de otimização, se a quantidade a ser otimizada é expressa em somente uma função objetivo, o problema é denominado de objetivo simples, problemas multi-objetivo especificam mais de um sub-objetivo a serem otimizados simultaneamente;
- número de soluções, para um problema onde existe uma única solução, é chamado unimodal, um problema com mais de uma solução é chamado multimodal.

O espaço de procura para as soluções, denominado espaço de procura S , representa todo o universo de soluções possíveis ou impossíveis. O espaço possível de procura, denominado espaço possível \mathcal{F} , é delimitado pelas restrições e é onde se encontram as soluções possíveis para o problema. A Figura 12 ilustra estes conceitos.

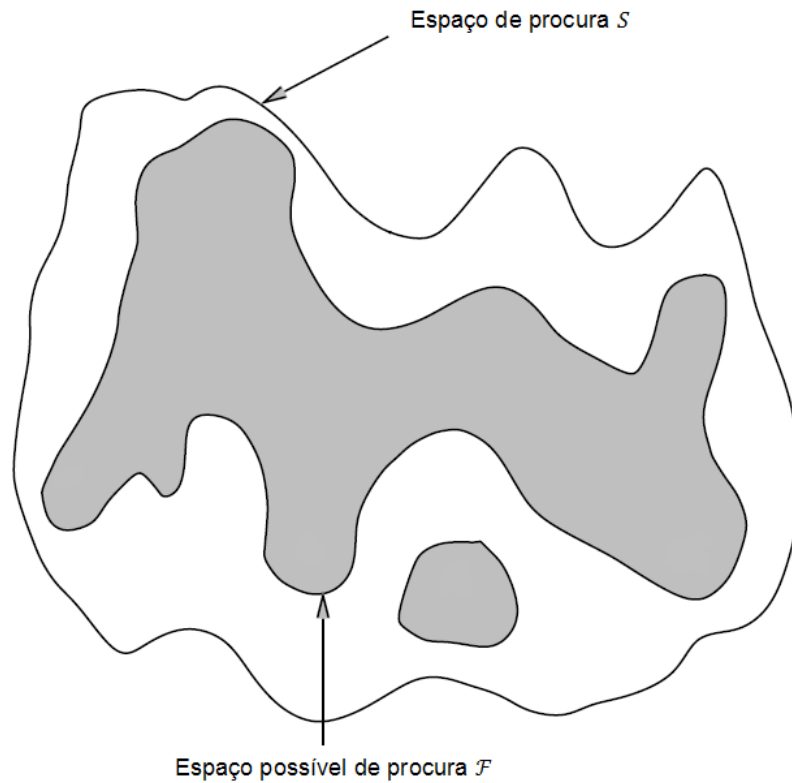


Figura 12: Ilustração mostrando o espaço de procura S e o espaço possível de procura \mathcal{F} . Adaptado de [37].

As soluções encontradas pelos algoritmos de otimização podem ser classificadas de acordo com a sua qualidade. Para apresentar as definições de diferentes tipos de soluções, será assumido um problema de minimização da função objetivo, cuja definição se encontra a seguir.

Definição (4.1): Um problema de minimização é composto por um conjunto de soluções válidas X do espaço possível de procura \mathcal{F} , e $f : X \rightarrow \mathcal{R}$ é a função objetivo. O objetivo é achar uma solução global ótima, ou seja, uma solução $x^* \in X$ tal como $f(x^*) \leq f(x), \forall x \in X$. Com isso, $f^* = f(x^*)$ é o custo ótimo e $X^* = \{x \in X \mid f(x) = f^*\}$ é o conjunto de soluções ótimas globais [38].

Definição (4.2): Uma solução mínima global de uma função objetivo f é uma solução $x^* \in \mathcal{F}$, onde x é uma possível solução, tal que,

$$f(x^*) < f(x), \forall x \in \mathcal{F}, \tag{4.1}$$

onde $\mathcal{F} \subseteq S$ e S é o espaço de procura.

Definição (4.3): Uma solução mínima local de uma função objetivo f é uma solução $x_{\mathcal{N}}^* \in \mathcal{N} \subseteq \mathcal{F}$, se,

$$f(x_{\mathcal{N}}^*) < f(x), \forall x \in \mathcal{N}, \quad (4.2)$$

onde $\mathcal{N} \subseteq \mathcal{F}$ e \mathcal{N} é um conjunto de pontos possíveis na vizinhança de $x_{\mathcal{N}}^*$ [36].

A Figura 13 exemplifica o que seriam os pontos mínimos de uma função.

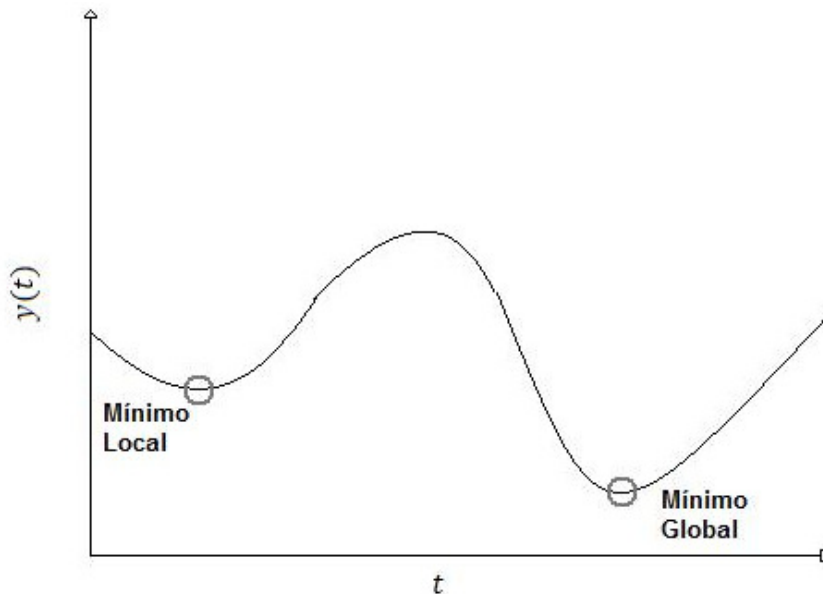


Figura 13: Mínimos de uma função.

Para procurar a melhor solução pode-se optar entre duas formas de procura, algoritmos de procura local e algoritmos de procura global. Os algoritmos de procura local somente procuram soluções ótimas locais, as quais podem ser ótimos globais. Os algoritmos de procura global fazem uso de mais informações para procurar por todo o espaço de busca e então obter uma solução ótima global.

Os métodos de busca podem ser divididos em três grupos, os métodos enumerativos, os métodos determinísticos e os métodos estocásticos.

Os métodos enumerativos, como por exemplo, a Programação Dinâmica, possuem um mecanismo de procura simples, estipula-se um universo finito de

busca, discretiza-se este para representar todas as possíveis soluções, e verificam-se todos os pontos. Como afirma [39], apesar de sua simples implementação, fica evidente que o custo computacional é grande para problemas onde o espaço de busca é grande, além da realidade que uma discretização, por mais fina que seja, dificilmente cobrirá todos os pontos possíveis, assim, corre-se o risco de a solução global não ter sido discretizada.

Os métodos determinísticos, como por exemplo, o método de Newton, utilizam cálculos de derivadas e necessitam de outras informações, como o gradiente da função objetivo, para determinar a sua direção. Devido ao uso de informações de pontos anteriores, sua busca pela melhor solução é local e em certos problemas estes tendem a ficar presos em mínimos locais.

Os métodos estocásticos usam regras de probabilidade para buscar a solução. A vantagem disso é aumentar a área de procura de soluções do algoritmo, também aumentando a possibilidade do algoritmo encontrar uma solução ótima global. Alguns exemplos de métodos estocásticos seriam o Algoritmo Genético e o Algoritmo de Enxames de Partículas, sendo que o último será apresentado em detalhes futuramente.

4.1. ALGORITMO DE ENXAME DE PARTÍCULAS

O algoritmo de otimização por enxame de partículas, em inglês, *Particle Swarm Optimization (PSO)*, foi proposto inicialmente por Kennedy e Eberhart em 1995 [40]. O algoritmo se baseia no comportamento social de um bando de pássaros procurando por comida, onde cada pássaro seria uma partícula, e no algoritmo, esta partícula seria uma possível solução para o problema.

O PSO já se mostrou robusto para diversas aplicações, como por exemplo, na otimização de formas elípticas para a concepção de modelos de próteses cranianas para reparação [41], e na otimização dos parâmetros de funcionamento de controladores do tipo Proporcional-Integral-Derivativo (PID) para a sincronização de dois sistemas caóticos [42].

A partir de [36], a explicação do funcionamento do algoritmo pode ser simplificada ao imaginar as partículas “voando” pelo espaço de busca multidimensional. Durante este “voo”, as partículas trocam experiências com seus vizinhos, a fim de que a nova posição da partícula seja ajustada de acordo com suas próprias experiências e a experiência de seus vizinhos. A Figura 14 exemplifica um espaço de busca onde as partículas estão se direcionando para a partícula (destacada) que possui a melhor solução da função objetivo até o presente momento.

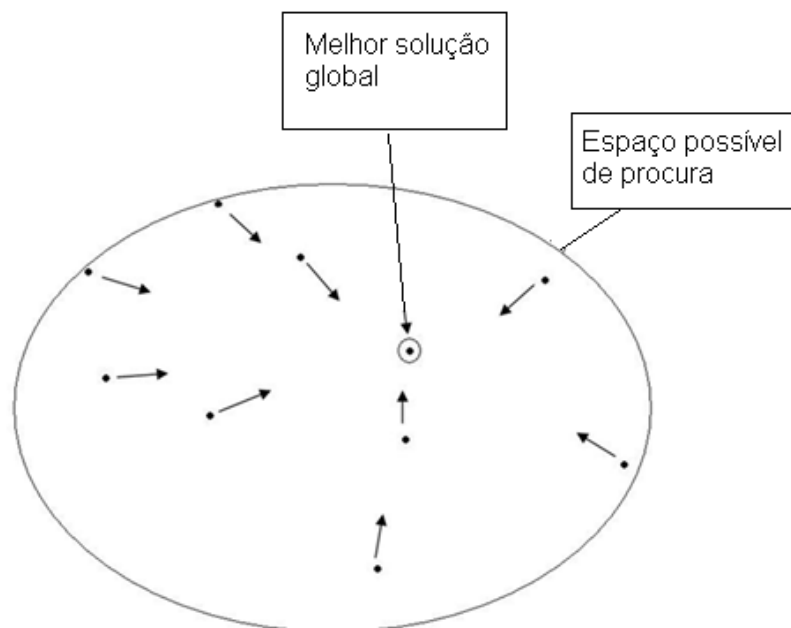


Figura 14: Movimento das partículas em direção à melhor solução.

Sendo assim, pode-se definir a mudança da posição da partícula como a adição de um termo à posição anterior, denotado da seguinte forma,

$$\varphi_i(it + 1) = \varphi_i(it) + \vartheta_i(it + 1), \quad (4.3)$$

onde $\varphi_i(it)$ é a posição atual da partícula i , ϑ_i é a velocidade da partícula i e it é a iteração atual do algoritmo.

A equação (4.3) indica que o vetor velocidade direciona o processo de otimização, sendo somado à posição anterior da partícula. Neste vetor estão contidos os conhecimentos trocados com as outras partículas que fazem parte do

enxame, e também as informações adquiridas anteriormente com a própria experiência da partícula.

Este conhecimento experimental também é chamado de componente cognitiva, já o conhecimento trocado entre a partícula e suas vizinhas é chamado de componente social.

A equação (4.4) apresenta como é calculado o vetor velocidade,

$$v_i(it + 1) = \omega \cdot v_i(it) + r_1 \cdot c_1 \cdot (lp_i(it) - \varphi_i(it)) + r_2 \cdot c_2 \cdot (gP(it) - \varphi_i(it)), \quad (4.4)$$

onde ω é o peso de inércia da partícula, lp_i é a melhor posição que a partícula alcançou até a iteração it , gP é a melhor posição alcançada entre todas as partículas do enxame até a iteração it , c_1 e c_2 são os coeficientes de aceleração, e r_1 e r_2 são valores aleatórios com distribuição uniforme no intervalo $[0,1]$.

A equação (4.4) contém o peso de inércia da partícula que não estava presente no algoritmo canônico criado em 1995 [40]. Este fator foi incorporado à equação da velocidade em 1998 por Shi e Eberhart [43] como uma tentativa de eliminar o grampeamento da velocidade, artifício utilizado como uma forma de controlar a exploração global e local do algoritmo [44] e para evitar o fenômeno conhecido como “explosão do enxame”. Em 1999 Maurice Clerc [45] propôs um termo de constrição χ que seria calculado de acordo com os valores dos coeficientes de aceleração c_1 e c_2 ,

$$\sigma = c_1 + c_2 > 4,0, \quad (4.5)$$

$$\chi = \frac{2}{2 - \sigma - \sqrt{\sigma^2 - 4 \cdot \sigma}}, \quad (4.6)$$

mas o estudo posterior de Eberhart e Shi [46] mostrou que o termo de constrição e o peso de inércia são equivalentes em condições especiais, sendo que o algoritmo com termo de constrição pode ser considerado um caso especial do algoritmo com peso de inércia, já que seus parâmetros estão conectados pela equação (4.6).

O peso de inércia é um parâmetro importante no algoritmo, influenciando diretamente na convergência do mesmo. Podendo ser descrito como um mecanismo

de controle do momento da partícula, ponderando a contribuição da velocidade anterior, como uma forma de memória da direção anterior da partícula, com isso, pode alterar a habilidade do algoritmo em explorar o espaço de busca. Para valores maiores que “1”, o algoritmo terá a tendência de divergir. Valores altos e menores que “1” priorizam uma busca global, aumentando a diversidade do enxame, enquanto valores mais próximos de “0” priorizam uma busca local.

Os coeficientes de aceleração c_1 e c_2 atuam em conjunto com os valores aleatórios r_1 e r_2 na influência estocástica dos componentes social e cognitivo. Estes coeficientes também são chamados de coeficientes de confiança, pois expressam quanto a partícula confia em si própria (c_1) e quanto confia nos seus vizinhos (c_2). Valores pequenos para estes parâmetros garantem uma trajetória suave para a partícula, já valores altos causam maior aceleração e movimentos bruscos. Eberhart e Shi afirmam que experiências anteriores mostraram empiricamente que os melhores valores para os coeficientes de aceleração são ambos iguais a 2 para a maioria das aplicações [44][47], inclusive quando o peso de inércia é substituído pelo fator de restrição, sendo que neste caso o valor sugerido é $c_1 = c_2 = 2,05$ por questões de convergência [48].

As interações da partícula com sua vizinhança são definidas como topologia do enxame. Analisando a equação (4.4), mais precisamente sua componente social, nota-se uma interação onde a partícula troca experiências com todas as outras do enxame, essa topologia é chamada de topologia de estrela, ou topologia global, e é ilustrada na Figura 15.

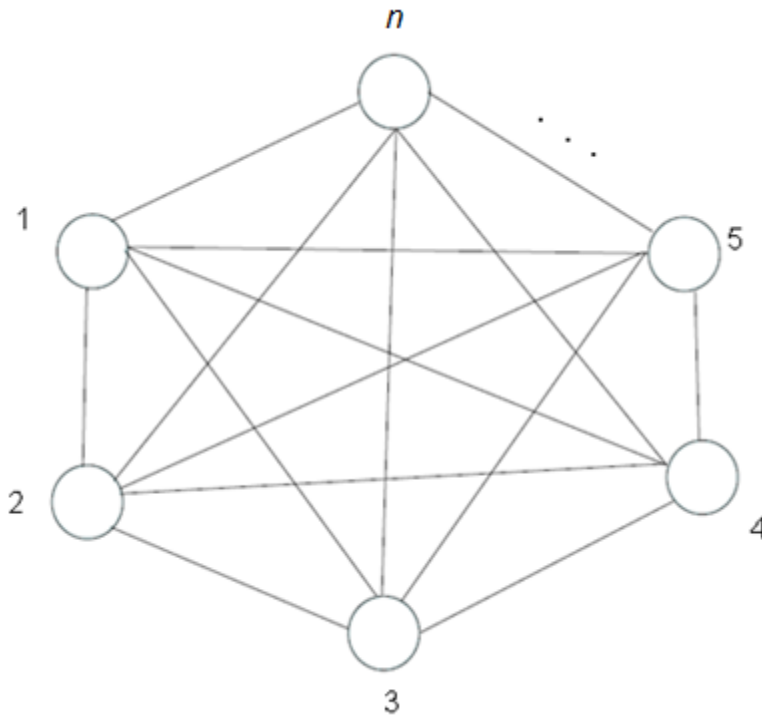


Figura 15: Topologia de estrela do PSO.

A utilização de outras topologias tem como objetivo outras interações sociais, como por exemplo, a topologia em anel, também conhecida como topologia local [48][49].

O pseudocódigo do algoritmo PSO se encontra no Anexo A.

4.1.1. ALGORITMO DE ENXAME DE PARTÍCULAS DINÂMICO

Uma modificação bem sucedida do algoritmo PSO implementada por diversos autores é tornar o peso de inércia dinâmico, sendo que um modo de variar o parâmetro ω é iniciando-o com o valor de 0,9 e linearmente decaindo para 0,4 [46][47]. Desta forma o algoritmo priorizará inicialmente uma busca global, eliminando possíveis mínimos locais, e posteriormente uma busca local, para alcançar o mínimo global. Esse decaimento linear pode ser descrito pela seguinte equação (4.7),

$$\omega(it) = \frac{\omega_2 - \omega_1}{\omega_e - 1} (it - 1) + \omega_2, \quad (4.7)$$

onde $\omega(it)$ é o peso de inércia atual, ω_2 é o valor final do peso de inércia, ω_1 é o valor inicial do peso de inércia, ω_e é a iteração onde peso de inércia deve ter o menor valor, e it é a iteração atual do algoritmo.

O pseudocódigo do algoritmo denominado aqui de DPSO, ou em inglês, *Dynamic Particle Swarm Optimization*, se encontra no Anexo A.

A Figura 16 mostra um exemplo de ω onde $\omega_1 = 0,9$, $\omega_2 = 0,4$ e $\omega_e = 700$.

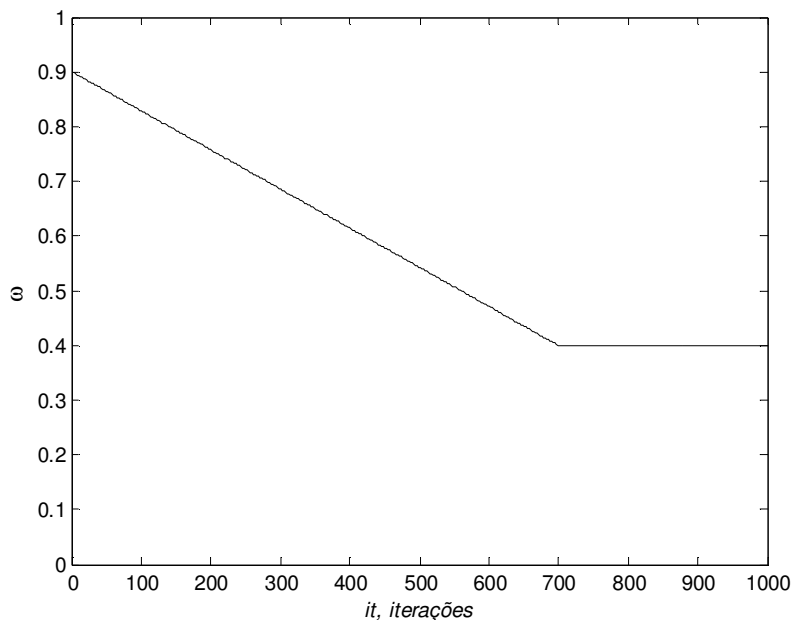


Figura 16: Exemplo de peso de inércia obtido através da equação (4.7) com os parâmetros $\omega_1 = 0,9$, $\omega_2 = 0,4$ e $\omega_e = 700$.

4.1.2. ALGORITMO DE ENXAME DE PARTÍCULAS COM FUNÇÃO GAMA

A distribuição gama pode não ser normalmente empregada, como a distribuição normal, mas possui diversas aplicações práticas, como a modelagem da quantidade de chuva diária em uma região [50][51][52], para ajustar conjuntos de dados hidrológicos [51], na segmentação de fonemas [53], na modelagem da quantidade de pagamento de dividendos nas reivindicações dos acionistas de uma

companhia de seguros [54], na previsão da distribuição dos tamanhos das gotas de um cabeçote de impressão [55],

A função gama incompleta pode ser representada por (4.8).

$$\begin{aligned} \Gamma_1(z_\Gamma, \alpha_\Gamma) &= \int_{\alpha}^{\infty} t^{z-1} e^{-t} dt , \\ \Upsilon_1(z_\Gamma, \alpha_\Gamma) &= \int_0^{\alpha} t^{z-1} e^{-t} dt , \end{aligned} \tag{4.8}$$

onde z_Γ deve ser maior que 0. Se $\alpha_\Gamma = 0$, então, $\Gamma(z_\Gamma, 0) = \Gamma(z_\Gamma)$ que é a função gama. Já quando $z_\Gamma = 0$, então, $\Gamma(0, \alpha_\Gamma) = E(\alpha_\Gamma)$ que é a integral exponencial [56][57].

Uma segunda representação é apresentada em (4.9).

$$\begin{aligned} \Gamma_2(z_\Gamma, \alpha_\Gamma) &= \frac{1}{\Gamma(z_\Gamma)} \int_{\alpha}^{\infty} t^{z-1} e^{-t} dt , \\ \Upsilon_2(z_\Gamma, \alpha_\Gamma) &= \frac{1}{\Gamma(z_\Gamma)} \int_0^{\alpha} t^{z-1} e^{-t} dt , \end{aligned} \tag{4.9}$$

com,

$$\Gamma_2 + \Upsilon_2 = 1 . \tag{4.10}$$

A representação (4.9) é utilizada para a resolução de problemas de matemática estatística e teoria de probabilidade, devido à propriedade definida pela equação (4.10) [58]. O *software* MATLAB, utilizado neste trabalho, também faz uso da representação (4.9) para o cálculo da função gama incompleta, obtida a partir dos valores de Υ_2 .

Esta distribuição ainda possui poucas aplicações em algoritmos de otimização, sendo que nenhuma publicação empregando a distribuição gama à inteligência de enxames.

O cálculo da função gama incompleta para uma determinada sequência de valores pode gerar curvas com formatos interessantes aos algoritmos de otimização, mais especificamente, ao peso de inércia ω do algoritmo PSO. A Figura 17 mostra

os valores da função gama incompleta Y_2 dada pela equação (4.9), a partir de uma sequência de 1000 valores de z_r . A sequência decrescente está dentro do intervalo $[10\ 0,008]$, e o parâmetro α_r foi definido como $\alpha_r = 10$.

Tanto o parâmetro α_r , quanto o intervalo de z_r foram estimados empiricamente. Na mesma figura é possível observar a curva da função gama com um ajuste definido pela equação (4.11).

$$\begin{aligned}\omega_\gamma &= Y_2(z_r, \alpha_r) + k_\gamma, \\ k_\gamma &= \omega_2 - Y_2(z_r(\text{max_it}), \alpha_r),\end{aligned}\tag{4.11}$$

onde ω_γ é o peso de inércia utilizando a função gama incompleta, k_γ é uma constante calculada a partir do valor do peso de inércia desejado na etapa final da otimização, e $z_r(\text{max_it})$ é o último valor da sequência usada no cálculo.

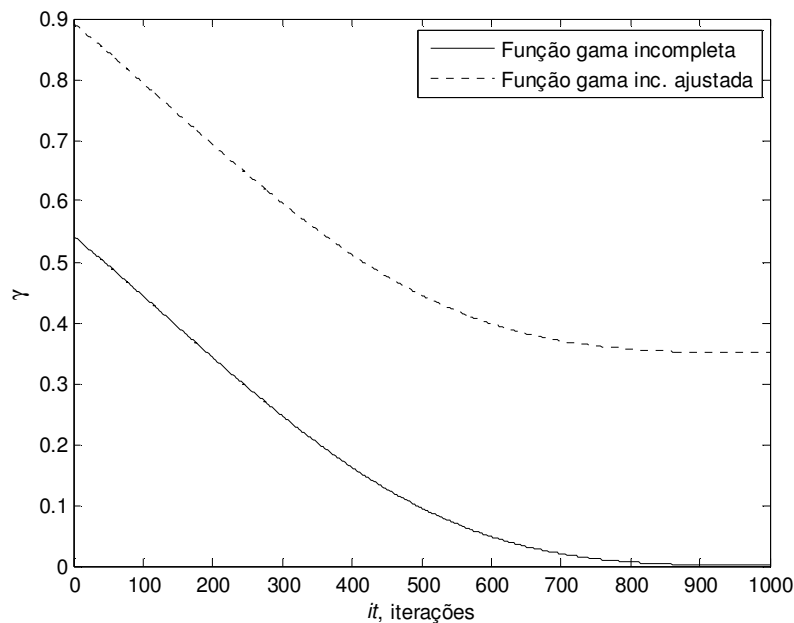


Figura 17: Curvas da função gama incompleta original e ajustada.

Para a curva ajustada mostrada na Figura 17, foi escolhido $\omega_2 = 0,35$, assim a curva tem um início próximo de 0,9, e um final próximo de 0,4. Este intervalo está de acordo com a afirmação de diversos autores citados anteriormente, como sendo um bom intervalo de funcionamento para o peso de inércia do PSO, garantindo inicialmente uma busca global, seguida de uma busca local, aumentando as chances de convergência do algoritmo.

Com isso, a equação (4.4) pode ser reescrita da seguinte maneira,

$$v_i(it + 1) = \omega_\gamma(it) \cdot v_i(it) + r_1 \cdot c_1 \cdot (lP_i(it) - \varphi_i(it)) + r_2 \cdot c_2 \cdot (gP(it) - \varphi_i(it)), \quad (4.12)$$

onde $\omega_\gamma(it)$ é o peso de inércia obtido pela função gama incompleta na iteração it .

O pseudocódigo do algoritmo *Gamma* PSO, ou GPSO, se encontra no Anexo A.

5. RESULTADOS

Diversos autores criaram métodos com o objetivo de melhorar o desempenho do algoritmo da máquina de aprendizagem extrema (ELM). Como afirma [59], desde que as ponderações de saída são calculadas de acordo com as ponderações de entrada e *bias*, e estes são escolhidos aleatoriamente, deve existir um conjunto de valores não-ótimos ou desnecessários para estes parâmetros. Isto faz com que o ELM necessite de mais neurônios ocultos que outros algoritmos de aprendizagem convencionais, o que pode fazer com que ele responda lentamente a conjuntos de dados desconhecidos. Para resolver este problema, [59] sugere a utilização do algoritmo de otimização baseado em evolução diferencial, sendo esta variação do algoritmo chamada de máquina de aprendizagem extrema evolucionária (E-ELM).

Outros autores também propuseram variações do algoritmo, como [60] que apresenta o ELM efetivo (EELM), onde é utilizado o critério da diagonal dominante para definição das ponderações de entrada e *bias* adequadas. Já [61] propõe um ELM com voto para problemas de classificação, onde em cada neurônio oculto existiria outro algoritmo ELM com o mesmo número de neurônios ocultos, as ponderações e *bias* seriam iniciados de forma aleatória para todos os neurônios, assim, existiriam mais neurônios para captar cada característica do problema, e o resultado da classe seria obtido por meio de votação entre os algoritmos ELM. Yuan, Wang e Cao fazem uma análise da equação (3.18) em [62], onde ela é substituída por outra, com o objetivo de melhorar a qualidade da solução do sistema linear, também discutem casos de *rank* cheio para a matriz H . O trabalho realizado em [63] explora a inteligência de enxame utilizando o algoritmo de procura em grupo para otimizar as ponderações iniciais e *bias*. Com o mesmo objetivo, [64] utiliza o algoritmo PSO para a otimização.

Neste capítulo serão apresentados os resultados obtidos na utilização do algoritmo PSO com peso de inércia $\omega = 0,4$, a variação do PSO com peso de inércia dinâmico, $\omega = 0,9 \rightarrow 0,4$. E também a variação proposta pelo autor empregando a função gama incompleta para gerar os valores do peso de inércia. Os três algoritmos irão otimizar as ponderações de entrada, *bias* e os diversos parâmetros adicionais da função TAF, implementada no algoritmo de aprendizagem extrema. O objetivo

será identificar as séries temporais de Box & Jenkins, Mackey-Glass e Lorenz, descritas no Capítulo 2, sendo que a aprendizagem será feita em batelada, conforme explicado anteriormente.

Um teste para comparar o desempenho dos modelos será realizado de forma recursiva, ou seja, os dados das previsões anteriores irão alimentar a rede. Este tipo de previsão também é conhecido como previsão N -passos-a-frente, sendo que a tendência do modelo é divergir devido ao erro acumulativo decorrente da realimentação citada anteriormente.

Também serão apresentados resultados obtidos utilizando o algoritmo ELM puro, com peso de entrada e *bias* aleatórios distribuídos uniformemente, e empregando o algoritmo de retropropagação para aprendizagem, ambos a título de comparação com a proposta apresentada neste trabalho.

A função objetivo escolhida para ser minimizada é a média da soma dos erros ao quadrado, em inglês, *Mean sum of Squares Error (MSE)*, dada pela seguinte equação,

$$MSE = \frac{1}{N} \sum_{i=1}^N (e_i)^2, \quad (5.1)$$

onde e_i é o erro de previsão da i -ésima amostra distinta, e N é o número de amostras distintas. Alterando a equação (5.1) e incluindo um termo que consiste na média da soma dos quadrados das ponderações, é possível aprimorar a generalização, pois a rede tenderá a selecionar ponderações menores, forçando-a a responder mais suavemente e com menos sobressinais. A nova função objetivo é definida pela equação (5.2),

$$MSEW = MSE + \frac{1}{\tilde{N}} \sum_{j=1}^{\tilde{N}} w_j^2, \quad (5.2)$$

onde w_j é a ponderação de saída do algoritmo ELM, e \tilde{N} é o número de neurônios ocultos.

Para o algoritmo PSO obter soluções de melhor qualidade para o problema de minimização do *fitness*, será adicionado à equação (5.2) mais um termo MSE, sendo que um deles será o MSE calculado a partir do erro de treinamento, o outro termo será o MSE calculado a partir do erro da validação do algoritmo, assim temos que a função objetivo, ou *fitness*,

$$fitness = (K)MSE_t + (1 - K) \frac{1}{\tilde{N}} \sum_{j=1}^{\tilde{N}} w_j^2, \quad (5.3)$$

onde MSE_t é o MSE do treinamento, K é uma ponderação na faixa entre $[0,1]$.

Adicionalmente ao cálculo do *fitness* será utilizado o parâmetro R^2 dado pela equação (5.4) como uma confirmação da qualidade da previsão. O parâmetro é definido como,

$$R^2 = 1 - \frac{\sum(Y_R - \hat{Y})^2}{\sum(Y_R - \bar{Y}_R)^2}, \quad (5.4)$$

onde \bar{Y}_R é a média da saída. O objetivo do parâmetro R^2 é alcançar valores próximos de “1”, ou seja, maximizá-lo, sendo que “1” indica que o modelo é perfeito e “0” indica que o modelo não foi capaz de reproduzir o comportamento do sistema.

Simulações realizadas com o PSO clássico e utilizando a série temporal de Box & Jenkins como exemplo, mostraram que o melhor valor para K é 0,9, conforme mostra a Tabela 5. Os parâmetros de teste escolhidos foram $0,1 \leq K \leq 0,9$ com passo 0,1 e o número de neurônios ocultos variando entre 5, 10, 30 e 50 neurônios, os parâmetros do algoritmo PSO foram um enxame composto por 20 partículas, $\omega = 0,4$, e $c_1 = c_2 = 2$, em cada experimento foram realizadas 1000 iterações do algoritmo, sendo que para cada teste do parâmetro K , foram realizados 10 experimentos.

Tabela 5: Teste de K para diferentes números de neurônios ocultos.

Neurônios Ocultos	K	<i>fitness</i>
5	0,9	0,002106
10	0,9	0,001711
30	0,9	0,000807
50	0,9	0,000727

Os resultados da identificação das séries temporais serão apresentados inicialmente para, as simulações empregando 5, 10, 30 e 50 neurônios na camada oculta do ELM, com a função de ativação SIG dada pela equação (3.24) com $c = 10$. Posteriormente serão apresentados os resultados utilizando a função de ativação TAF definida pela equação (3.26), juntamente com a função de ativação linear dada pela equação (3.25), combinando as duas funções no mesmo ELM.

Para a otimização das duas abordagens do ELM serão aplicados o algoritmo de enxame de partículas e suas variações apresentadas no Capítulo 4.

O algoritmo PSO clássico foi implementado com $\omega = 0,4$, $c_1 = c_2 = 2$, o PSO dinâmico (DPSO), com os parâmetros $\omega = 0,9 \rightarrow 0,4$, $w_e = 900$, $c_1 = c_2 = 2$. O GPSO foi elaborado de acordo com os parâmetros citados anteriormente no Capítulo 4.

O vetor de posição das partículas do algoritmo PSO e suas variantes foram ordenados da seguinte maneira para o ELM com a função SIG,

$$\varphi_i = [w_1, \dots, w_N, \beta_1, \dots, \beta_N], \quad (5.5)$$

e para o ELM com a função TAF,

$$\varphi_i = [w_1, \dots, w_N, \beta_1, \dots, \beta_N, \zeta_1, \dots, \zeta_N, \alpha_1, \dots, \alpha_N, \delta_1, \dots, \delta_N, \mu_1, \dots, \mu_N, k_{lin_1}, \dots, k_{lin_N}]. \quad (5.6)$$

As combinações de funções de ativação linear e não-linear podem ser observadas na Tabela 6, onde as proporções de neurônios lineares escolhidas para 5 neurônios ocultos foram de 20% e 80%, já para 10, 30 e 50 neurônios ocultos as proporções escolhidas foram de 10%, 50% e 90% de neurônios lineares.

O erro absoluto médio de previsão considerado aceitável para este trabalho será de até 5%. Para os testes de correlação, a fim de validar os modelos, a faixa de confiança, região onde a correlação não é significativa, escolhida será de 95%. Esta faixa de confiança é determinada pela seguinte equação [17],

$$conf_{95\%} = \pm \frac{1,96}{\sqrt{N}}, \quad (5.7)$$

onde N é o número de amostras.

Tabela 6: Número de neurônios lineares e não-lineares por camada de neurônios ocultos.

Neurônios Ocultos	Neurônios Lineares	Neurônios Não-lineares
5	1	4
	4	1
10	1	9
	5	5
	9	1
30	3	27
	15	15
	27	3
50	5	45
	25	25
	45	5

5.1. BOX & JENKINS

Os resultados das simulações para identificação e previsão do tipo um-passo-a-frente da série temporal de Box & Jenkins serão apresentados a seguir. Serão empregadas as funções de ativação SIG e TAF, a qual é a combinação da função de ativação variável com função de ativação linear.

Inicialmente serão discutidos os resultados utilizando as entradas sugeridas na literatura, quando também será apresentada uma análise correlacional do melhor resultado obtido pelo algoritmo de otimização. Posteriormente, estes serão comparados com os resultados obtidos após as correções sugeridas pela análise correlacional.

5.1.1. ESTUDO DE CASO ORIGINAL BOX & JENKINS

Os resultados obtidos empregando a função SIG podem ser observados na Tabela 7, onde os valores em negrito indicam o melhor valor entre os três métodos de otimização e o valor sublinhado indica o melhor resultado para este estudo de caso.

Tabela 7: Resultados de previsão da série Box & Jenkins com a função SIG.

Neurônios Ocultos	<i>Fitness</i>		
	PSO	DPSO	GPSO
5	$6,369 * 10^{-3}$	$6,056 * 10^{-3}$	$6,456 * 10^{-3}$
10	$2,847 * 10^{-3}$	$2,743 * 10^{-3}$	$2,769 * 10^{-3}$
30	$7,174 * 10^{-4}$	$7,151 * 10^{-4}$	$7,644 * 10^{-4}$
50	$3,959 * 10^{-4}$	$4,724 * 10^{-4}$	$4,281 * 10^{-4}$

O resultado das simulações utilizando a função de ativação TAF e a função de ativação linear em conjunto, conforme citado anteriormente na Tabela 6, podem ser observados na Tabela 8.

Tabela 8: Resultado para a série Box & Jenkins utilizando a função de ativação TAF.

Neurônios Ocultos	Neurônios Não-Lineares	PSO	DPSO	GPSO
5	1	3,099 * 10⁻⁴	3,155 * 10 ⁻⁴	3,113 * 10 ⁻⁴
	4	7,738 * 10⁻⁵	7,842 * 10 ⁻⁵	7,935 * 10 ⁻⁵
10	1	2,608 * 10 ⁻⁴	2,239 * 10⁻⁴	2,240 * 10 ⁻⁴
	5	6,909 * 10⁻⁵	6,988 * 10 ⁻⁵	6,998 * 10 ⁻⁵
	9	6,319 * 10⁻⁵	6,672 * 10 ⁻⁵	6,542 * 10 ⁻⁵
30	3	7,236 * 10 ⁻⁵	7,214 * 10⁻⁵	7,274 * 10 ⁻⁵
	15	4,780 * 10⁻⁵	4,907 * 10 ⁻⁵	4,791 * 10 ⁻⁵
	27	4,363 * 10 ⁻⁵	4,280 * 10 ⁻⁵	4,130 * 10⁻⁵
50	5	6,557 * 10⁻⁵	6,655 * 10 ⁻⁵	6,558 * 10 ⁻⁵
	25	4,370 * 10 ⁻⁵	4,022 * 10⁻⁵	4,107 * 10 ⁻⁵
	45	8,594 * 10 ⁻⁵	4,491 * 10 ⁻⁵	4,208 * 10⁻⁵

Analisando a Tabela 7 e a Tabela 8 pode-se concluir que para este estudo de caso o algoritmo PSO obteve o maior número de ‘melhores soluções’, com 7 vitórias, contra 6 vitórias do DPSO e 2 vitórias do GPSO.

O melhor resultado obtido foi através do algoritmo DPSO empregando 50 neurônios ocultos, com a função de ativação TAF combinada com a função linear, sendo 25 neurônios utilizando cada função. A Tabela 9 mostra os detalhes estatísticos pertinentes da simulação com melhor *fitness*.

Tabela 9: Dados estatísticos para simulação com a série de Box & Jenkins e função TAF com as entradas originais.

Melhor <i>fitness</i> :	$3,332 * 10^{-5}$
Média do <i>fitness</i> em 30 exp.:	$4,022 * 10^{-5}$
Desvio Padrão:	$3,466 * 10^{-6}$
R ² Treinamento:	0,999
R ² Validação:	0,951
Desvio Padrão do Erro de Treinamento:	$6,058 * 10^{-3}$
Desvio Padrão do Erro de Validação:	$3,618 * 10^{-2}$
MSE Treinamento:	0,254
MSE Validação:	0,429

A partir da Tabela 9 é possível observar que o parâmetro R² do conjunto de validação ficou próximo de “1”, indicando que o modelo identificado ficou próximo do modelo real, os gráficos referentes ao melhor experimento serão mostrados a seguir.

A Figura 18 mostra a convergência do algoritmo DPSO para o melhor experimento e também a convergência média durante os 30 experimentos, pode-se ressaltar que o algoritmo DPSO produziu boas soluções para a maioria dos experimentos, com a curva média ficando próxima da curva da melhor solução.

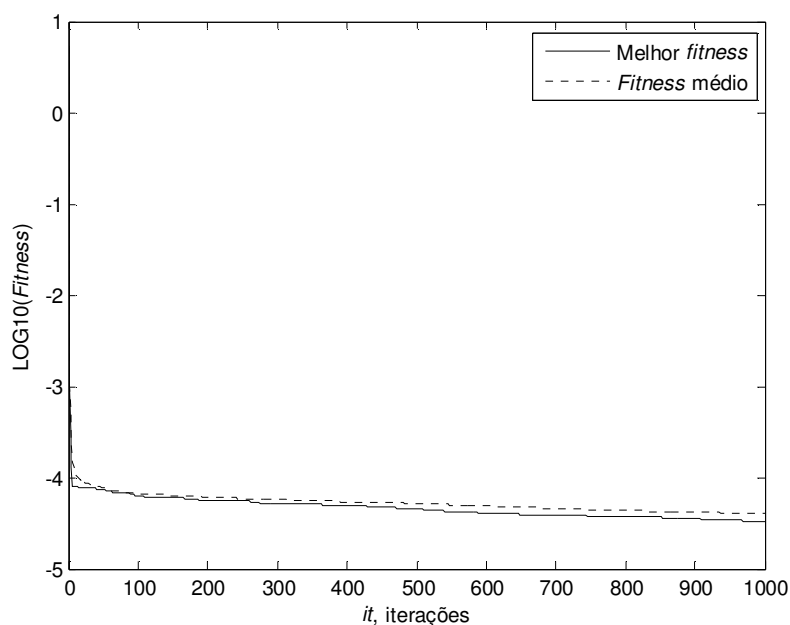


Figura 18: Curva de convergência do algoritmo DPSO para a série de Box & Jenkins.

A Figura 19 apresenta a previsão e o sinal de erro de previsão para o conjunto de treinamento.

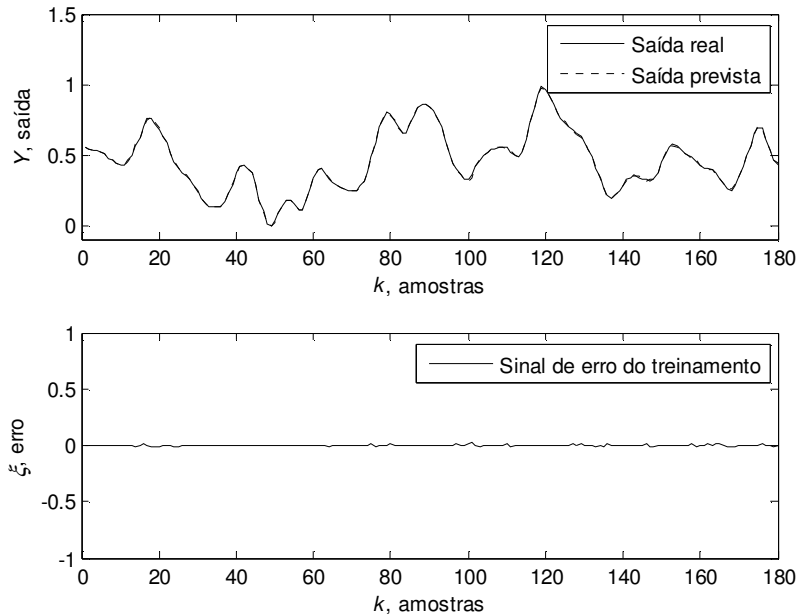


Figura 19: Previsão e sinal de erro para o conjunto de treinamento da série de Box & Jenkins utilizando a função de ativação TAF.

Ao observar a Figura 19, pode-se ressaltar que o conjunto de parâmetros otimizados pelo algoritmo DPSO, em conjunto com o algoritmo ELM e a função TAF, foram capazes de identificar as dinâmicas dos dados de treinamento com êxito.

A Figura 20 mostra a previsão e o sinal de erro para o conjunto de dados escolhido para a validação do modelo.

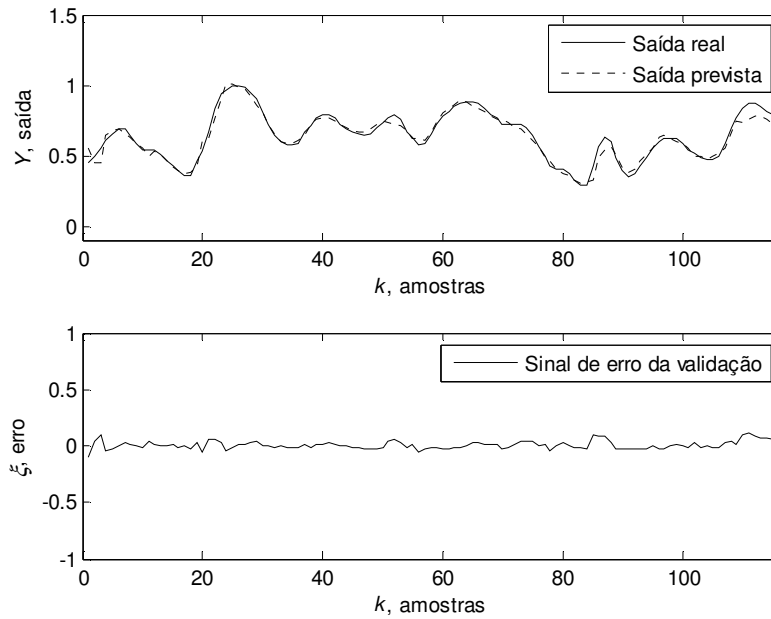


Figura 20: Previsão e sinal de erro para o conjunto de validação da série de Box & Jenkins com a função de ativação TAF.

Ao observar a Figura 20 é possível reparar que o modelo obteve um desempenho razoável, mesmo não prevendo todas as dinâmicas do sistema, fato esse comprovado pelo valor do parâmetro R^2 apresentado na Tabela 9. O erro absoluto máximo foi na 111ª amostra com uma amplitude de 11,4%, apesar disto, o erro absoluto médio foi de 2,75%, dentro da faixa considerada aceitável.

A Figura 21 mostra a previsão do tipo N -passos-a-frente, onde N é o número de amostras do conjunto de validação. A Tabela 10 apresenta os dados estatísticos referentes a esta previsão.

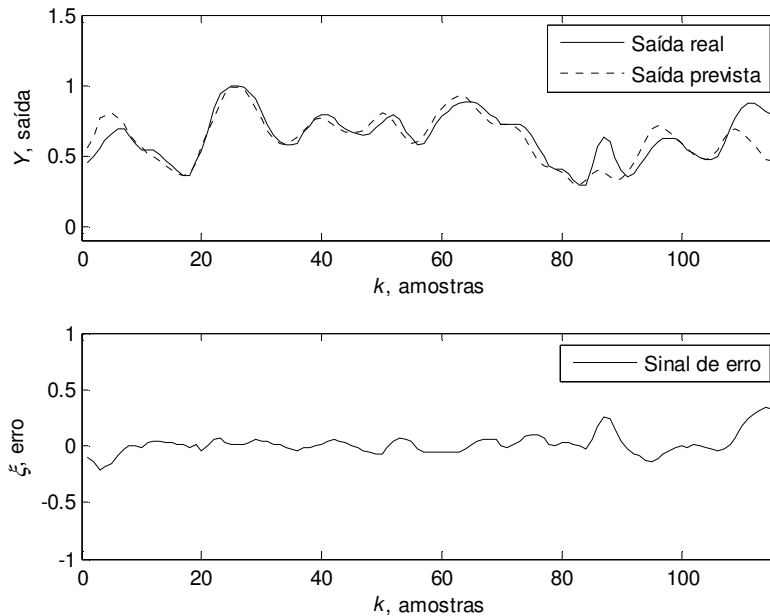


Figura 21: Previsão e sinal de erro da previsão do tipo N -passos-a-frente para a série de Box & Jenkins.

Tabela 10: Dados estatísticos referentes à previsão do tipo N -passos-a-frente para a série de Box & Jenkins.

R^2 :	0,656
MSE:	$9,649 * 10^{-3}$
Desvio Padrão do Sinal de Erro:	$9,694 * 10^{-2}$

É possível observar pela comparação do parâmetro R^2 que o modelo teve um desempenho inferior quando requisitado para uma previsão com um horizonte mais longo. O erro absoluto médio foi de 6,39%, no entanto o pico foi de 34,3%, ambos acima do limite aceitável.

Ao se analisar o sinal de erro ξ através dos testes de correlação apresentados no Capítulo 3, são traçadas os gráficos que podem ser observados no Anexo B. A partir das interpretações sugeridas por [17][33][34][35], onde valores positivos no eixo das ordenadas indicam uma correlação positiva com um atraso que é indicado no eixo das abscissas. Assim, as entradas escolhidas para serem adicionadas são $x(k-1)y(k-1)$ e $x(k-1)x(k-2)$ devido ao pico em $\tau = -1$

identificado na Figura 41 e na Figura 45, referentes aos testes de autocorrelação e correlação cruzada entre x e ξ , e x^2 e ξ .

Com as novas entradas o modelo deixa de ser do tipo ARX, para se tornar um modelo do tipo NARX.

Da Figura 48 até a Figura 56 são mostrados os novos gráficos de autocorrelação e correlação cruzada, confirmando a queda das correlações para valores próximos ou dentro da faixa de confiança determinada pela equação (5.7). Apenas as correlações referentes à entrada $y(k-1)x(k-1)$ apresentadas na Figura 52 exibiram valores significativamente acima da faixa de confiança.

A fim de comparar o método de aprendizagem proposto com o método original, e o método com maior popularidade, no caso, o método de retropropagação. Para este estudo de caso foram realizados dez experimentos com o ELM original, com ponderações escolhidas aleatoriamente com distribuição uniforme, função de ativação SIG. Da mesma forma foram realizados dez experimentos utilizando o método de retropropagação com 20000 iterações para treinar a rede neural. Ambas as redes empregam 50 neurônios ocultos. O número de iterações escolhido para o método de retropropagação foi de acordo com o número de avaliações da função objetivo num único experimento do algoritmo PSO para este trabalho, que é de 20000.

A Figura 22 apresenta a previsão e o sinal de erro utilizando o método de retropropagação. O erro absoluto máximo foi 48,7%, enquanto o erro absoluto médio foi 8,62%, ambos acima da faixa aceitável.

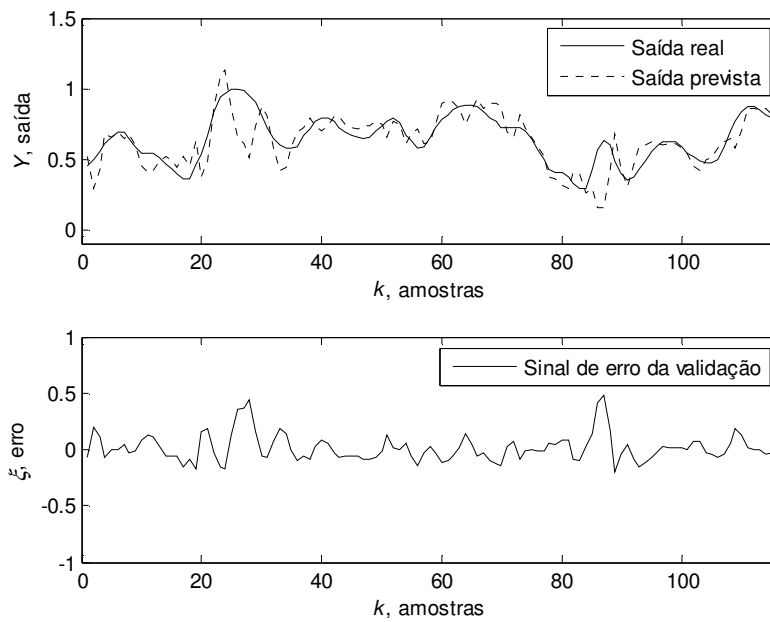


Figura 22: Previsão e sinal de erro utilizando o método de retropropagação para a série de Box & Jenkins.

A Figura 23 apresenta a previsão e o sinal de erro do algoritmo ELM original.

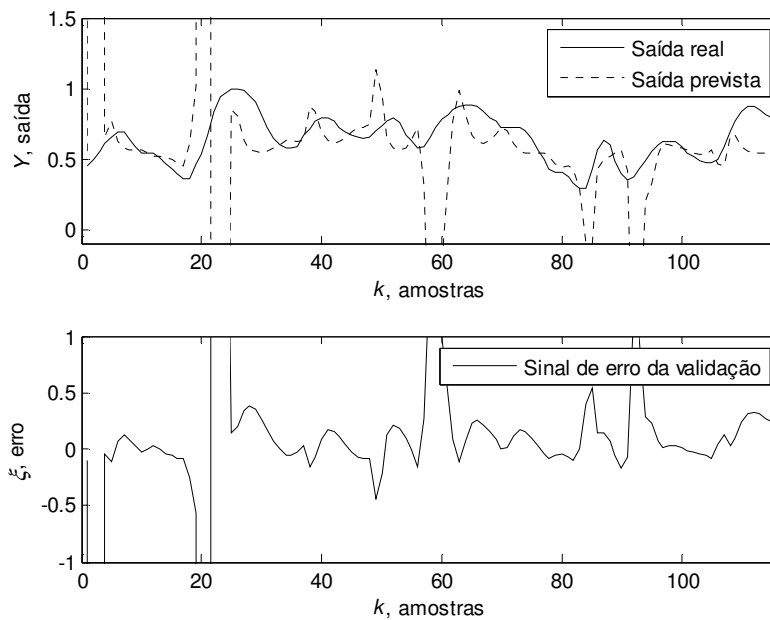


Figura 23: Previsão e sinal de erro utilizando o algoritmo ELM original para a série de Box & Jenkins.

Pode-se notar que o algoritmo ELM original não foi capaz de prever a dinâmica do sistema, provavelmente com ponderações que fazem parte do grupo de soluções não-ótimas do espaço possível de busca.

A Tabela 11 traz algumas informações comparativas entre todos os métodos.

Tabela 11: Comparação entre o algoritmo ELM original, método de retropropagação e ELM otimizado pelo algoritmo PSO, para a série de Box & Jenkins.

	ELM	Retroprop.	ELM+PSO
Tempo:	$7,08 * 10^{-3}s$	176,24s	107s
R^2 :	$-1,53 \equiv 0$	0,467	0,931
Média do erro absoluto:	613%	8,62%	3,02%
Máx. erro absoluto:	$7,062 * 10^4\%$	68%	20,9%

Nota-se que o algoritmo ELM original possui um tempo de processamento pequeno, mas produz soluções de baixa qualidade. Isto é devido ao uso de parâmetros não-ótimos.

A união do tempo de processamento baixo do ELM, com a busca poderosa do algoritmo PSO faz com que o ELM+PSO produza soluções de qualidade superior ao método de retropropagação, em menos tempo.

5.1.2. ESTUDO DE CASO MODIFICADO BOX & JENKINS

As simulações empregando as novas entradas $x(k-1)y(k-1)$ e $x(k-1)x(k-2)$ escolhidas a partir dos testes de correlação citados anteriormente serão realizadas somente pelo algoritmo DPSO que apresentou o melhor desempenho entre os 3 algoritmos de otimização. Foram realizados 30 experimentos e os parâmetros foram escolhidos conforme as simulações anteriores.

A função de ativação foi escolhida de acordo com o mesmo critério e será a TAF com 50 neurônios ocultos, sendo que 50% deles serão não-lineares e o restante lineares.

A Tabela 12 mostra os dados estatísticos referentes aos experimentos realizados.

Tabela 12: Dados estatísticos referentes aos experimentos com as novas entradas para a série de Box & Jenkins utilizando a função TAF.

Melhor <i>fitness</i> :	$3,421 * 10^{-5}$
Média do <i>fitness</i> em 30 exp.:	$3,902 * 10^{-5}$
Desvio Padrão:	$3,012 * 10^{-6}$
R ² Treinamento:	0,999
R ² Validação:	0,963
Desvio Padrão do Erro de Treinamento:	$6,143 * 10^{-3}$
Desvio Padrão do Erro de Validação:	$3,188 * 10^{-2}$
MSE Treinamento:	0,254
MSE Validação:	0,432

Comparando a Tabela 12 com a Tabela 9 nota-se que apesar do melhor *fitness* ter sido obtido com as entradas originais, a melhor média dos 30 experimentos foi alcançada utilizando as novas entradas, assim como o R² para o conjunto de validação superior ao modelo anterior, verificando que o novo modelo obteve um desempenho superior em relação ao anterior. O desvio padrão do erro de validação inferior do novo modelo significa que existe menos variação do sinal de erro em relação à média do erro, sendo mais um ponto positivo para o novo modelo.

A Figura 24 mostra a previsão e o sinal de erro da previsão para o conjunto de dados de treinamento da série Box & Jenkins empregando as novas entradas. Pode-se observar que o modelo foi capaz de identificar com êxito as dinâmicas do conjunto de dados de treinamento.

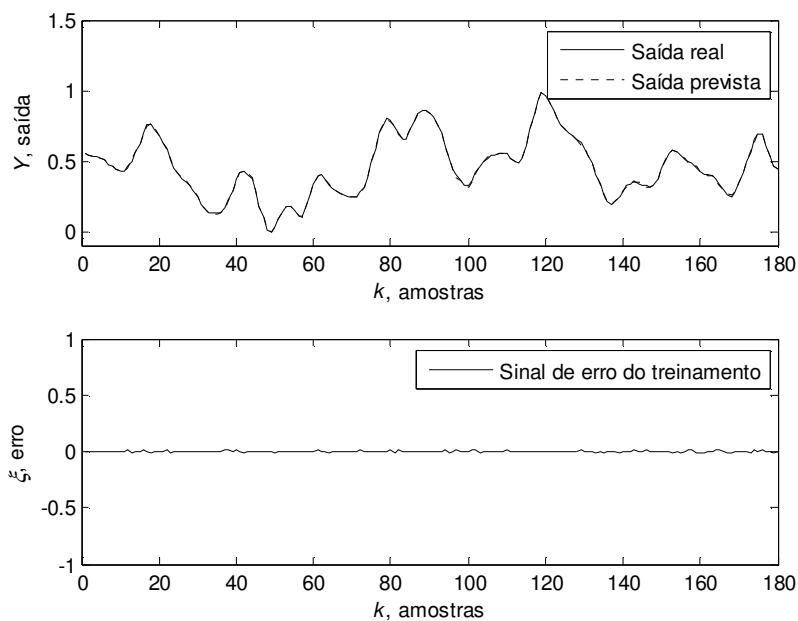


Figura 24: Previsão e sinal de erro para o conjunto de treinamento da série Box & Jenkins empregando as novas entradas.

A Figura 25 mostra a previsão e o sinal de erro para o conjunto de dados de validação empregando as novas entradas.

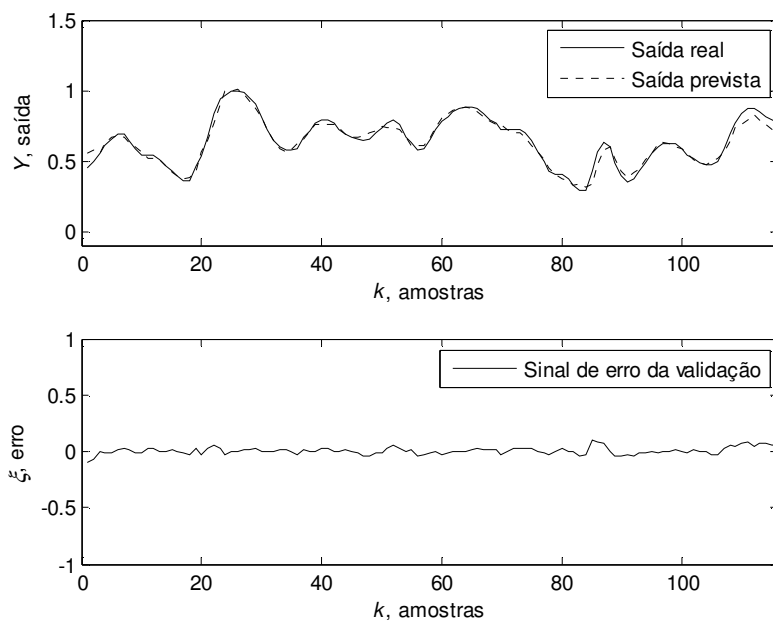


Figura 25: Previsão e sinal de erro para o conjunto de validação da série Box & Jenkins empregando as novas entradas.

Ao comparar a Figura 25 com a Figura 20 observa-se visualmente que o sinal de erro possui amplitudes menores, mostrando que as novas entradas melhoraram o comportamento do modelo em relação ao anterior. A amplitude máxima do erro foi 10%, sendo que o erro absoluto médio foi 2,35%, dentro da faixa considerada aceitável.

A Figura 26 mostra a previsão do tipo N -passos-a-frente empregando o novo modelo.

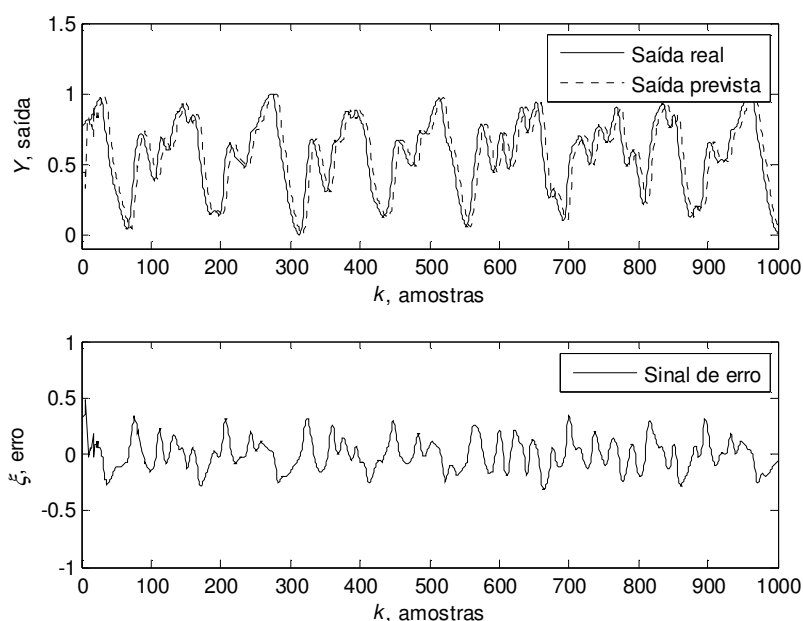


Figura 26: Previsão do tipo N -passos-a-frente e sinal de erro utilizando o modelo com as novas entradas para a série Box & Jenkins.

A Tabela 13 expõe os dados estatísticos pertinentes à previsão N -passos-a-frente. Ao comparar a Tabela 13 com a Tabela 10 percebe-se que o novo modelo empregando as duas novas entradas obteve um desempenho superior tanto no parâmetro R^2 , quanto no MSE. Estes fatos colaboram em comprovar a superioridade do novo modelo também em previsões com horizontes longos. O erro absoluto máximo foi 32,3%, abaixo do modelo anterior. O erro absoluto médio foi de 5,92%, próximo da faixa considerada aceitável.

Tabela 13: Dados estatísticos referentes à previsão N -passos-a-frente utilizando o modelo com as novas entradas para a série de Box & Jenkins.

R^2 :	0,703
MSE:	$8,311 * 10^{-3}$
Desvio Padrão do Sinal de Erro:	$8,784 * 10^{-2}$

5.2. MACKEY-GLASS

A seguir serão apresentados os resultados para a previsão do tipo um-passo-a-frente da série temporal de Mackey-Glass obtida a partir da equação (2.1) e com os parâmetros citados também no Capítulo 2. Logo após serão apresentados os resultados após as análises de correlação do melhor resultado obtido pelos algoritmos de otimização. Inicialmente serão analisados os resultados com a função SIG, logo em seguida, os resultados com a função TAF.

5.2.1. ESTUDO DE CASO ORIGINAL MACKEY-GLASS

A Tabela 14 mostra os resultados das simulações com a função SIG para os diferentes números de neurônios ocultos.

Tabela 14: Resultado da otimização para a série Mackey-Glass e função SIG.

Neurônios Ocultos	<i>Fitness</i>		
	PSO	DPSO	GPSO
5	$1,328 * 10^{-2}$	$1,327 * 10^{-2}$	$1,343 * 10^{-2}$
10	$8,126 * 10^{-3}$	$8,098 * 10^{-3}$	$7,909 * 10^{-3}$
30	$4,302 * 10^{-3}$	$4,712 * 10^{-3}$	$4,557 * 10^{-3}$
50	$3,430 * 10^{-3}$	$4,046 * 10^{-3}$	$3,776 * 10^{-3}$

A Tabela 15 exibe a média dos experimentos de cada simulação aplicando a função de ativação TAF.

Tabela 15: Resultados das otimizações para a série de Mackey-Glass com a função TAF.

Neurônios Ocultos	Neurônios Não-Lineares	PSO	DPSO	GPSO
5	1	$7,179 * 10^{-3}$	$7,200 * 10^{-3}$	$7,176 * 10^{-3}$
	4	$5,176 * 10^{-3}$	$5,328 * 10^{-3}$	$5,300 * 10^{-3}$
10	1	$7,071 * 10^{-3}$	$7,071 * 10^{-3}$	$7,079 * 10^{-3}$
	5	$4,516 * 10^{-3}$	$5,566 * 10^{-3}$	$4,539 * 10^{-3}$
	9	$4,168 * 10^{-3}$	$4,401 * 10^{-3}$	$4,364 * 10^{-3}$
30	3	$5,191 * 10^{-3}$	$5,260 * 10^{-3}$	$5,234 * 10^{-3}$
	15	$3,332 * 10^{-3}$	$3,539 * 10^{-3}$	$3,406 * 10^{-3}$
	27	$3,320 * 10^{-3}$	$3,133 * 10^{-3}$	$3,063 * 10^{-3}$
50	5	$4,539 * 10^{-3}$	$4,566 * 10^{-3}$	$4,533 * 10^{-3}$
	25	$3,460 * 10^{-3}$	$3,116 * 10^{-3}$	<u>$3,034 * 10^{-3}$</u>
	45	$8,982 * 10^{-3}$	$5,236 * 10^{-3}$	$5,195 * 10^{-3}$

Examinando a Tabela 14 e a Tabela 15, conclui-se que para este estudo de caso a algoritmo PSO obteve novamente o maior número de “melhores soluções”, com 8 vitórias, contra, 2 vitórias para o DPSO e 5 vitórias para o algoritmo GPSO.

Apesar do maior número de vitórias, a melhor solução obtida entre todas as simulações foi alcançada pelo algoritmo GPSO empregando 50 neurônios ocultos, sendo que 50% dos neurônios empregam a função de ativação TAF, e o restante emprega a função linear.

A Figura 27 mostra a curva de convergência do algoritmo GPSO para melhor solução e a curva de convergência média dos 30 experimentos. Pode-se concluir pela proximidade das curvas que o algoritmo GPSO convergiu para valores próximos do melhor *fitness* na maioria dos experimentos.

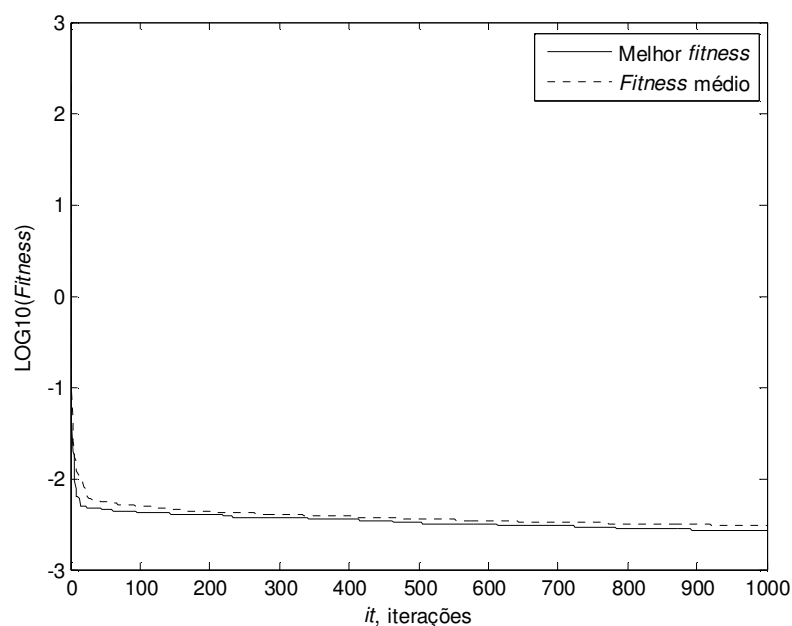


Figura 27: Curvas de convergência média e do melhor experimento do algoritmo GPSO para a série de Mackey-Glass empregando a função TAF.

A Tabela 16 apresenta os dados estatísticos referentes à melhor simulação descrita acima.

Tabela 16: Dados estatísticos referentes à melhor simulação para a série de Mackey-Glass utilizando a função TAF.

Melhor <i>fitness</i> :	$2,662 * 10^{-3}$
Média do <i>fitness</i> em 30 exp.:	$3,034 * 10^{-3}$
Desvio Padrão:	$2,782 * 10^{-4}$
R^2 Treinamento:	0,956
R^2 Validação:	0,824
Desvio Padrão do Erro de Treinamento:	$5,415 * 10^{-2}$
Desvio Padrão do Erro de Validação:	0,107
MSE Treinamento:	0,376
MSE Validação:	0,377

Observando a Tabela 16, mais precisamente o parâmetro R^2 , constata-se que o modelo identificado foi capaz de prever razoavelmente a dinâmica caótica do sistema. Este fato pode ser comprovado visualmente ao observar o sinal de erro da

previsão para o conjunto de treinamento e o conjunto de validação nas figuras abaixo.

A Figura 28 mostra a previsão e o sinal de erro para o conjunto de dados de treinamento, já a Figura 29 apresenta a previsão e o sinal de erro para o conjunto de dados de validação. Observando a Figura 28 e a Figura 29, é possível contemplar o fato ressaltado anteriormente, onde o modelo responde de forma parecida para ambos os conjuntos de dados, gerando sinais de erro absolutos com médias próximas. A média do erro absoluto de treinamento foi de 4%, dentro da faixa aceitável, e 6,9% para o erro absoluto de validação.

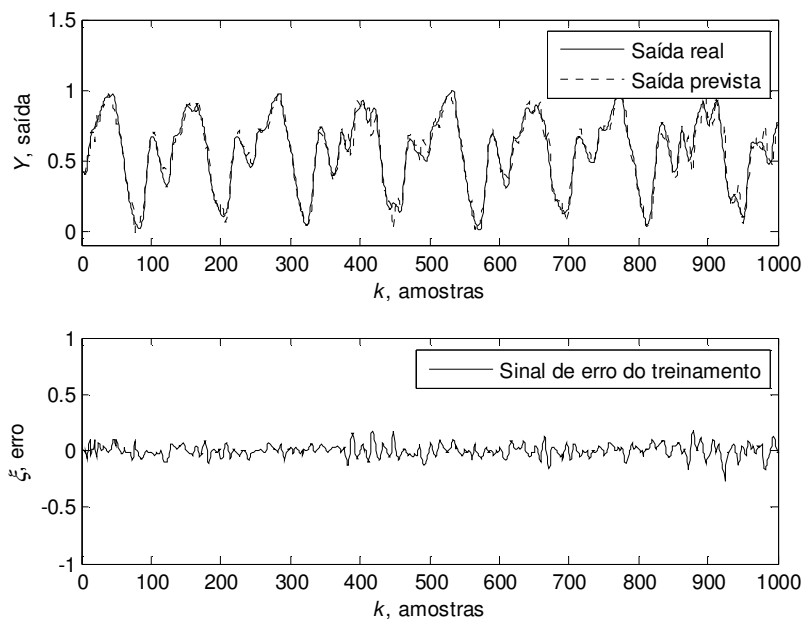


Figura 28: Previsão e sinal de erro utilizando a função TAF para o conjunto de treinamento da série de Mackey-Glass.

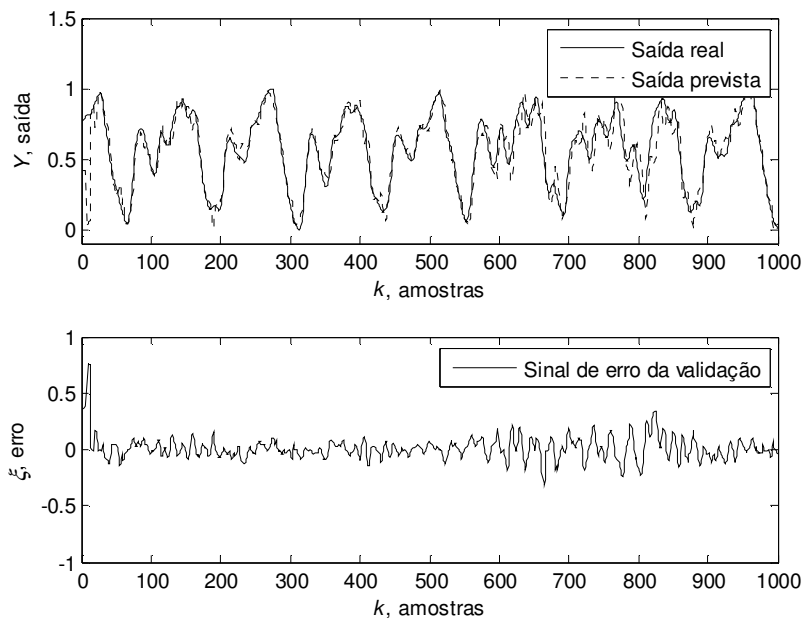


Figura 29: Previsão e sinal de erro empregando a função TAF para o conjunto de validação da série de Mackey-Glass.

A Figura 30 apresenta a previsão do tipo N -passos-a-frente utilizando o modelo obtido pelo melhor experimento.

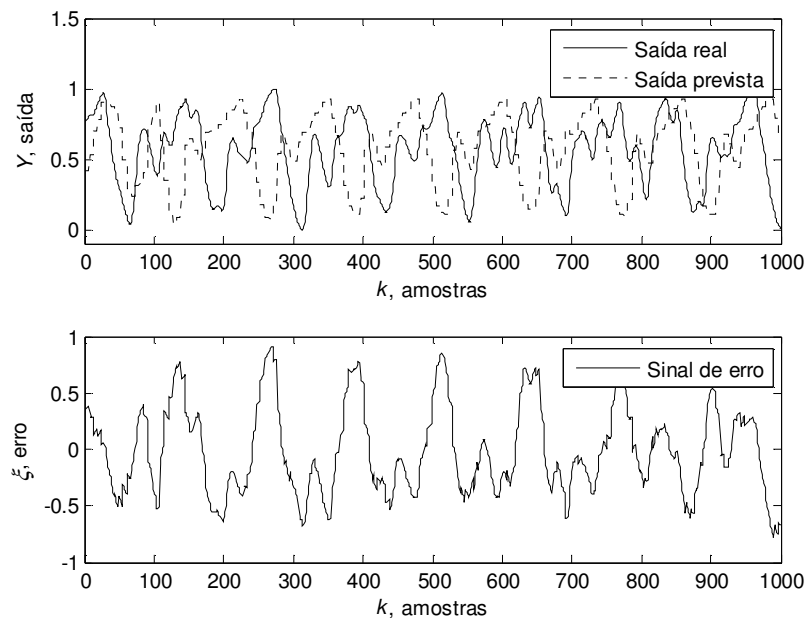


Figura 30: Previsão do tipo N -passos-a-frente e sinal de erro utilizando a função TAF e o conjunto de dados de validação da série de Mackey-Glass.

Ao observar a Figura 30, é possível concluir que o modelo não foi capaz de prever a dinâmica do sistema quando o mesmo é realimentado com as saídas previstas anteriormente, acumulando o erro a cada amostra e contribuindo para que o modelo não tenha um desempenho satisfatório.

A Tabela 17 abaixo reafirma esta conclusão com o parâmetro R^2 negativo, sendo que valores negativos de R^2 são interpretados como sendo iguais a 0, significando que o modelo não foi capaz de identificar as dinâmicas do sistema.

Tabela 17: Dados estatísticos referentes à previsão do tipo N -passos-a-frente para a série de Mackey-Glass.

R^2 :	$-1,39 \equiv 0$
MSE:	0,158
Desvio Padrão do Sinal de Erro:	0,397

Aplicando os testes de autocorrelação e correlação cruzada no resíduo ξ da previsão do tipo um-passo-a-frente, são traçados os gráficos apresentados entre a Figura 57 e a Figura 61 no Anexo B, para as respectivas entradas citadas no Capítulo 2.

Ao avaliar a Figura 57, os gráficos de autocorrelação e autocorrelação de alta ordem estão fora da faixa de confiança até a 6^a amostra.

A Figura 58 apresenta os gráficos de correlação cruzada, onde é possível observar correlações positivas a partir de aproximadamente $\tau = -4$.

A partir das duas observações, foram escolhidas as seguintes novas entradas, $x(k - 11)$, $e(k - 5)$ e $e(k - 6)x(k - 6)$. Com as novas entradas, o modelo passa do tipo AR, para um modelo do tipo NARMA.

Da Figura 62 até a Figura 69 são mostrados os gráficos de autocorrelação e correlação cruzada para o modelo obtido a partir das novas entradas.

A Figura 62 mostra os gráficos de autocorrelação, se comparado com a Figura 57, as novas entradas foram bem sucedidas ao diminuírem a correlação dos dados.

Ao comparar a Figura 58 com a Figura 63 também pode ser notada a melhora no desempenho do modelo, com os valores de correlação se mantendo dentro da faixa de confiança de 95%.

5.2.2. ESTUDO DE CASO MODIFICADO MACKEY-GLASS

As simulações com as novas entradas foram realizadas com o algoritmo de otimização GPSO, que obteve o melhor desempenho dentre os algoritmos. A rede neural será composta por 50 neurônios, sendo que 50% deles são não-lineares, utilizando a função TAF, o restante é composto por neurônios lineares. Foram realizados 30 experimentos, sendo que os dados referentes aos experimentos podem ser observados na Tabela 18.

Tabela 18: Dados estatísticos referentes à simulação com as novas entradas para a série de Mackey-Glass.

Melhor <i>fitness</i> :	$7,200 * 10^{-7}$
Média do <i>fitness</i> em 30 exp.:	$1,182 * 10^{-6}$
Desvio Padrão:	$3,665 * 10^{-7}$
R ² Treinamento:	0,999
R ² Validação:	0,997
Desvio Padrão do Erro de Treinamento:	$8,027 * 10^{-4}$
Desvio Padrão do Erro de Validação:	$1,372 * 10^{-2}$
MSE Treinamento:	0,379
MSE Validação:	0,388

Ao comparar a Tabela 18 com a Tabela 16, pode-se observar um decréscimo significativo na função objetivo e um crescimento no parâmetro R² do conjunto de validação, que ficou próximo de “1”. Estes indicativos mostram que as novas entradas melhoraram o desempenho do novo modelo em relação ao anterior.

A Figura 31 mostra a previsão e o sinal de erro para o conjunto de treinamento utilizando o novo modelo.

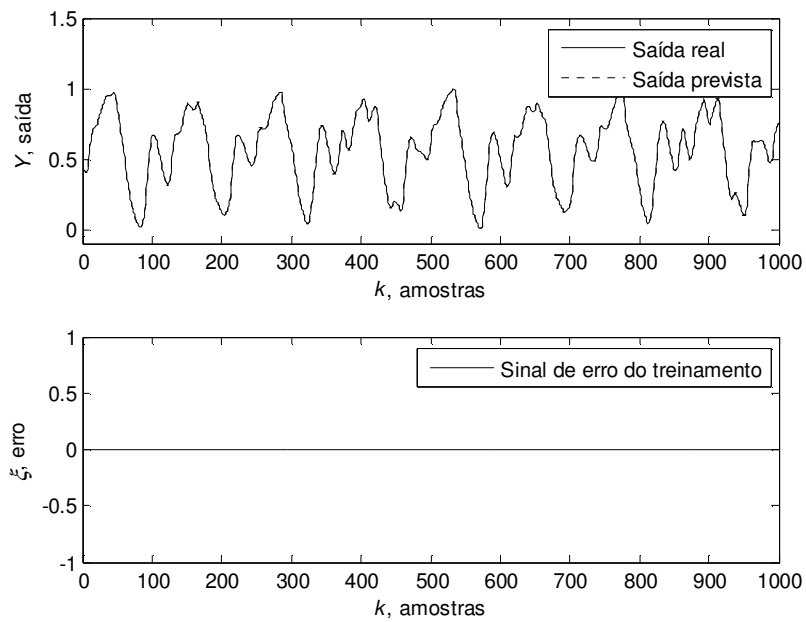


Figura 31: Previsão e sinal de erro utilizando a função TAF para o conjunto de treinamento da série de Mackey-Glass com as novas entradas.

Já a Figura 32 apresenta a previsão e o sinal de erro para o conjunto de validação, onde é possível reparar que logo na 6ª amostra há um pico no sinal de erro de aproximadamente 30%, apesar disso, a média do erro absoluto para o conjunto de validação aplicando o novo modelo foi de 0,2%, dentro da margem considerada aceitável de erro.

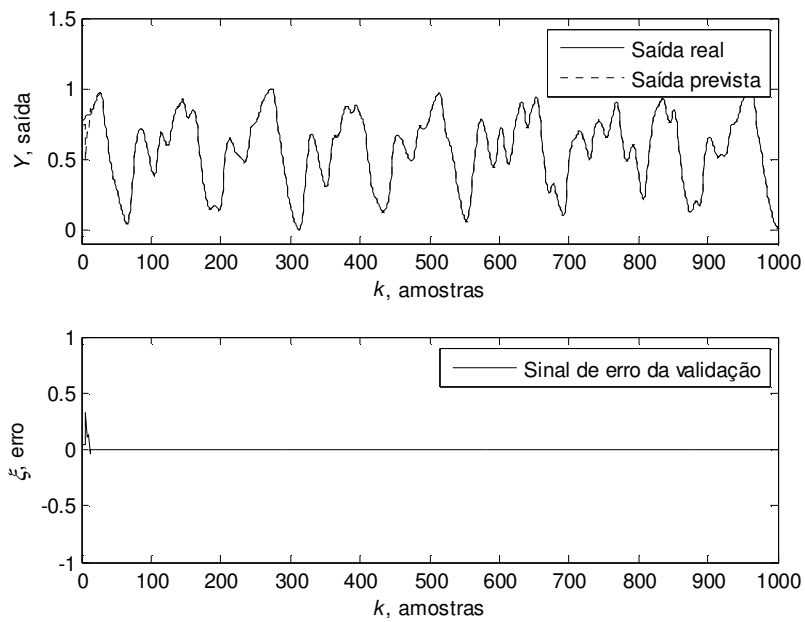


Figura 32: Previsão e sinal de erro empregando a função TAF para o conjunto de validação da série de Mackey-Glass com as novas entradas.

A Figura 33 traz a previsão do tipo N -passos-a-frente utilizando o novo modelo.

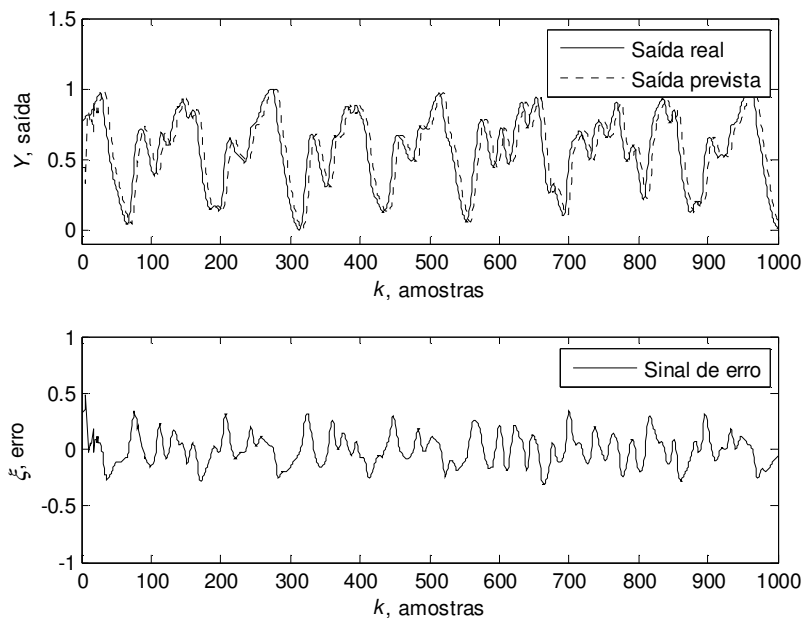


Figura 33: Previsão do tipo N -passos-a-frente e sinal de erro utilizando a função TAF para o conjunto de validação da série de Mackey-Glass com as novas entradas.

Ao observar a Figura 33 e a Figura 30 é possível concluir que as novas entradas melhoraram o desempenho do modelo também para horizontes de previsão mais longos. A Tabela 19 traz alguns dados estatísticos relevantes sobre a previsão.

Tabela 19: Dados estatísticos referentes à previsão do tipo *N*-passos-a-frente da série de Mackey-Glass utilizando as novas entradas.

R ² :	0,702
MSE:	0,019
Desvio Padrão do Sinal de Erro:	0,140

Ao observar o parâmetro R² da Tabela 19 em comparação com o mesmo parâmetro para o modelo original, percebe-se que o modelo com as novas entradas tornou-se capaz de prever certas dinâmicas do sistema, mesmo que com um desempenho apenas razoável, mas superior ao modelo anterior, que não foi capaz de identificar nenhuma dinâmica do sistema. O erro absoluto médio ainda foi superior ao limite aceitável, sendo de 11,3%.

5.3. SISTEMA DE LORENZ

Os resultados para a otimização com o objetivo de identificar e realizar a previsão do tipo um-passo-a-frente do sistema de Lorenz na coordenada *x*, conforme foi citado no Capítulo 2, empregando as funções de ativação SIG e TAF, se encontram a seguir. Posteriormente serão apresentados os resultados após a análise de correlação realizada a partir do melhor resultado obtido pelos algoritmos de otimização.

5.3.1. ESTUDO DE CASO ORIGINAL SISTEMA DE LORENZ

A Tabela 20 apresenta os valores de *fitness* das simulações utilizando neurônios com função de ativação SIG.

Tabela 20: Valores de *Fitness* para a série de Lorenz com função SIG.

Neurônios Ocultos	<i>Fitness</i>		
	PSO	DPSO	GPSO
5	$7,360 * 10^{-3}$	$7,071 * 10^{-3}$	$7,030 * 10^{-3}$
10	$3,551 * 10^{-3}$	$3,648 * 10^{-3}$	$3,482 * 10^{-3}$
30	$1,295 * 10^{-3}$	$1,314 * 10^{-3}$	$1,260 * 10^{-3}$
50	$8,092 * 10^{-4}$	$8,642 * 10^{-4}$	$8,275 * 10^{-4}$

A Tabela 21 contém os valores do *fitness* das simulações utilizando a função de ativação variável.

Tabela 21: Valores de *fitness* dos experimentos com a série de Lorenz e função TAF.

Neurônios Ocultos	Neurônios Não-Lineares	PSO	DPSO	GPSO
5	1	$2,281 * 10^{-4}$	$2,407 * 10^{-4}$	$2,509 * 10^{-4}$
	4	$8,821 * 10^{-5}$	$1,061 * 10^{-4}$	$9,227 * 10^{-5}$
10	1	$1,783 * 10^{-4}$	$1,727 * 10^{-4}$	$1,639 * 10^{-4}$
	5	$5,022 * 10^{-5}$	$4,707 * 10^{-5}$	$4,559 * 10^{-5}$
	9	$3,813 * 10^{-5}$	$4,093 * 10^{-5}$	$3,616 * 10^{-5}$
30	3	$6,243 * 10^{-5}$	$5,681 * 10^{-5}$	$5,987 * 10^{-5}$
	15	$1,728 * 10^{-5}$	$1,865 * 10^{-5}$	$1,565 * 10^{-5}$
	27	$1,605 * 10^{-5}$	$1,324 * 10^{-5}$	$1,181 * 10^{-5}$
50	5	$3,341 * 10^{-5}$	$3,264 * 10^{-5}$	$3,804 * 10^{-5}$
	25	$1,323 * 10^{-5}$	$1,151 * 10^{-5}$	$1,044 * 10^{-5}$
	45	$3,252 * 10^{-5}$	$6,928 * 10^{-6}$	$5,976 * 10^{-6}$

Constata-se pela Tabela 20 e pela Tabela 21 que o algoritmo que obteve o maior número de “melhores soluções” foi o GPSO, com 10 vitórias, em seguida o PSO com 3 vitórias, seguido do DPSO com 2 vitórias.

O algoritmo que obteve a melhor solução dentre todas foi também o GPSO, com a rede neural empregando 50 neurônios, sendo que 45 neurônios utilizando a função TAF e o restante utilizando a função linear.

A Figura 34 traz a curva de convergência da melhor solução e a curva de convergência média dos 30 experimentos realizados.

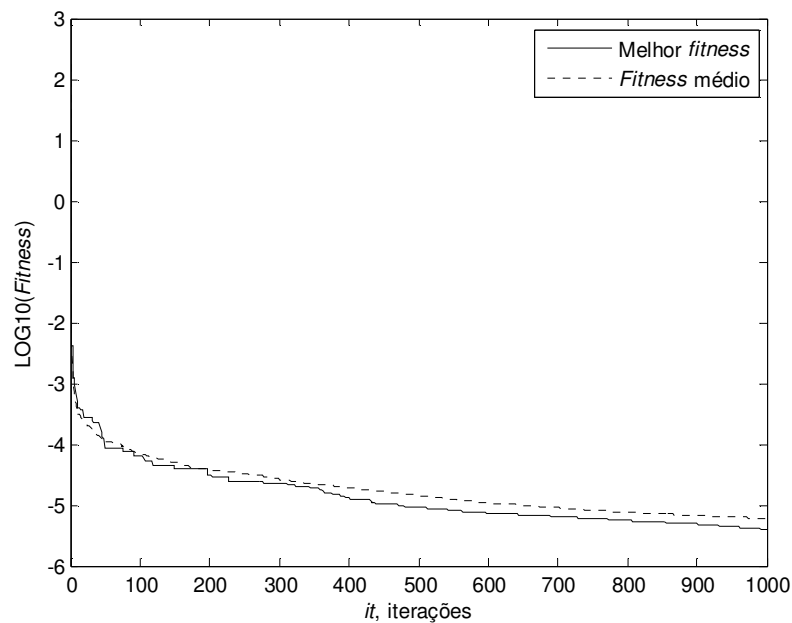


Figura 34: Curvas de convergência média e da melhor solução obtida pelo algoritmo GPSO para a série de Lorenz.

Os detalhes pertinentes à melhor solução são apresentados na Tabela 22.

Tabela 22: Dados estatísticos referentes à melhor solução com a função TAF para a série de Lorenz.

Melhor <i>fitness</i> :	$3,840 * 10^{-6}$
Média do <i>fitness</i> em 30 exp.:	$5,976 * 10^{-6}$
Desvio Padrão:	$1,453 * 10^{-7}$
R ² Treinamento:	0,999
R ² Validação:	0,996
Desvio Padrão do Erro de Treinamento:	$1,967 * 10^{-3}$
Desvio Padrão do Erro de Validação:	$1,232 * 10^{-2}$
MSE Treinamento:	0,270
MSE Validação:	0,305

Pode-se ressaltar o fato de que o parâmetro R² para o conjunto de validação ficou próximo do conjunto de treinamento, indicando que o modelo foi capaz de responder aos dados de validação, da mesma forma como responde ao conjunto de dados de treinamento, que são dados já apresentados ao modelo na etapa de identificação.

A Figura 35 apresenta a previsão e o sinal de erro para o conjunto de treinamento.

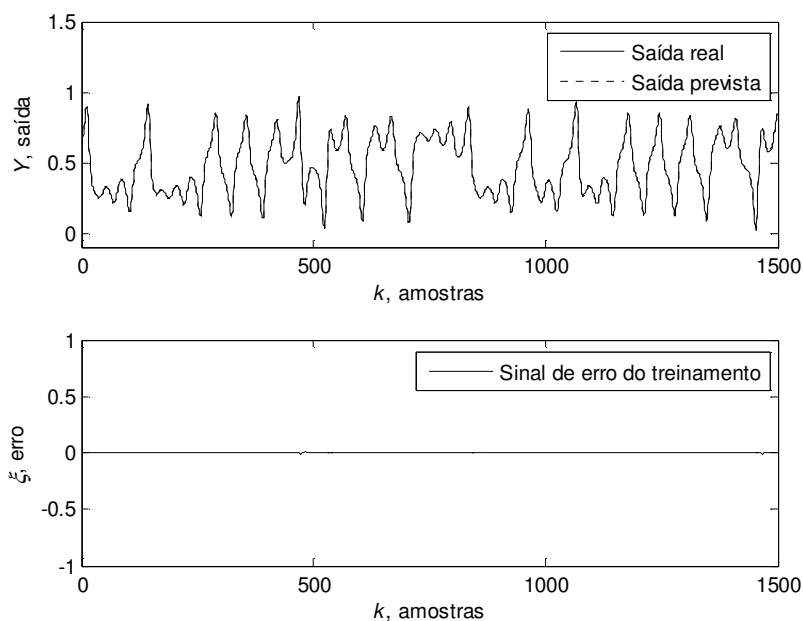


Figura 35: Previsão e sinal de erro utilizando a função TAF para o conjunto de dados de treinamento da série de Lorenz.

A Figura 36 apresenta a previsão e o sinal de erro para o conjunto de validação. Avaliando o sinal de erro, houve um pico de aproximadamente 21% de erro logo na 1ª amostra, após este evento, o maior pico de erro absoluto é de no máximo 7%, sendo que ambos estão fora da faixa aceitável de erro. No entanto, o erro absoluto médio da previsão é de 0,4%, dentro da faixa aceitável.

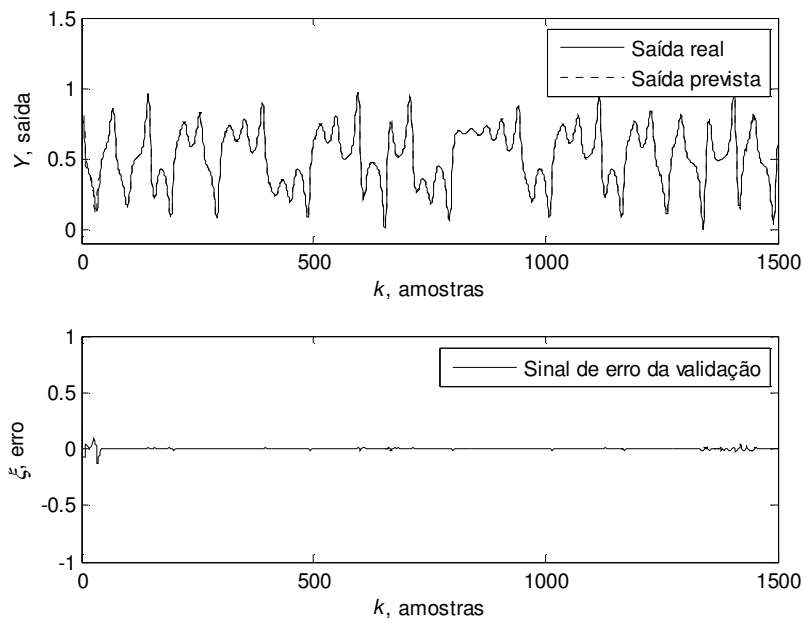


Figura 36: Previsão e sinal de erro utilizando a função TAF para o conjunto de dados de validação da série de Lorenz.

A Figura 37 mostra a previsão do tipo N -passos-a-frente para o conjunto de validação.

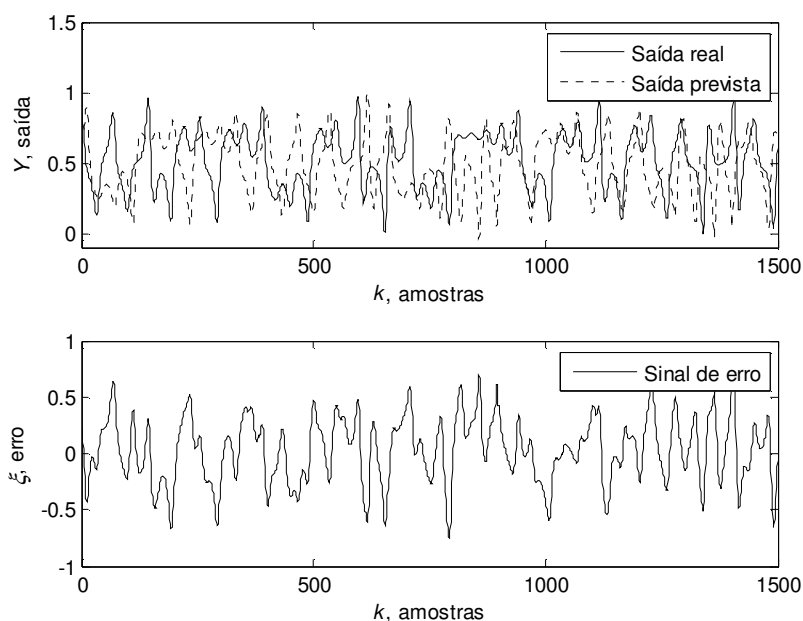


Figura 37: Previsão do tipo N -passos-a-frente e sinal de erro empregando a função TAF para o conjunto de validação da série de Lorenz.

Ao observar a Figura 37, nota-se que o modelo não foi capaz de prever a dinâmica caótica do sistema quando realimentado com dados previstos anteriormente, comportamento semelhante ao relatado para a série temporal caótica de Mackey-Glass.

A Tabela 23 traz alguns dados estatísticos referentes à esta previsão.

Tabela 23: Dados estatísticos referentes à previsão do tipo N -passos-a-frente.

R^2 :	$-1,06 \equiv 0$
MSE:	0,085
Desvio Padrão do Sinal de Erro:	0,291

Examinando a Tabela 23, o parâmetro R^2 negativo confirma a observação anterior de que o modelo não foi capaz de prever com um horizonte longo. O valor alto do desvio padrão também é mais um indício que o modelo se comportou de forma indesejada.

Ao analisar o resíduo ξ da previsão do tipo um-passo-a-frente com os testes de autocorrelação e correlação cruzada para validar o modelo, foi possível traçar os gráficos para as seis entradas originais da série. Os gráficos se encontram a partir da Figura 70 até a Figura 76 no Anexo B deste trabalho.

Ao examinar a Figura 70 nota-se que há uma correlação positiva acima da faixa de confiança até aproximadamente o atraso $\tau = 4$ para o teste de autocorrelação e $\tau = 9$ no teste autocorrelação de alta ordem.

Observando a Figura 71, pode-se constatar nenhuma correlação positiva nos teste de correlação cruzada.

A partir das afirmações anteriores, foram escolhidas as entradas $e(k - 1)$, $e(k - 5)$, $e(k - 8)$, $e(k - 1)e(k - 8)^3$ e $x(k - 1)e(k - 8)$, a serem adicionadas ao modelo. Com isso, o modelo anterior do tipo AR, passa a ser do tipo NARMA devido às novas entradas.

Os testes de correlação do resíduo do novo modelo podem ser observados no Anexo B, da Figura 77 até a Figura 88.

Ao comparar a Figura 70 com a Figura 77, nota-se uma melhora significativa no teste autocorrelação de alta ordem. A Figura 86 e a Figura 88 que apresentam os testes de correlação cruzada para as novas entradas $e(k - 5)$ e $x(k - 1)e(k - 8)$ resultaram em correlações positivas em parte dos testes de alta ordem.

5.3.2. ESTUDO DE CASO MODIFICADO SISTEMA DE LORENZ

O algoritmo GPSO foi escolhido para otimizar o novo modelo, empregando 50 neurônios ocultos, sendo 45 deles compostos pela função TAF e o restante neurônios lineares.

A Tabela 24 apresenta os dados estatísticos referentes à simulação com as novas entradas.

Tabela 24: Dados estatísticos referentes ao melhor experimento com as novas entradas para a série de Lorenz.

Melhor <i>fitness</i> :	$8,752 * 10^{-15}$
Média do <i>fitness</i> em 30 exp.:	$5,582 * 10^{-12}$
Desvio Padrão:	$2,900 * 10^{-11}$
R ² Treinamento:	1
R ² Validação:	1
Desvio Padrão do Erro de Treinamento:	$2,924 * 10^{-10}$
Desvio Padrão do Erro de Validação:	$9,436 * 10^{-10}$
MSE Treinamento:	0,270
MSE Validação:	0,305

Ao comparar os valores de melhor *fitness* da Tabela 22 com os valores da Tabela 24, pode-se salientar a queda do valor de *fitness* e os valores do parâmetro R² que ficaram iguais a “1”, demonstrando que o modelo foi capaz de prever com exatidão a dinâmica do modelo em questão. Ambos indicam uma superioridade do modelo novo em relação ao anterior.

A Figura 38 apresenta a previsão e o sinal de erro para o conjunto de treinamento.

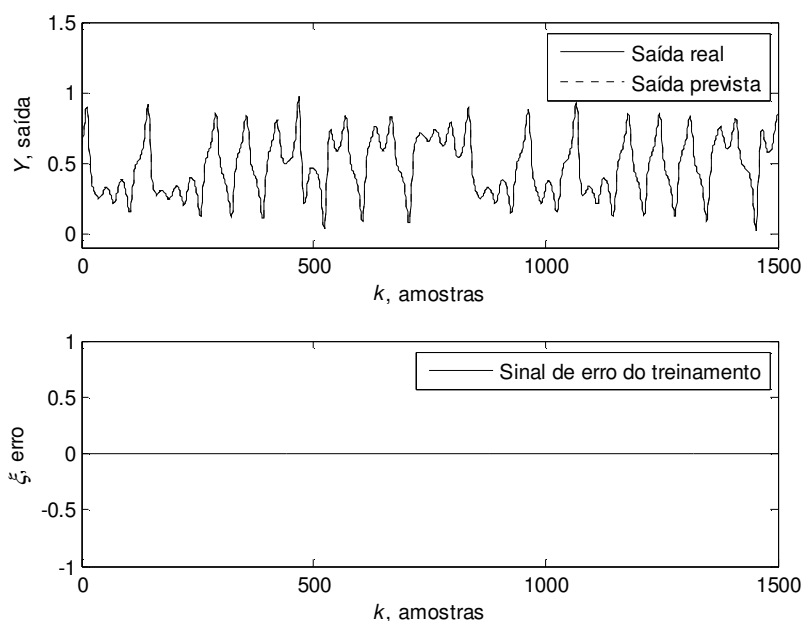


Figura 38: Previsão e sinal de erro utilizando a função TAF para o conjunto de dados de treinamento da série de Lorenz.

A Figura 39 traz a previsão e o sinal de erro para o conjunto de validação.

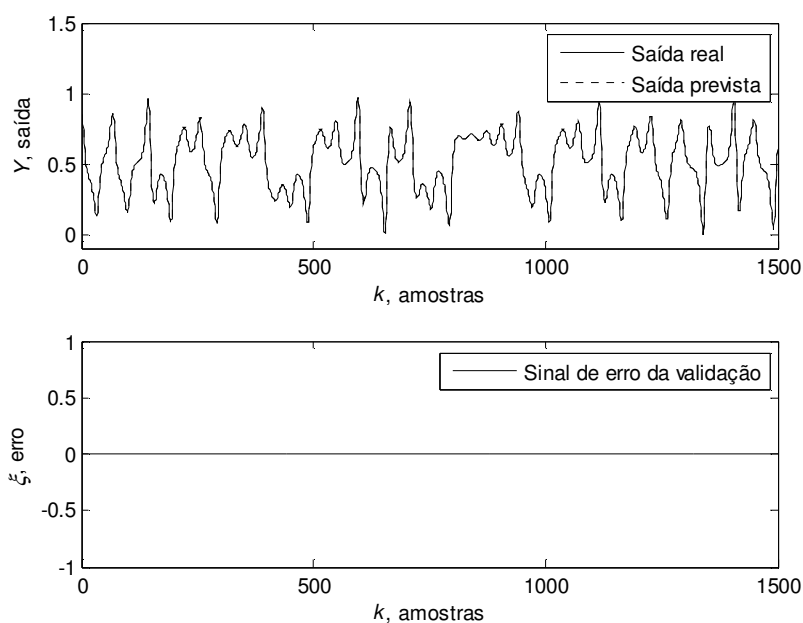


Figura 39: Previsão e sinal de erro utilizando a função TAF para o conjunto de dados de validação da série de Lorenz.

Ao examinar a Figura 39 observa-se que o modelo conseguiu identificar todas as dinâmicas caóticas do sistema. O maior erro registrado foi $2,459 \cdot 10^{-8}$ na 1ª

amostra. O erro absoluto médio da previsão foi $3,846 * 10^{-10}$. Ambos os valores estão dentro da margem de erro considerada aceitável.

A Figura 40 apresenta a previsão do tipo N -passos-a-frente empregando o modelo com as novas entradas para o conjunto de validação.

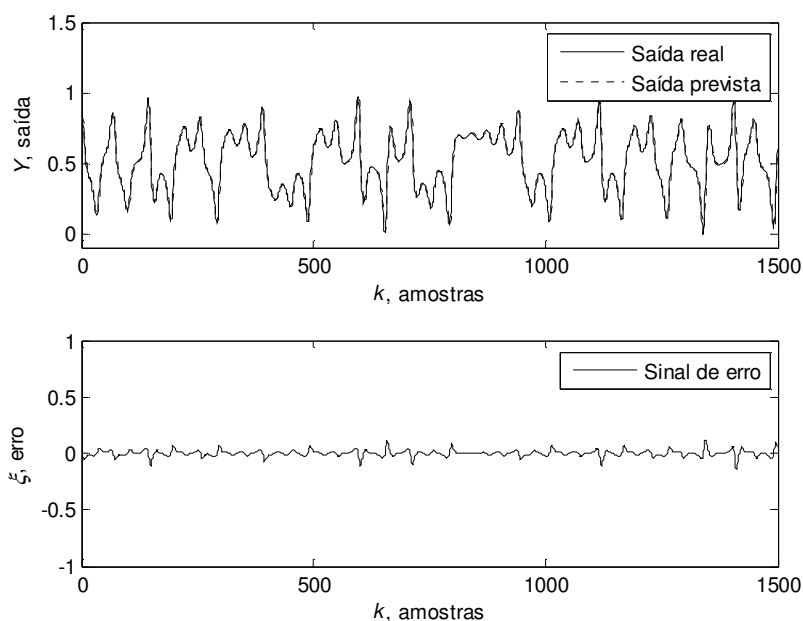


Figura 40: Previsão do tipo N -passos-a-frente e sinal de erro utilizando a função TAF para o conjunto de dados de validação para a série de Lorenz.

Analisando a Figura 40, o modelo foi capaz de prever a dinâmica caótica do sistema mesmo sendo realimentado com seus dados previstos anteriormente. O erro absoluto máximo observado foi 84% na 1ª amostra, no entanto, o erro absoluto médio foi 1,99%, dentro da margem considerada aceitável.

A Tabela 25 mostra os dados estatísticos referentes à previsão N -passos-a-frente.

Tabela 25: Dados estatísticos referentes à previsão do tipo N -passos-a-frente para a série de Lorenz com as novas entradas.

R^2 :	0,970
MSE:	$1,240 * 10^{-3}$
Desvio Padrão do Sinal de Erro:	$3,522 * 10^{-3}$

Pode-se destacar na Tabela 25 o parâmetro R^2 , que teve uma alteração expressiva em relação à previsão N -passos-a-frente empregando o modelo com as entradas originais. A proximidade do parâmetro ao valor “1” é comprovado pelo baixo valor do erro absoluto médio. O parâmetro MSE para este modelo foi inferior em relação ao modelo anterior.

6. CONCLUSÕES E SUGESTÕES PARA FUTUROS TRABALHOS

Após os apresentar os resultados, conclui-se que para o estudo de caso baseado na série de Box & Jenkins, os algoritmos propostos foram capazes de otimizar os parâmetros e melhorar o desempenho da identificação para ambas as funções de ativação. Apesar de o erro médio absoluto estar dentro da faixa aceitável de 5%, estudos posteriores podem procurar atenuar o pico de erro absoluto, que se manteve acima da faixa aceitável.

O emprego dos testes de autocorrelação e correlação cruzada para validação do modelo foi bem sucedido. A escolha de novas entradas de forma estratégica e coerente melhorou o desempenho do modelo nas duas formas de previsão, um-passo-a-frente e N -passos-a-frente. Fato este constatado por testes, como por exemplo, o R^2 e a verificação dos valores de erro absoluto médio e máximo.

Para o estudo de caso de Mackey-Glass, pode-se concluir que os algoritmos de otimização foram aptos para gerar soluções, possibilitando a identificação e previsão da série caótica pela rede neural, com ambas as funções de ativação. O valor do erro absoluto médio superou a faixa aceitável na previsão um-passo-a-frente. Na previsão N -passos-a-frente, o modelo não foi capaz de prever o comportamento da série.

A análise dos testes de autocorrelação e correlação cruzada colaboraram na escolha de novas entradas para a identificação do modelo, sendo que o desempenho do novo modelo superou o anterior conforme era esperado.

O novo modelo foi capaz fazer a previsão do tipo um-passo-a-frente com um erro absoluto médio dentro da faixa aceitável. Apesar da previsão do tipo N -passos-a-frente ter superado esta faixa, o modelo NARMA foi capaz de prever com um

desempenho razoável, mas significativamente melhor que o modelo anterior, a série caótica.

A série caótica de Lorenz apresentou os melhores resultados entre os três estudos de caso. O algoritmo de otimização obteve soluções de qualidade para estimação do modelo. A previsão do tipo um-passo-a-frente alcançou o desempenho esperado, com o erro absoluto médio dentro da faixa aceitável. No entanto, o modelo não foi capaz de realizar a previsão do tipo N -passos-a-frente.

A aplicação dos testes de autocorrelação e correlação cruzada possibilitaram a escolha de novas entradas, alterando o modelo original para um modelo NARMA. As novas entradas melhoraram expressivamente o desempenho do modelo, alcançando o valor 1 no parâmetro R^2 . Esta melhora pode ser notada principalmente na previsão N -passos-a-frente, na qual foi o único modelo que obteve o erro absoluto médio dentro da faixa aceitável.

Os testes de autocorrelação e correlação cruzada para validação dos modelos geraram resultados positivos em todos os estudos de caso, sendo que os testes não-lineares foram capazes de encontrar não-linearidades em todos os modelos, melhorando o desempenho deles substancialmente após a escolha das entradas adicionais.

Sobre as funções de ativação escolhidas neste trabalho, pode-se concluir que a função de ativação sigmóide teve um desempenho inferior em relação ao poder de adaptação da função de ativação variável. Porém, a complexidade em sintonizar os parâmetros da função TAF faz com que seja quase uma obrigação a presença de algum algoritmo de otimização para tal tarefa. Quando isto não é possível, a função SIG se torna a opção mais atrativa.

O algoritmo ELM se mostrou uma ferramenta poderosa e eficiente para a aprendizagem de redes neurais. Com um tempo de treinamento inferior a 0,1s, a sua aplicação em conjunto com algoritmos de otimização, para a sintonia das ponderações, supera tanto em tempo, quanto em qualidade, o seu concorrente mais popular, o método de retropropagação.

Sobre o algoritmo de otimização, conclui-se que o algoritmo de enxame de partículas foi capaz de otimizar os parâmetros da função de ativação variável, além das ponderações do algoritmo ELM com sucesso, com todas variações do algoritmo

PSO original convergindo para valores próximos da melhor solução em todos os experimentos.

A modificação proposta pelo autor, empregando a função gama incompleta para a geração do peso de inércia obteve bons resultados, superando as duas outras versões nos dois estudos de caso com séries caóticas. No estudo de caso da série de Lorenz, o GPSO obteve também o maior número de vitórias nas simulações realizadas.

Este trabalho abre as portas para pesquisas na aplicação da função gama incompleta em algoritmos de otimização. Mais especificamente, na área de inteligência de enxames, onde a aplicação da função é ainda pouco explorada, tanto no cenário nacional, quanto internacional.

Também ressalta o potencial do algoritmo ELM em aplicações conjuntas com algoritmos de otimização, tanto mono-objetivos, quanto multiobjetivos, sendo possível a sua aplicação em sistemas reais, graças ao seu tempo de processamento baixo.

REFERÊNCIAS

- [1] HAYKIN, S. *Neural Networks and Learning Machines*. 3rd Ed. **Prentice Hall**. USA, 2008.
- [2] VILLANUEVA, W. J. P. **Síntese Automática de Redes Neurais Artificiais com Conexões à Frente Arbitrárias**. Tese (Doutorado em Engenharia Elétrica), Universidade Estadual de Campinas, Campinas, 2011.
- [3] VILLANUEVA, W. J. P. **Comitê de Máquinas em Predição de Séries Temporais**. Dissertação (Mestrado em Engenharia Elétrica), Universidade Estadual de Campinas, Campinas, 2006.
- [4] MORETTIN, P. A.; TOLOI, C. M. *Análise de Séries Temporais*. 2^a Ed. **Edgard Blücher**, São Paulo, 2006.
- [5] NEWBOLD, P. *Statistics for Business & Economics*. 4th ed. **Prentice Hall**, 1995.
- [6] FIGUEREDO, C. J. **Previsão de Séries Temporais Utilizando a Metodologia Box & Jenkins e Redes Neurais para Inicialização de Planejamento e Controle de Produção**. Dissertação (Mestrado em Métodos Numéricos), Universidade Federal do Paraná, Curitiba, 2008.
- [7] EHLERS, R. S. *Análise de Séries Temporais*. Departamento de Estatística, UFPR, 2005. Disponível em <<http://leg.est.ufpr.br/~ehlers/notas>>. Acesso em: 05/11/2012.
- [8] ROCHA, V. B. **Uma Abordagem de Wavelets Aplicada à Combinação de Previsões: Uma Análise Teórica e Experimental**. Dissertação (Mestrado em Métodos Numéricos), Universidade Federal do Paraná, Curitiba, 2008.
- [9] G. E. P. BOX; G. M. JENKINS. *Time Series Analysis: Forecasting and Control*, 2nd ed. San Francisco, CA: **Holden-Day**, 1976.

- [10] Banco de dados da Universidade de York. Disponível em: <<http://www.york.ac.uk/depts/math/data/ts/>>. Acesso em: 10/05/2012.
- [11] TZAFESTAS, S. G.; ZIKIDIS, K. C. NeuroFAST: On-Line Neuro-Fuzzy ART-Based Structure and Parameter Learning TSK Model. **IEEE Transactions on Systems, Man and Cybernetics**. v. 31, n. 5, p. 797-802, 2001.
- [12] ABE, S. Fuzzy Function Approximators with Ellipsoidal Regions. **IEEE Transactions on Systems, Man and Cybernetics**. v. 29, n. 4, p. 654-661, 1999.
- [13] LI, D.; HAN, M.; WANG, J. Chaotic Time Series Prediction Based on a Novel Robust Echo State Network. **IEEE Transactions on Neural Networks and Learning Systems**. v. 23, n. 5, p. 787-799, 2012.
- [14] ALLIGOOD, K. T.; SAUER, T. D.; YORKE, J. A. Chaos: An Introduction To Dynamical Systems. 1st Ed. **Springer**, US. 1996.
- [15] KINSNER, W. Characterizing Chaos Through Lyapunov Metrics. **IEEE Transactions on Systems, Man and Cybernetics**, v. 36, n. 2, p. 141-151, 2006.
- [16] PAN, F.; ZHANG, H.; XIA, M. A Hybrid Time-series Forecasting Model Using Extreme Learning Machines. **In: Proceedings of the Second International Conference on Intelligent Computation Technology and Automation**. v. 1, p. 933-936, 2009.
- [17] AGUIRRE, L. A. Introdução à Identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais. 3ª Ed. **Editora UFMG**. Belo Horizonte, Brasil, 2007.
- [18] CORRÊA, M. V. **Identificação caixa-cinza de sistemas não-lineares utilizando representações NARMAX racionais e polinomiais**. Tese (Doutorado

em Engenharia Elétrica), Universidade Federal de Minas Gerais, Belo Horizonte, 2001.

[19] SOUZA, A. V. P. **Modelagem e Análise Não-Lineares de Sinais Respiratórios e Cardiovasculares**. Dissertação (Mestrado em Engenharia Elétrica), Universidade Federal de Minas Gerais, Belo Horizonte, 2001.

[20] BITENCOURT, H. J. **Métodos de Estimação Recursiva Baseados no Filtro de Kalman Aplicados a Sistemas Não-Lineares**. Dissertação (Mestrado em Engenharia Elétrica), Universidade Federal de Minas Gerais, Belo Horizonte, 2003.

[21] HENSON, M. A.; SEBORG, D. E. Nonlinear Process Control. 1st Edition. **Prentice Hall**. 1997.

[22] GUYTON, A. C.; HALL, J. E. Textbook of Medical Physiology. 11th Ed. **Elsevier Saunders**. USA, 2006.

[23] HU, Y. H.; HWANG, J-N. Handbook of Neural Network Signal Processing. **Crc Press**. USA, 2001.

[24] HUANG, G-B.; ZHU, Q-Y.; SIEW, C-K. Extreme Learning Machine: A new Learning Scheme of Feedforward Neural Networks. **In: Proceedings of International Joint Conference on Neural Networks**. p. 985-990. Budapest, Hungary, 2004.

[25] HUANG, G-B.; ZHU, Q-Y.; SIEW, C-K. Extreme Learning Machine: Theory and Applications. **Neurocomputing**. v. 70, p. 489-501, 2006.

[26] PENROSE, R. A Generalized Inverse for Matrices. **In: Proc. Cambridge Philos. Soc.** v. 51, p. 406-413, 1955.

- [27] RAO, C. R.; MITRA, S. K. Generalized Inverse of a Matrix and its Applications. **In: Proceedings of Sixth Berkeley Symposium on Math. Statist. And Prob.** University of California Press. v. 1, p. 601-620, 1972.
- [28] GOLUB, H.G.; LOAN, C. F. V. Matrix Computations. 3rd Ed. **The John Hopkins University Press.** Maryland, USA, 1996.
- [29] SERRE, D. Matrices: Theory and Applications. **Springer.** New York, USA, 2002.
- [30] HUANG, G-B.; CHEN, L.; SIEW, C-K. Universal Approximation Using Incremental Constructive Feedforward Networks with Random Hidden Nodes. **IEEE Transactions on Neural Networks.** v. 17, n. 4, p. 879-892, 2006.
- [31] LI, B.; LI, Y.; RONG, X. The Extreme Learning Machine Learning Algorithm with Tunable Activation Function. **Neural Computing and Applications.** p. 1-9, 2012.
- [32] WANG, G-T.; LI, P.; CAO, J-T. Variable Activation Function Extreme Learning Machine Based on Residual Prediction Compensation. **Soft Computing.** v. 16, n. 9, p. 1477-1484, 2012.
- [33] BILLINGS, S. A.; VOON, W. S. F. Correlation Based Model Validity Tests for Non-Linear Models. **International Journal of Control.** v. 44, n. 1, p. 235-244, 1986.
- [34] BILLINGS, S. A.; ZHU, Q. M. Nonlinear Model Validation Using Correlations Tests. **International Journal of Control.** v. 60, n. 6, p. 1107-1120, 1994.
- [35] BILLINGS, S. A.; VOON, W. S. F. A Prediction-error and Stepwise-regression Estimation Algorithm for Non-Linear Systems. **International Journal of Control.** v. 44, n. 3, p. 803-822, 1986.
- [36] ENGELBRECHT, A. P. *Fundamentals of Computational Swarm Intelligence.* 1st. ed. **John Wiley & Sons,** 2005.

- [37] MICHALEWICZ, Z. A Survey of Constraint Handling Techniques in Evolutionary Computation Methods. **In: Proceedings of the 4th Annual Conference on Evolutionary Programming**. MIT Press, Cambridge, MA, USA, p. 135-155, 1995.
- [38] MENDES, R. **Population Topologies and Their Influence in Particle Swarm Performance**. Tese (Doutorado em Engenharia de Computação), Universidade de Minho, Portugal, 2004.
- [39] ÁVILA, S. L. **Algoritmos Genéticos Aplicados na Otimização de Antenas Refletores**. Dissertação (Mestrado em Engenharia Elétrica), Universidade Federal de Santa Catarina, Florianópolis, 2002.
- [40] KENNEDY, J.; EBERHART, R. C. Particle Swarm Optimization. **In: Proceeding of the IEEE Conference on Neural Networks**. v. 4, p. 1942-1948. Perth, Australia, 1995.
- [41] GREBOGE, T.; et al. Geometric Prosthesis Modelling to Skull Repairing Using Artificial Intelligence Methods. **In: Proceedings of the 41st International Conference on Computers and Industrial Engineering (CIE'41)**. p. 1-7, Los Angeles, USA, 2011.
- [42] GREBOGI, R. B.; SILVEIRA, C. C.; COELHO, L. S. Otimização dos Ganhos de um Controlador PID Baseado em PSO Aplicado à Sincronização de Sistemas Caóticos. **In: Proceedings of 10th Brazilian Congresso n Computational Intelligence (CBIC'2011)**. p. 1-8, Fortaleza, Brasil, 2011.
- [43] SHI, Y.; EBERHART, R. C. Parameter selection in particle swarm optimization. **In: Proceedings of the 7th International Conference on Evolutionary Programming VII**. p. 591-600, London, UK, 1998.
- [44] EBERHART, R. C.; SHI, Y. Particle Swarm Optimization: developments, applications and resources. **In: Proceedings of the Evolutionary Computation**. v. 1, p. 81-86. Indianapolis, USA, 2001.

- [45] CLERC, M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. **In: Proceedings of the Evolutionary Computation.** v. 3, p. 1951-1957. Washington D.C., USA, 1999.
- [46] EBERHART, R. C.; SHI, Y. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. **In: Proceedings of the Evolutionary Computation.** v. 1, p. 84-88. Indianapolis, USA, 2000.
- [47] SHI, Y.; EBERHART, R. C. Empirical Study of Particle Swarm Optimization. **In: Proceedings of the Evolutionary Computation.** v. 3, p. 1945-1950. Washington D.C., USA, 1999.
- [48] BRATTON, D.; KENNEDY, J. Defining a Standard for Particle Swarm Optimization. **In: Proceedings of the Swarm Intelligence Symposium.** p. 120-127. Honolulu, USA, 2007.
- [49] OCA, M. A. M. et al. Frankenstein's PSO: A Composite Particle Swarm Optimization Algorithm. **IEEE Transactions on Evolutionary Computation,** v. 13, n. 5, p. 1120-1132, 2009.
- [50] MELLO, M. H. A.; ARRUDA, H. V.; ORTOLANI, A. A. Probabilidade de Ocorrência de Totais Pluviais Máximos Horários, em Campinas – São Paulo. **Revista do Instituto Geológico.** v. 15, n. 1/2, p. 59-67, 1994.
- [51] KRISHNAMOORTHY, K.; MATHEW, T.; MUKHERJEE, S. Normal-Based Methods for a Gamma Distribution: Prediction and Tolerance Intervals and Stress-Strength Reliability. **Technometrics.** v. 51, p. 69-78, 2008.
- [52] YUE, S.; OUARDA, T. B. M. J.; BOBÉE, B. A Review of Bivariate Gamma Distributions for Hydrological Application. **Journal of Hydrology.** v. 246, n. 1-4, p. 1-18, 2001.

- [53] ALMPANIDIS, G.; KOTROPOULOS, C. Phonemic Segmentation Using the Generalized Gamma Distribution and Small Sample Bayesian Information Criterion. **Speech Communication**. v. 50, n. 1, p. 38-55, 2008.
- [54] KARAPETYAN, N. V. Dividend Optimization under the Gamma Distribution of Claims. **Moscow University Mathematics Bulletin**. v. 64, n. 5, p. 219-221, 2009.
- [55] TEMBELY, M.; LECOT, C.; SOUCEMARIANADIN, A. Prediction and Evolution of Drop-size Distribution for a New Ultrasonic Atomizer. **Applied Thermal Engineering**. v. 31, n. 5, p. 656-667, 2011.
- [56] QI, F. Monotocity Results and Inequalities for the Gamma and Incomplete Gamma Functions. **Mathematics Inequalities Applications**. v. 5, n. 1, p. 61-67, 2002.
- [57] ALZER, H. On Some Inequalities for the Incomplete Gamma Function. **Mathematics of Computation**. v. 66, n. 218, p. 771-778, 1997.
- [58] TEMME, N. M. Asymptotic Inversion of Incomplete Gamma Functions. **Mathematics of Computation**. v. 58, n. 198, p. 755-764, 1992.
- [59] ZHU, Q-Y.; et al. Evolutionary Extreme Learning Machine. **Pattern Recognition**. v. 38, n. 10, p. 1759-1763, 2005.
- [60] WANG, Y.; CAO, F.; YUAN, Y. A Study on Effectiveness of Extreme Learning Machine. **Neurocomputing**. v. 74, n. 16, p. 2483-2490, 2011.
- [61] CAO, J.; et al. Voting Based Extreme Learning Machine. **Information Sciences**. v. 185, n. 1, p. 66-77, 2012.
- [62] YUAN, Y.; WANG, Y.; CAO, F. Optimization Approximation Solution for Regression Problem Based on Extreme Learning Machine. **Neurocomputing**. v. 74, n. 16, p. 2475-2482, 2011.

[63] SILVA, D. N. G.; PACIFICO, L. D. S.; LUDERMIR, T. B. An Evolutionary Extreme Learning Machine Based on Group Search Optimization. **In: Proceedings of the IEEE Congress on Evolutionary Computation.** p. 574-580, 2011.

[64] XU, Y.; SHU, Y. Evolutionary Extreme Learning Machine Based on Particle Swarm Optimization. **In: Proceedings of the International Symposium on Neural Networks.** {s. n.} , p. 644-652, 2006.

ANEXO A

Pseudocódigo 1: Algoritmo da Máquina de Aprendizagem Extrema (ELM)

Dado um conjunto de treinamento $\mathfrak{X} = \{(x_i, y_{R_i}) \mid x_i \in R^n, y_{R_i} \in R^m, i = 1, \dots, N\}$, função de ativação $g(u)$ e número de neurônios ocultos \tilde{N} , faça:

Passo 1: assinalar aleatoriamente as ponderações de entrada w_i e os *bias* b_i , $i = 1, \dots, \tilde{N}$;

Passo 2: calcular a matriz de saída da camada oculta H ;

Passo 3: calcular a ponderação de saída β através da inversa de Moore-Penrose, $\beta = H^\dagger Y_R$, onde $Y_R = [y_{R_1}, \dots, y_{R_N}]^T$.

Pseudocódigo 2: Otimização por Enxame de Partícula

$it = 0$

Iniciar aleatoriamente as posições das partículas com distribuição uniforme

Iniciar as velocidades de todas as partículas com velocidades nulas

Enquanto critério de parada não é alcançado

Para cada partícula i do enxame:

Calcular *fitness*: Fazer $f_i = fitness$ de $\varphi_i(it)$

Atualizar $lP_i(it)$: se f_i é melhor que o *fitness* de $lP_i(it)$

então $lP_i(it) = \varphi_i(it)$

Fim para

Atualizar a melhor posição do enxame na iteração it , $X_{melhor}(it)$

Se $X_{melhor}(it)$ for melhor que $gP(it)$, então $gP(it) = X_{melhor}(it)$

Para cada partícula i do enxame:

Atualizar a velocidade pela equação (4.4)

Atualizar a posição pela equação (4.3)

Fim para

$it = it + 1$

Fim enquanto

Pseudocódigo 3: Otimização por Enxame de Partícula com Peso de Inércia Dinâmico
Linear

$it = 0$

Iniciar aleatoriamente as posições das partículas com distribuição uniforme

Iniciar as velocidades de todas as partículas com velocidades nulas

Enquanto critério de parada não é alcançado

Para cada partícula i do enxame:

Calcular *fitness*: Fazer $f_i = fitness$ de $\varphi_i(it)$

Atualizar $lP_i(it)$: se f_i é melhor que o *fitness* de $lP_i(it)$

então $lP_i(it) = \varphi_i(it)$

Fim para

Atualizar a melhor posição do enxame na iteração it , $X_{melhor}(it)$

Se $X_{melhor}(it)$ for melhor que $gP(it)$, então $gP(it) = X_{melhor}(it)$

Atualizar o peso de inércia a partir da equação (4.7)

Para cada partícula i do enxame:

Atualizar a velocidade pela equação (4.4)

Atualizar a posição pela equação (4.3)

Fim para

$it = it + 1$

Fim enquanto

Pseudocódigo 4: Otimização por Enxame de Partícula com a Função Gama
Incompleta

$it = 0$

Iniciar aleatoriamente as posições das partículas com distribuição uniforme

Iniciar as velocidades de todas as partículas com velocidades nulas

Gerar os valores do peso de inércia de acordo com a equação (4.9)

Enquanto critério de parada não é alcançado

Para cada partícula i do enxame:

 Calcular *fitness*: Fazer $f_i = fitness$ de $\varphi_i(it)$

 Atualizar $lP_i(it)$: se f_i é melhor que o *fitness* de $lP_i(it)$

 então $lP_i(it) = \varphi_i(it)$

Fim para

Atualizar a melhor posição do enxame na iteração it , $X_{melhor}(it)$

Se $X_{melhor}(it)$ for melhor que $gP(it)$, então $gP(it) = X_{melhor}(it)$

Para cada partícula i do enxame:

 Atualizar a velocidade pela equação (4.4)

 Atualizar a posição pela equação (4.3)

Fim para

$it = it + 1$

Fim enquanto

ANEXO B

Abaixo se encontram os testes de autocorrelação e correlação cruzada entre diversos sinais citados no Capítulo 3.

Os testes de autocorrelação foram realizados com o resíduo e também com o resíduo elevado ao quadrado, caracterizando um teste de alta ordem não-linear.

Um dos sinais analisados e proposto em [34] é o resultado da multiplicação do sinal de saída com o resíduo, sendo este sinal analisado em relação ao resíduo elevado ao quadrado e em relação às entradas elevadas ao quadrado. Outro sinal também proposto por em [34] é o resultado das entradas multiplicadas pelo resíduo, sendo este sinal analisado em relação ao resíduo.

As outras análises de correlações cruzadas realizadas são entre: as entradas e o resíduo; entradas elevadas ao quadrado e o resíduo; entradas elevadas ao quadrado e o resíduo elevado ao quadrado.

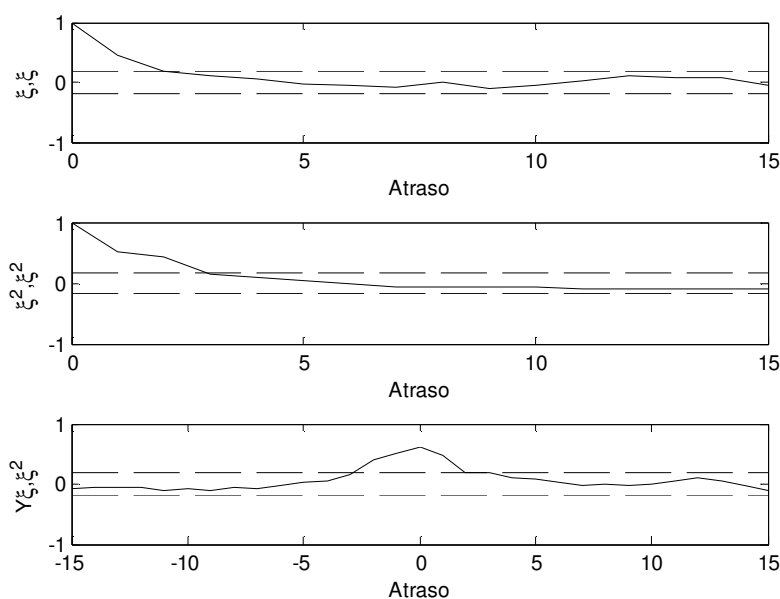


Figura 41: Gráficos de autocorrelação, autocorrelação de alta ordem e correlação cruzada de alta ordem entre a saída com o resíduo e o resíduo, para a série de Box & Jenkins com as entradas originais.

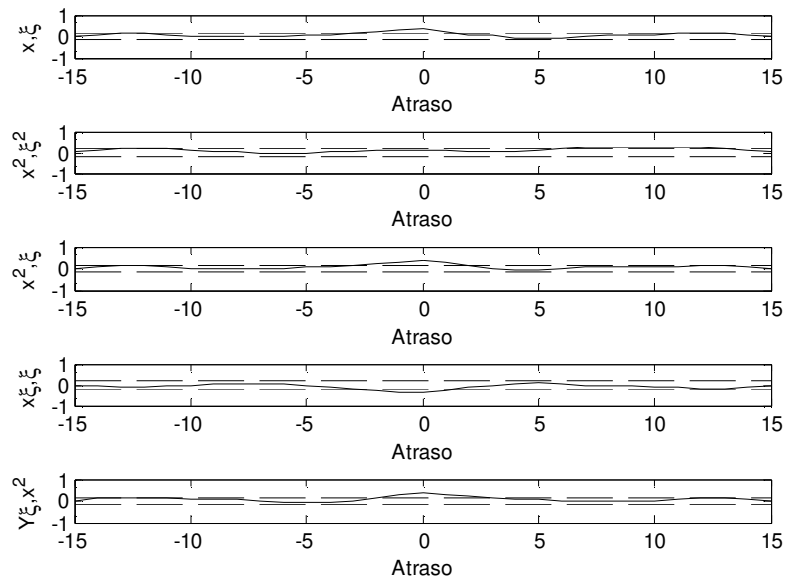


Figura 42: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 1)$ da série de Box & Jenkins com as entradas originais.

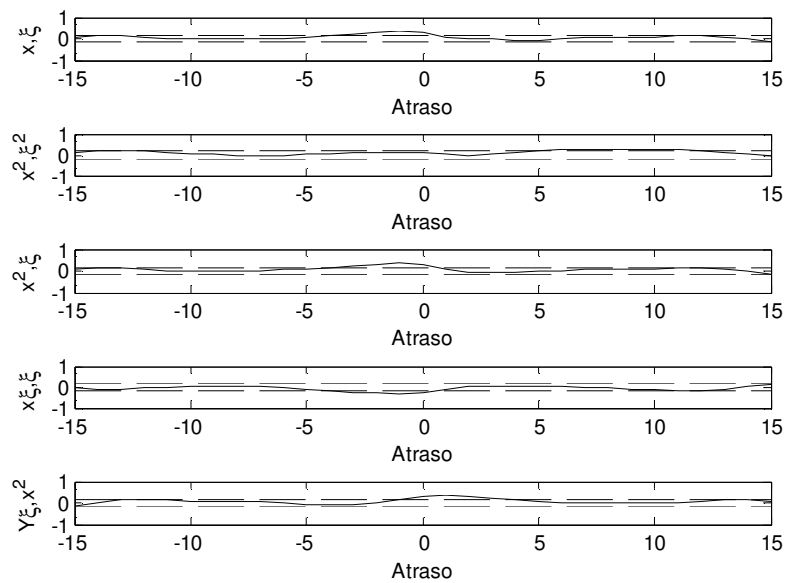


Figura 43: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 2)$ da série de Box & Jenkins com as entradas originais.

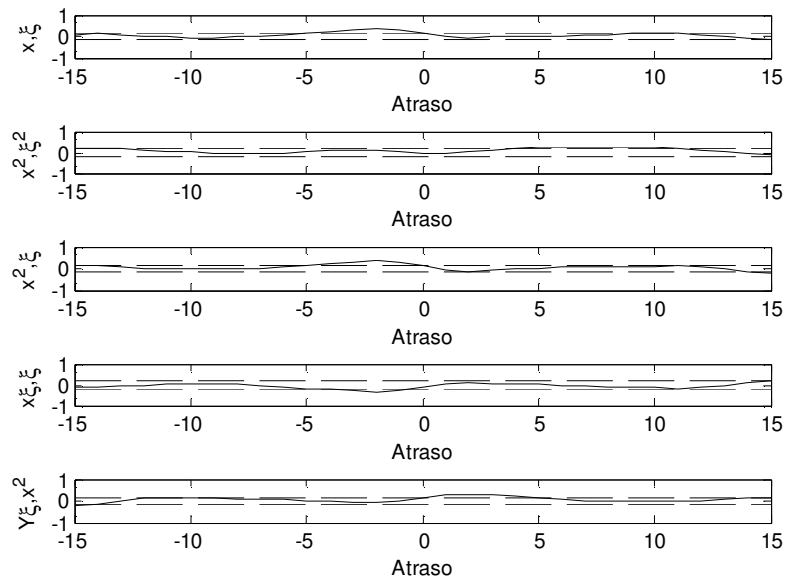


Figura 44: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 3)$ da série de Box & Jenkins com as entradas originais.

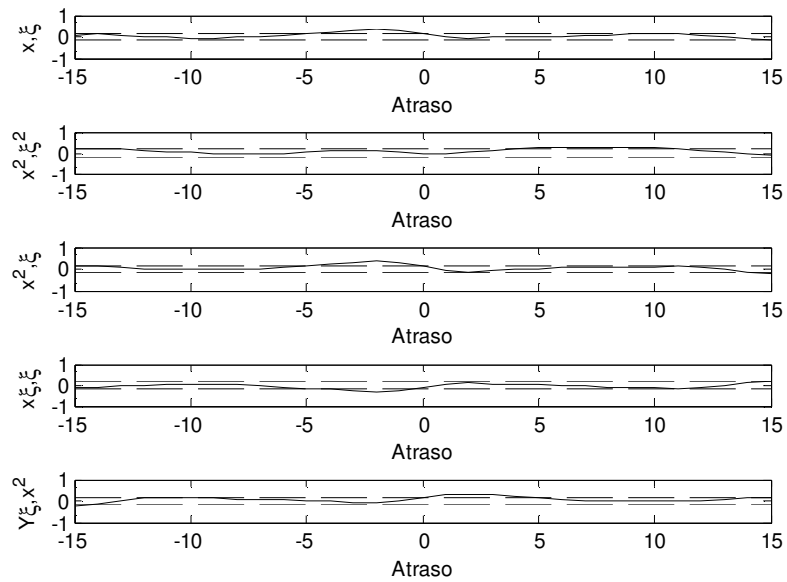


Figura 45: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 1)$ da série de Box & Jenkins com as entradas originais.

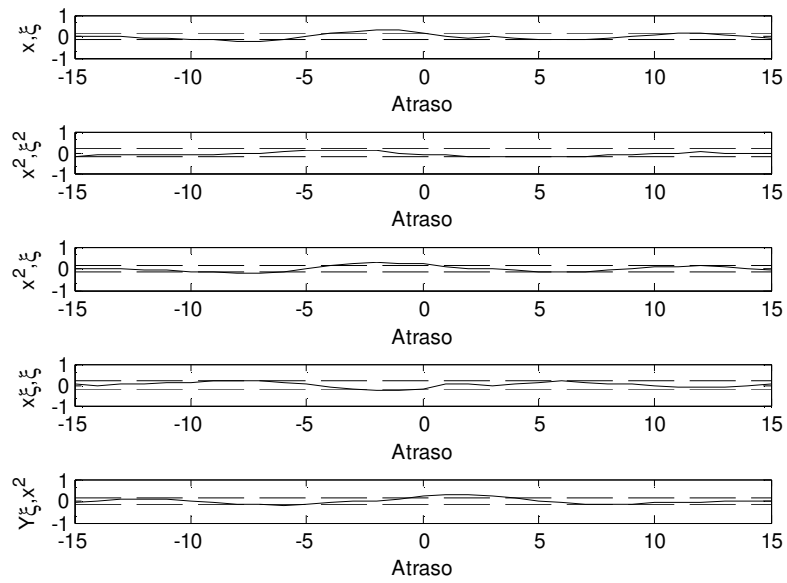


Figura 46: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 2)$ da série de Box & Jenkins com as entradas originais.

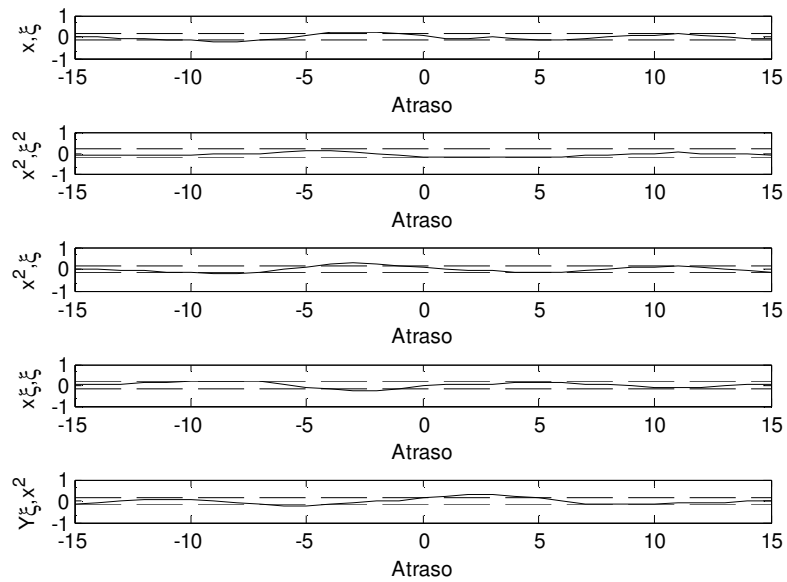


Figura 47: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 3)$ da série de Box & Jenkins com as entradas originais.

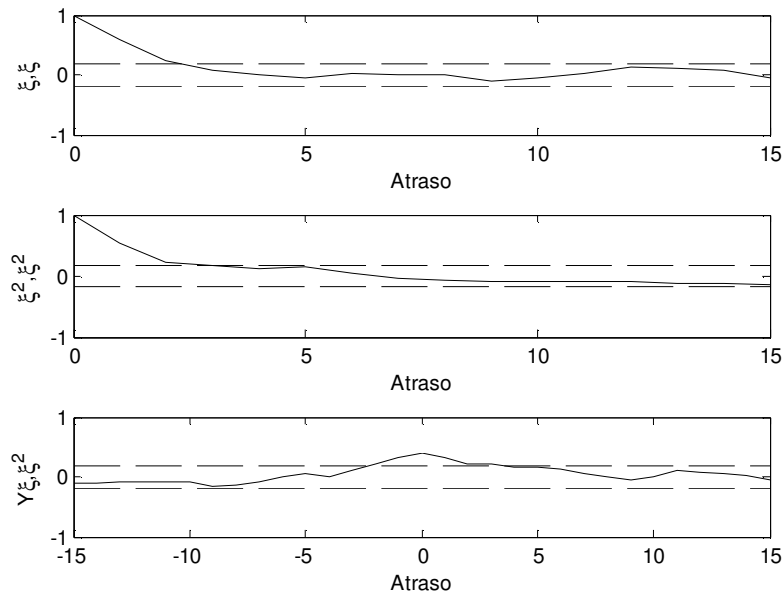


Figura 48: Gráficos de autocorrelação, autocorrelação de alta ordem e correlação cruzada de alta ordem entre a saída com o resíduo e o resíduo, para a série de Box & Jenkins com as novas entradas.

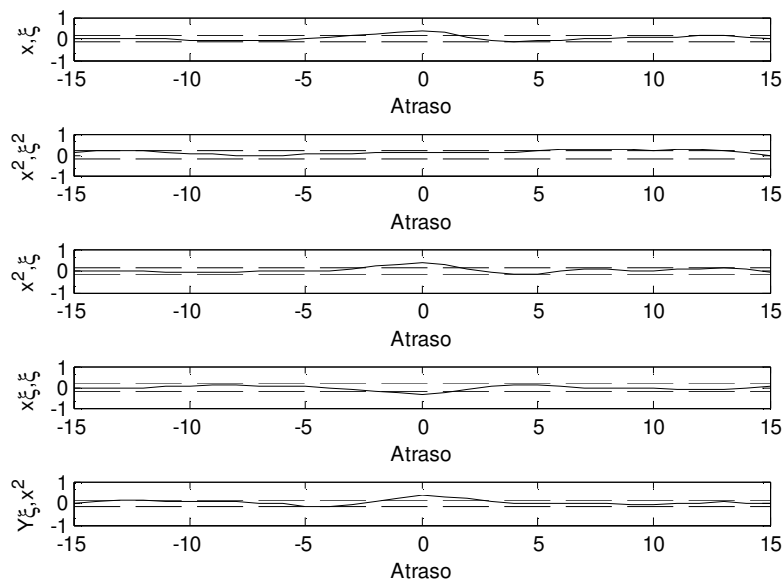


Figura 49: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 1)$ da série de Box & Jenkins com as novas entradas.

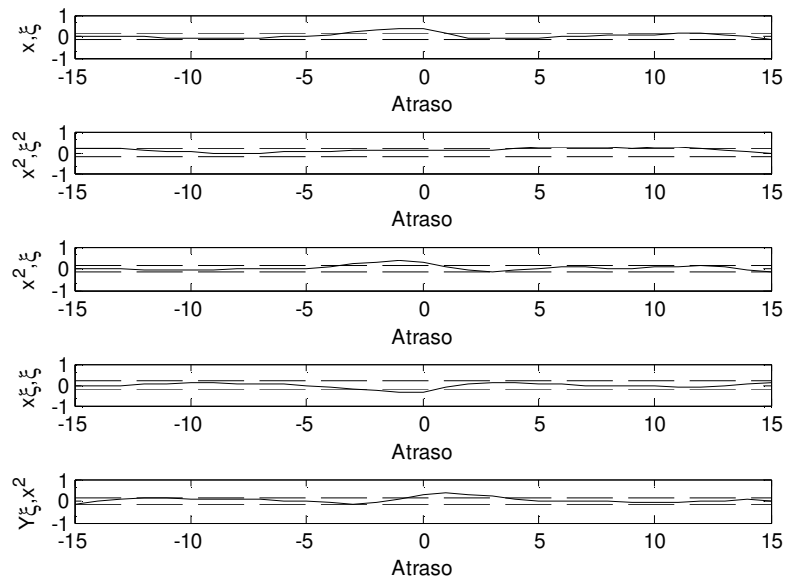


Figura 50: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 2)$ da série de Box & Jenkins com as novas entradas.

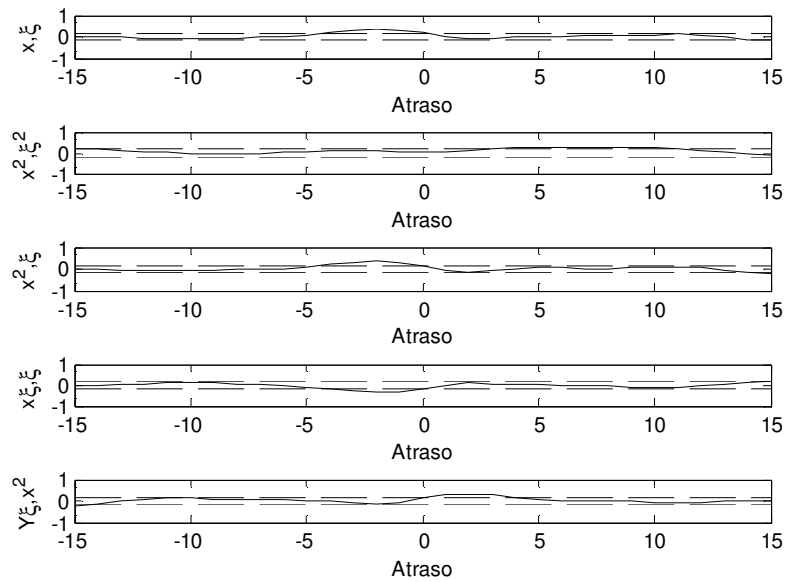


Figura 51: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 3)$ da série de Box & Jenkins com as novas entradas.

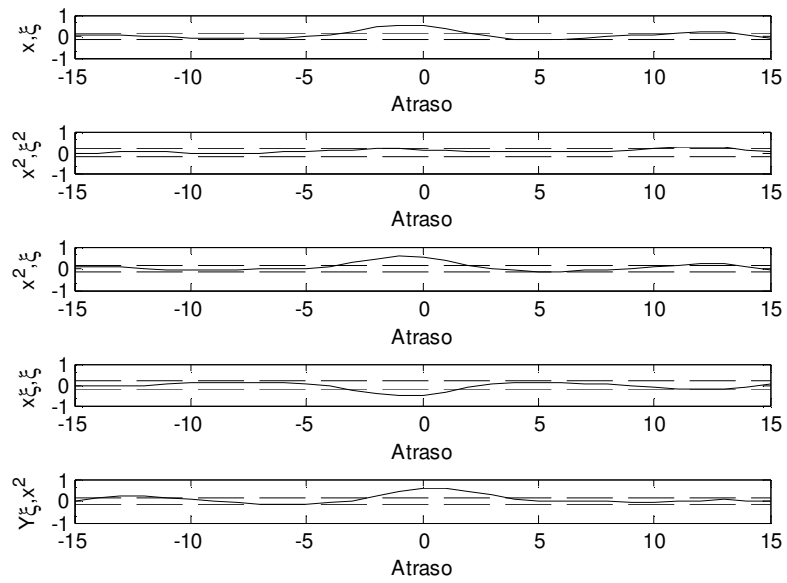


Figura 52: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 1)x(k - 1)$ da série de Box & Jenkins com as novas entradas.

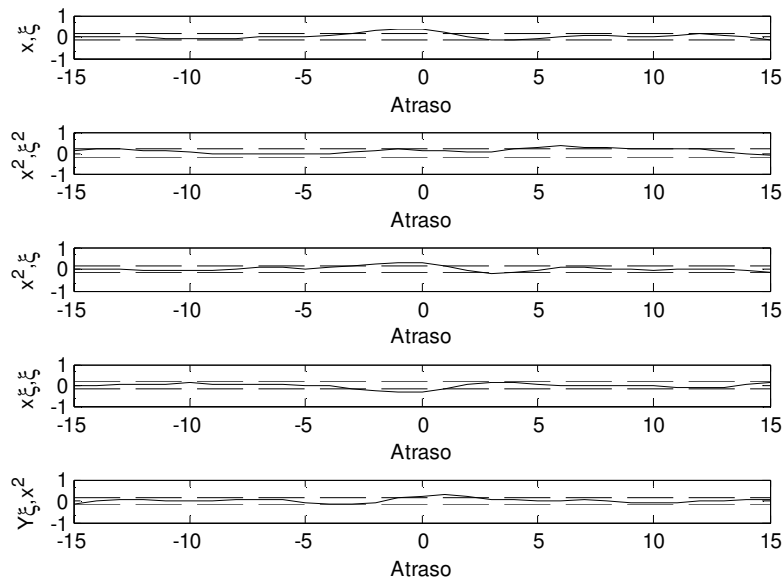


Figura 53: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 1)x(k - 2)$ da série de Box & Jenkins com as novas entradas.

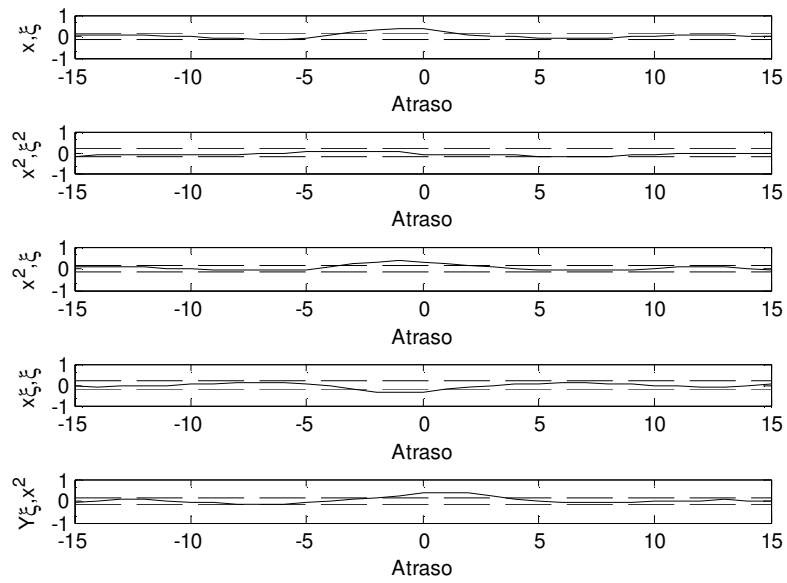


Figura 54: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 1)$ da série de Box & Jenkins com as novas entradas.

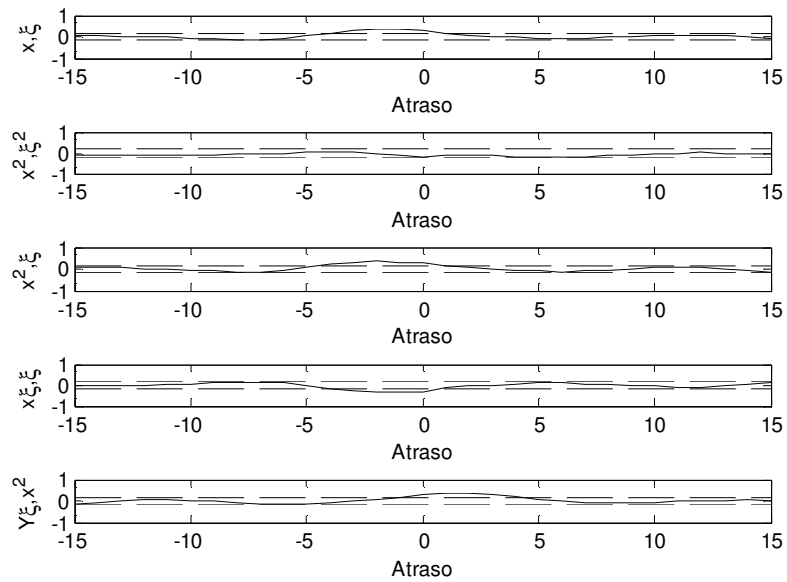


Figura 55: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 2)$ da série de Box & Jenkins com as novas entradas.

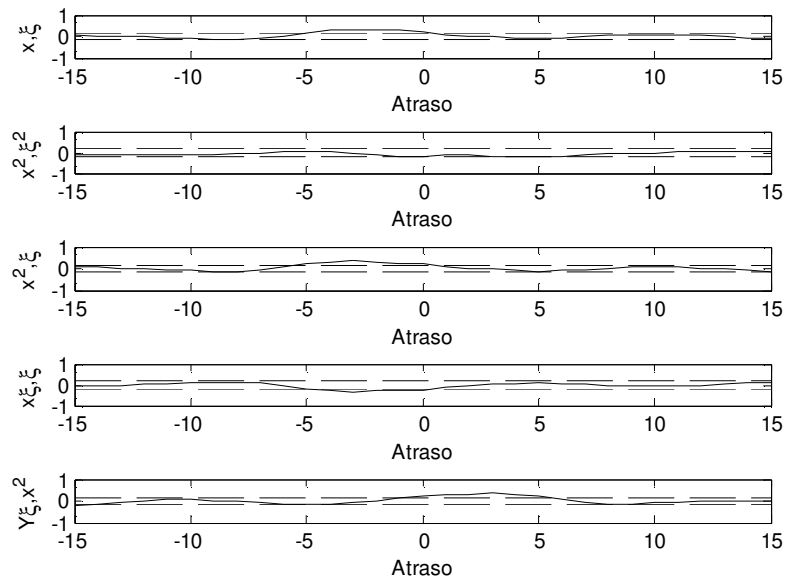


Figura 56: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $y(k - 3)$ da série de Box & Jenkins com as novas entradas.

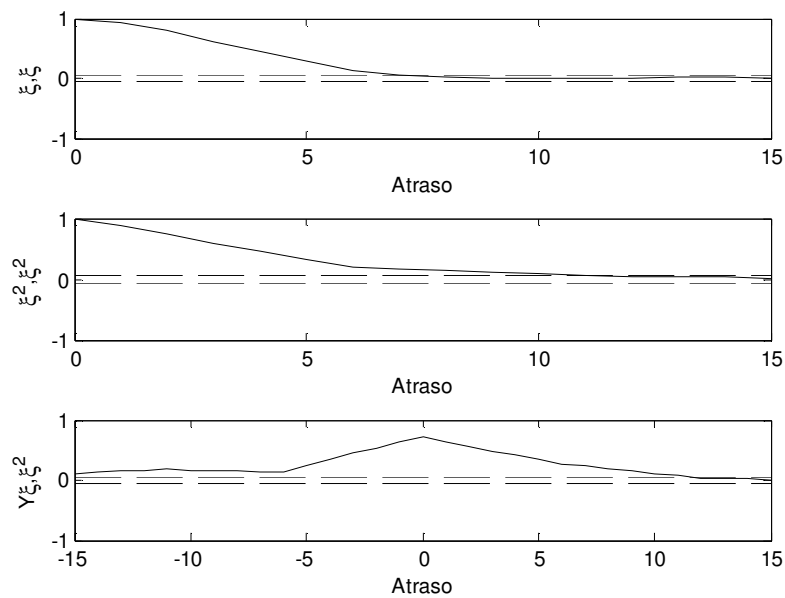


Figura 57: Gráficos de autocorrelação, autocorrelação de alta ordem e correlação cruzada de alta ordem entre a saída com o resíduo e o resíduo, para a série de Mackey-Glass com as entradas originais.

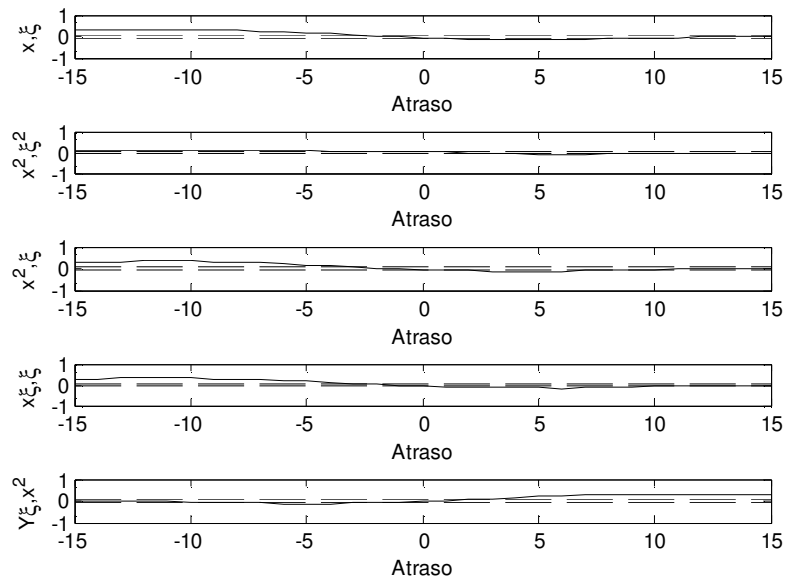


Figura 58: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k)$ da série de Mackey-Glass com as entradas originais.

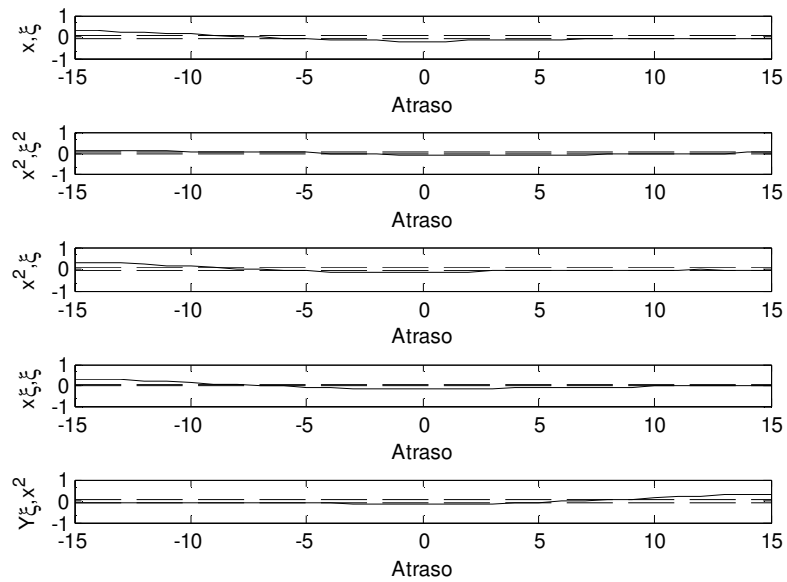


Figura 59: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 6)$ da série de Mackey-Glass com as entradas originais.

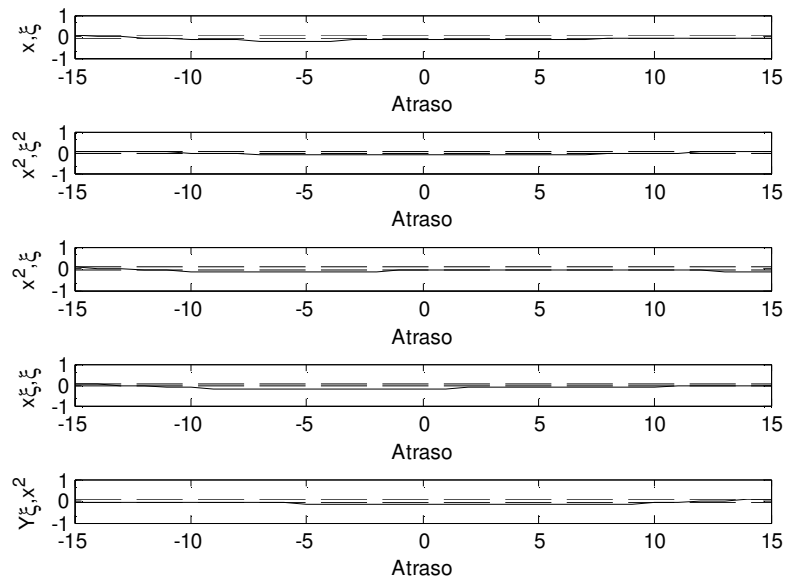


Figura 60: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 12)$ da série de Mackey-Glass com as entradas originais.

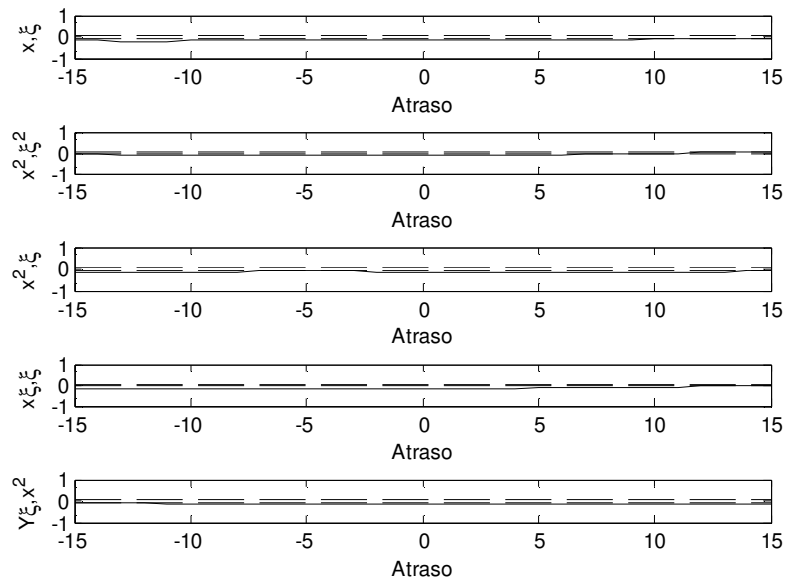


Figura 61: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 18)$ da série de Mackey-Glass com as entradas originais.

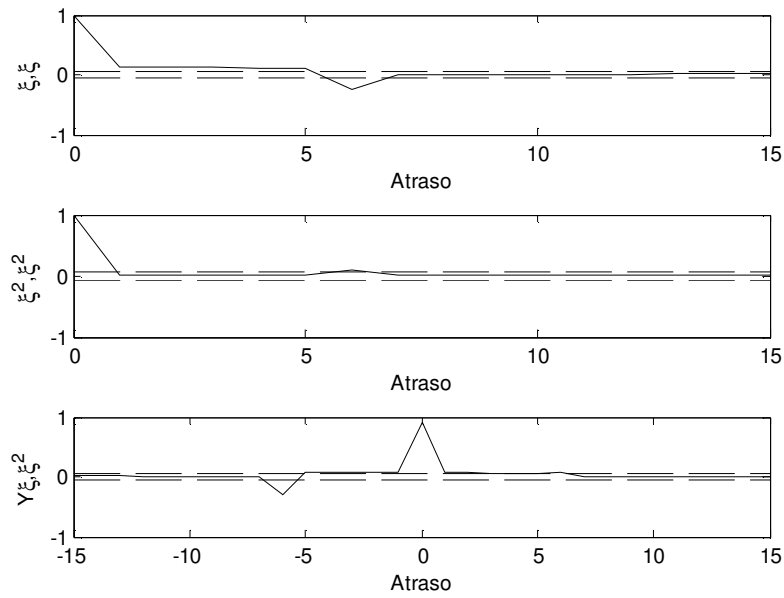


Figura 62: Gráficos de autocorrelação, autocorrelação de alta ordem e correlação cruzada de alta ordem entre a saída com o resíduo e o resíduo, para a série de Mackey-Glass com as novas entradas adicionadas.

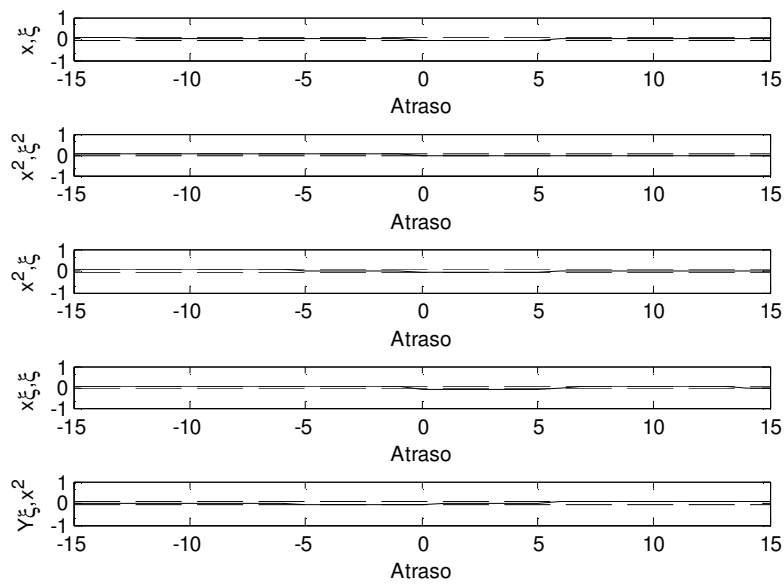


Figura 63: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k)$ da série de Mackey-Glass com as novas entradas adicionadas.

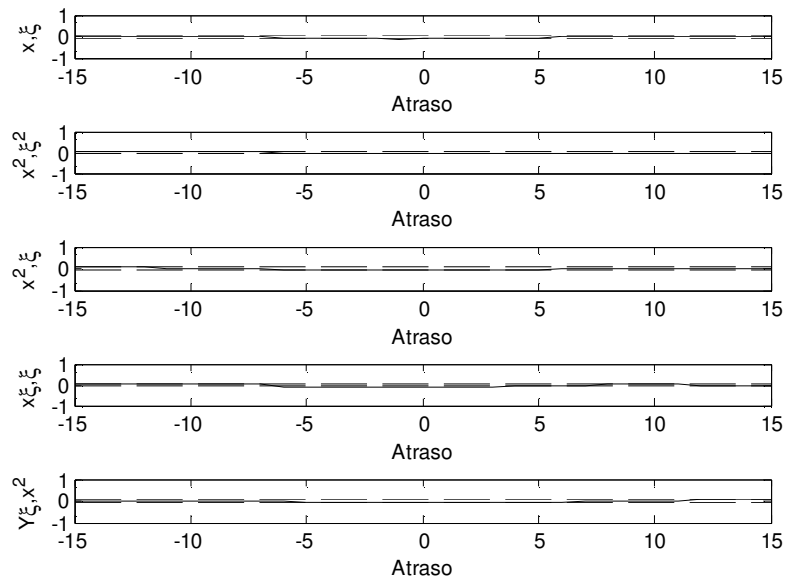


Figura 64: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 6)$ da série de Mackey-Glass com as novas entradas adicionadas.

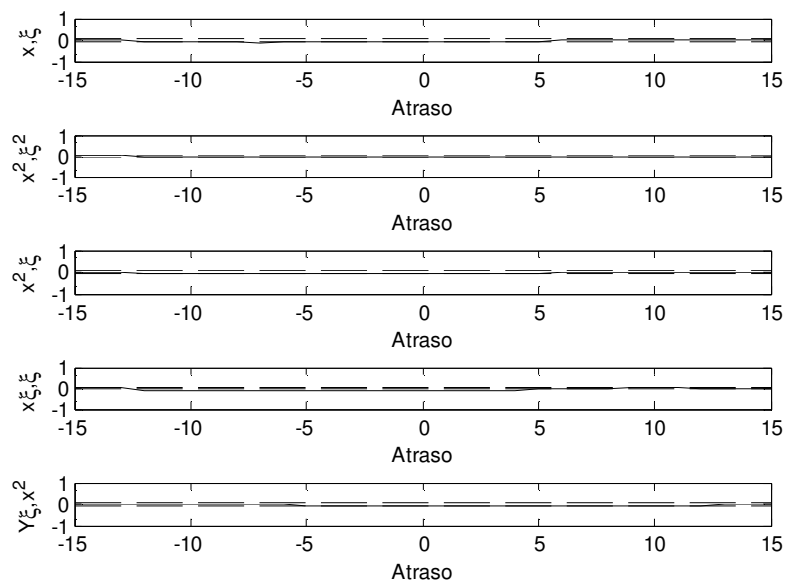


Figura 65: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 12)$ da série de Mackey-Glass com as novas entradas adicionadas.

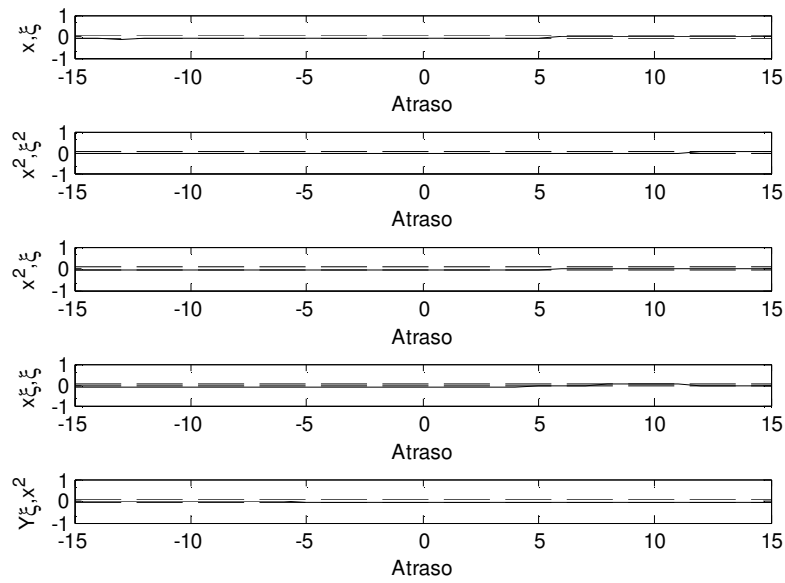


Figura 66: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 18)$ da série de Mackey-Glass com as novas entradas adicionadas.

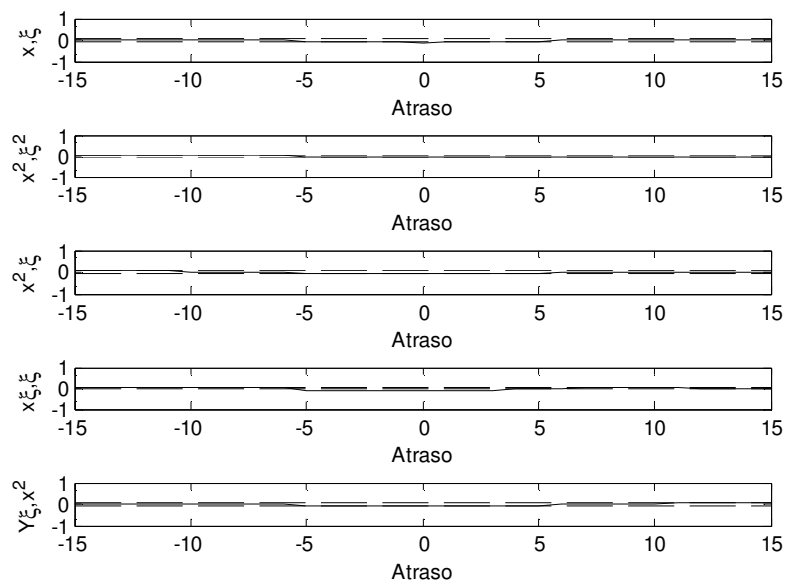


Figura 67: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 11)$ da série de Mackey-Glass com as novas entradas adicionadas.

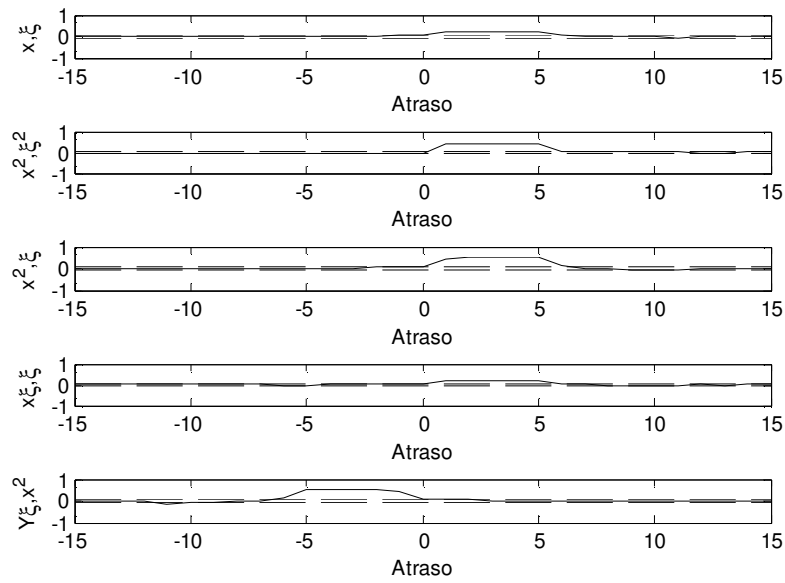


Figura 68: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $e(k - 5)$ da série de Mackey-Glass com as novas entradas adicionadas.

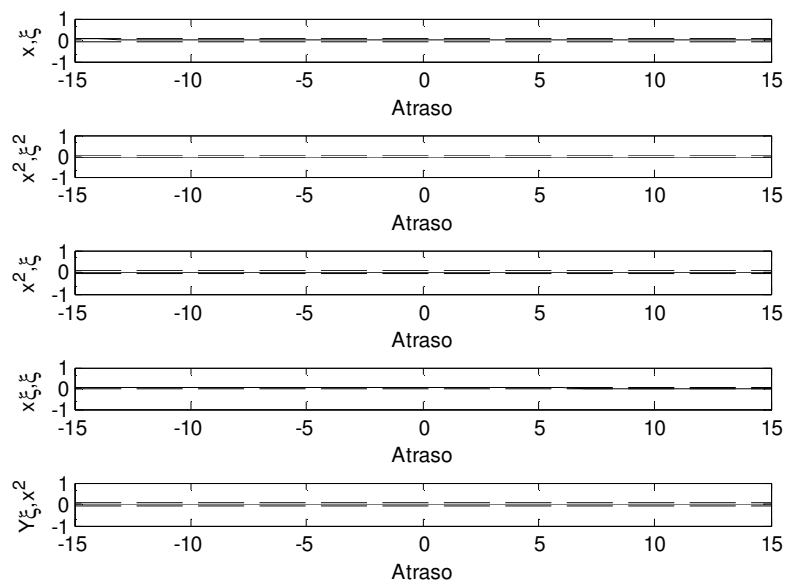


Figura 69: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $e(k - 6)x(k - 6)$ da série de Mackey-Glass com as novas entradas adicionadas.

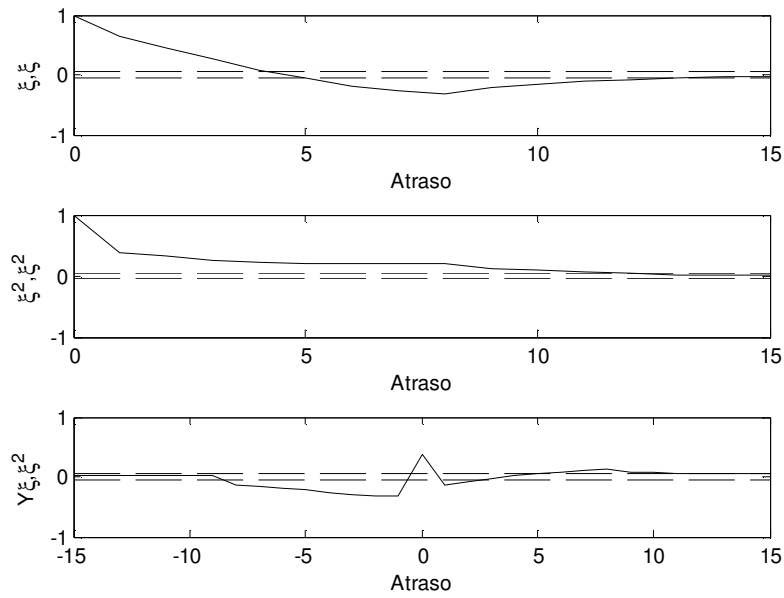


Figura 70: Gráficos de autocorrelação, autocorrelação de alta ordem e correlação cruzada de alta ordem entre a saída com o resíduo e o resíduo, para a série de Lorenz com as entradas originais.

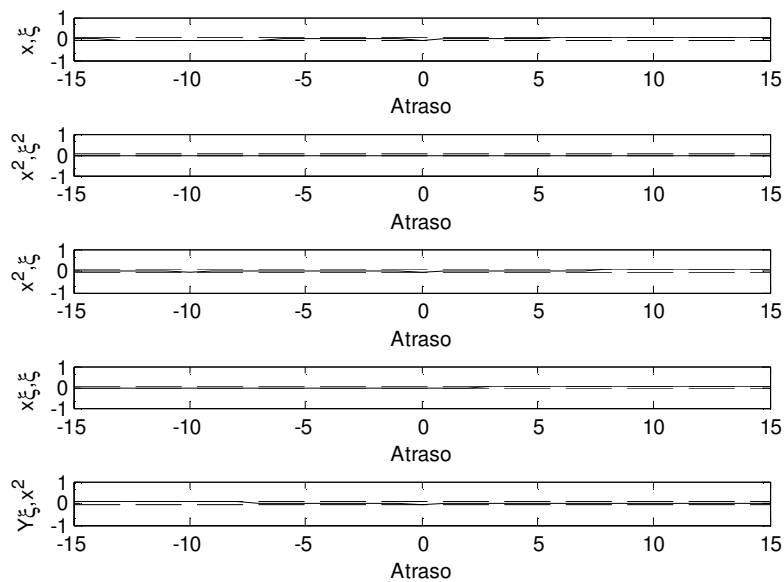


Figura 71: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k)$ da série de Lorenz com as entradas originais.

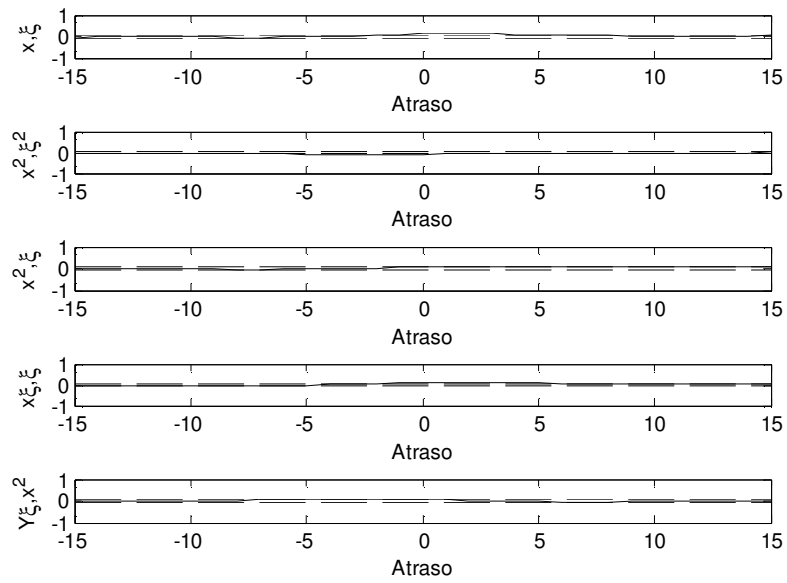


Figura 72: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 8)$ da série de Lorenz com as entradas originais.

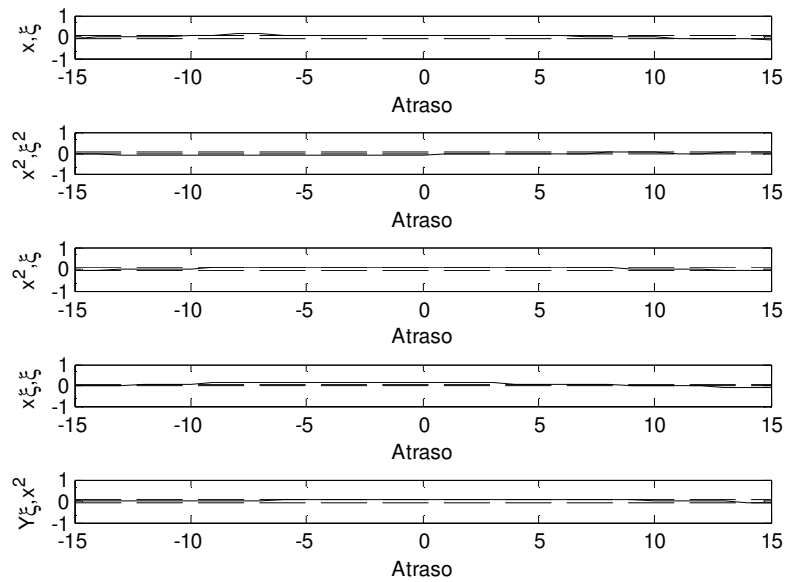


Figura 73: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 16)$ da série de Lorenz com as entradas originais.

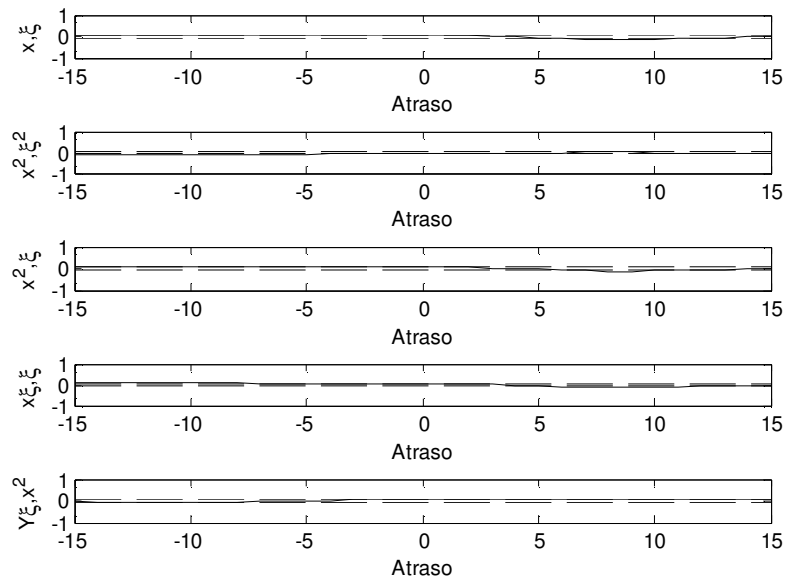


Figura 74: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 24)$ da série de Lorenz com as entradas originais.

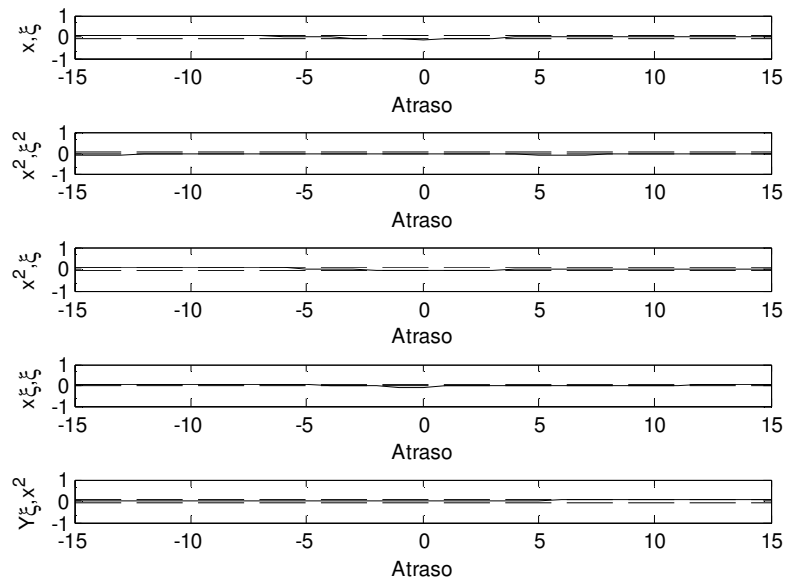


Figura 75: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 32)$ da série de Lorenz com as entradas originais.

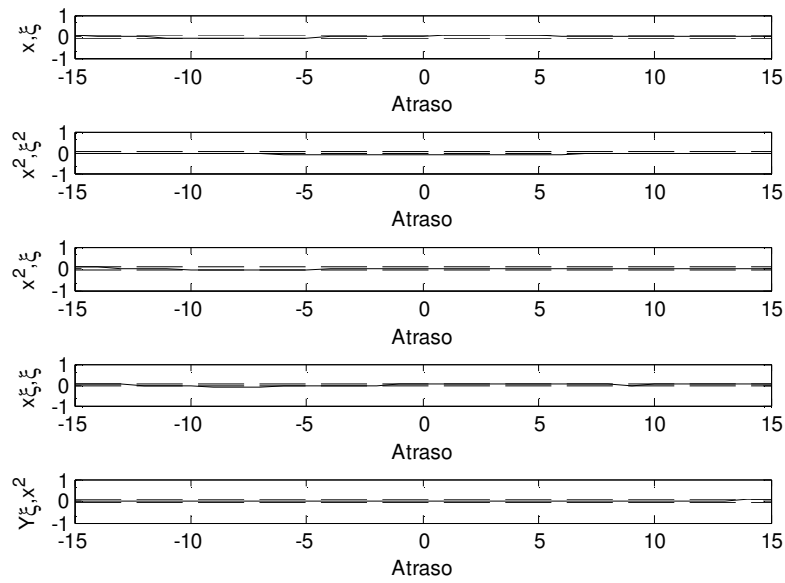


Figura 76: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 40)$ da série de Lorenz com as entradas originais.

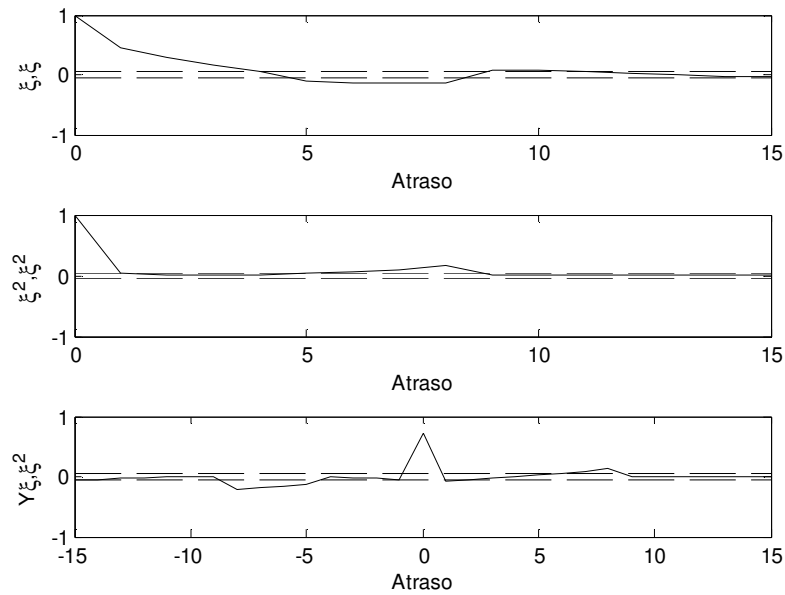


Figura 77: Gráficos de autocorrelação, autocorrelação de alta ordem e correlação cruzada de alta ordem entre a saída com o resíduo e o resíduo, para a série de Lorenz com as novas entradas adicionadas.

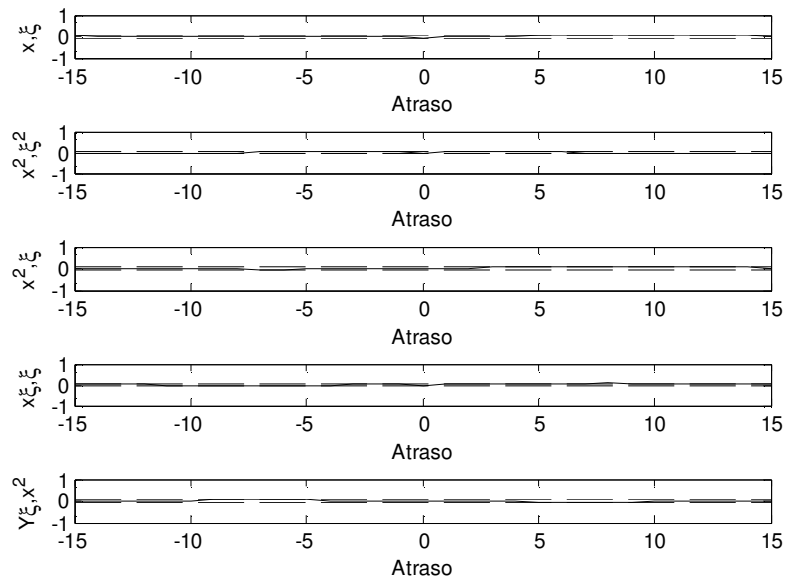


Figura 78: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k)$ da série de Lorenz com as novas entradas adicionadas.

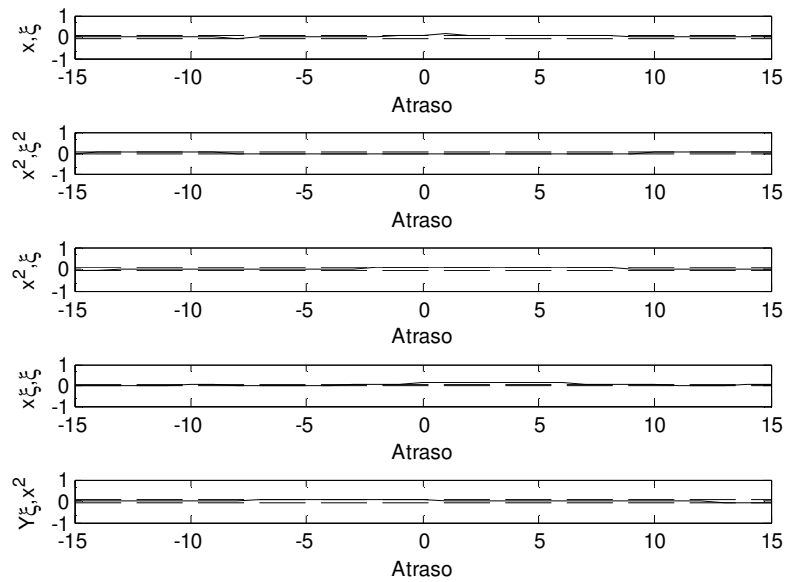


Figura 79: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 8)$ da série de Lorenz com as novas entradas adicionadas.

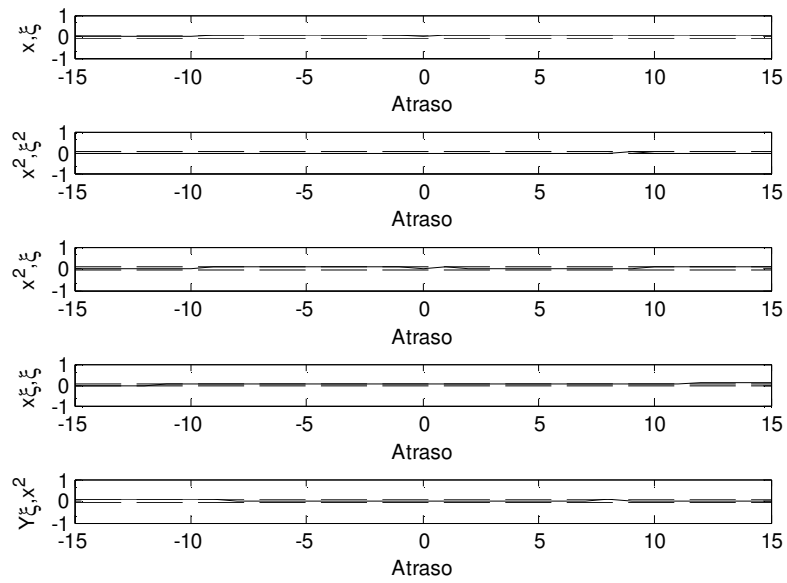


Figura 80: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 16)$ da série de Lorenz com as novas entradas adicionadas.

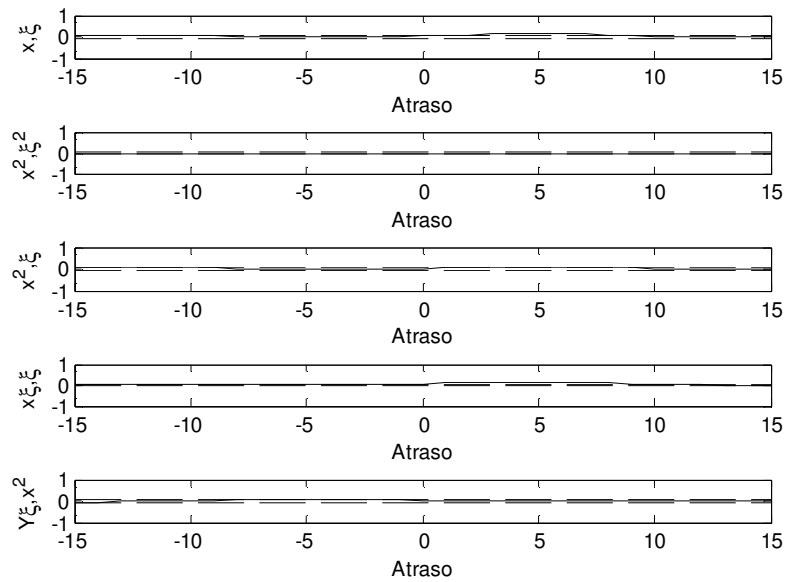


Figura 81: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 24)$ da série de Lorenz com as novas entradas adicionadas.

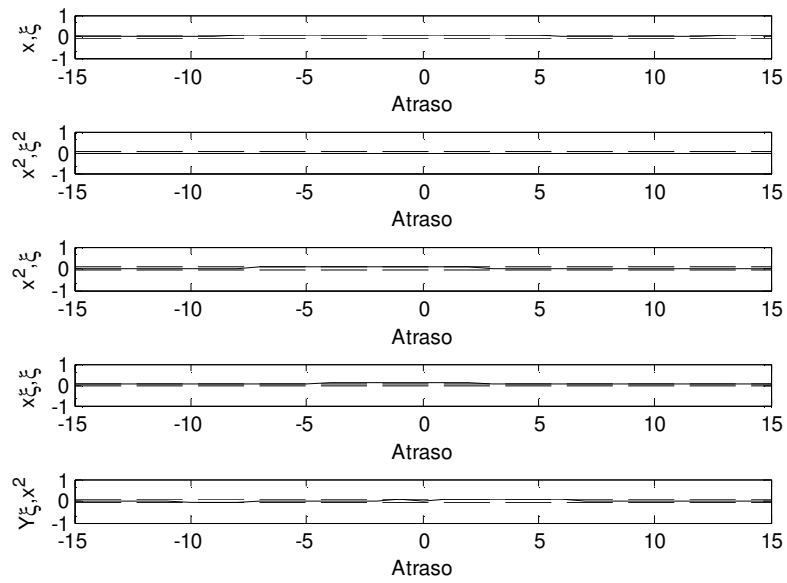


Figura 82: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 32)$ da série de Lorenz com as novas entradas adicionadas.

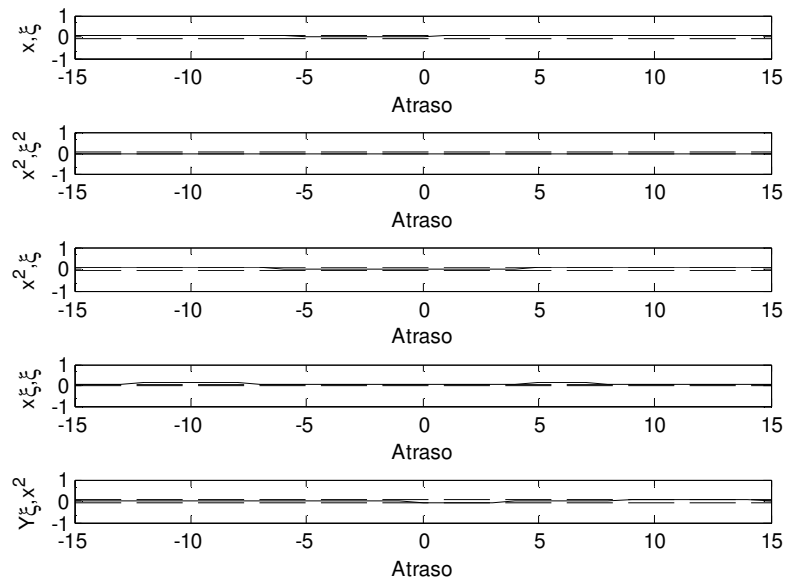


Figura 83: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k - 40)$ da série de Lorenz com as novas entradas adicionadas.

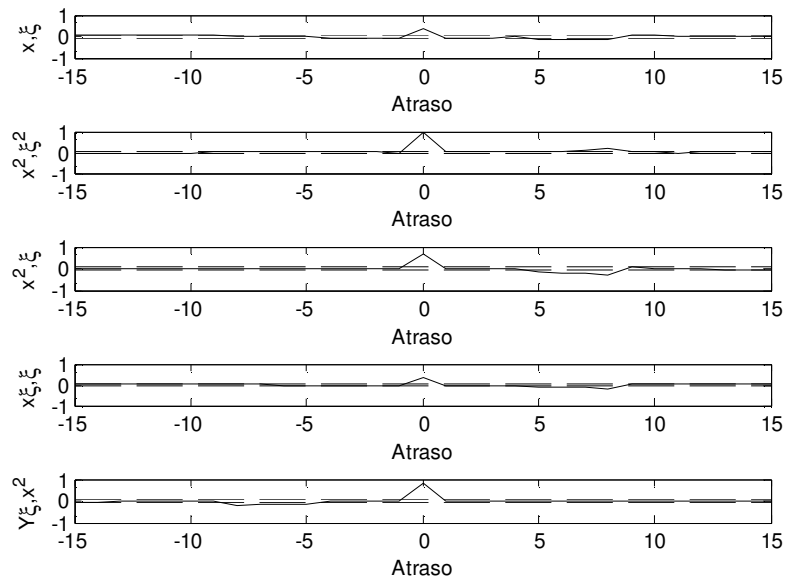


Figura 84: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $e(k - 1)$ da série de Lorenz com as novas entradas adicionadas.

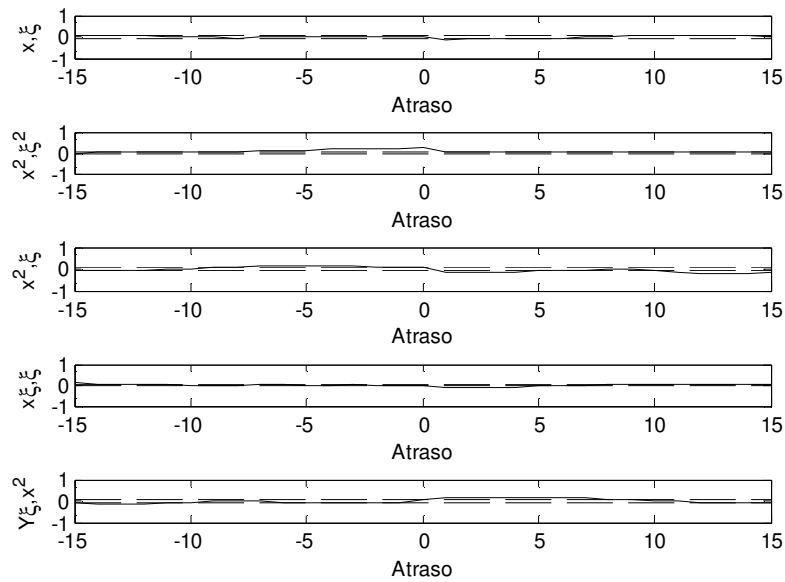


Figura 85: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $e(k - 8)$ da série de Lorenz com as novas entradas adicionadas.

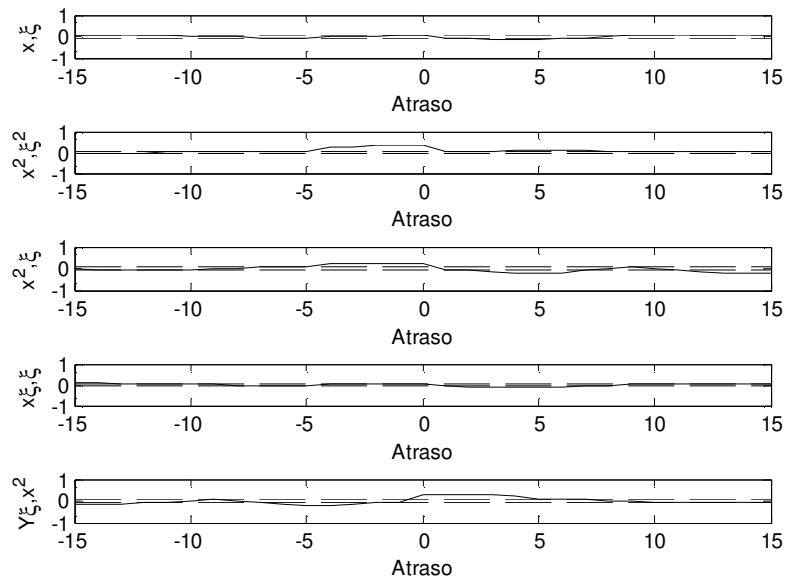


Figura 86: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $e(k - 5)$ da série de Lorenz com as novas entradas adicionadas.

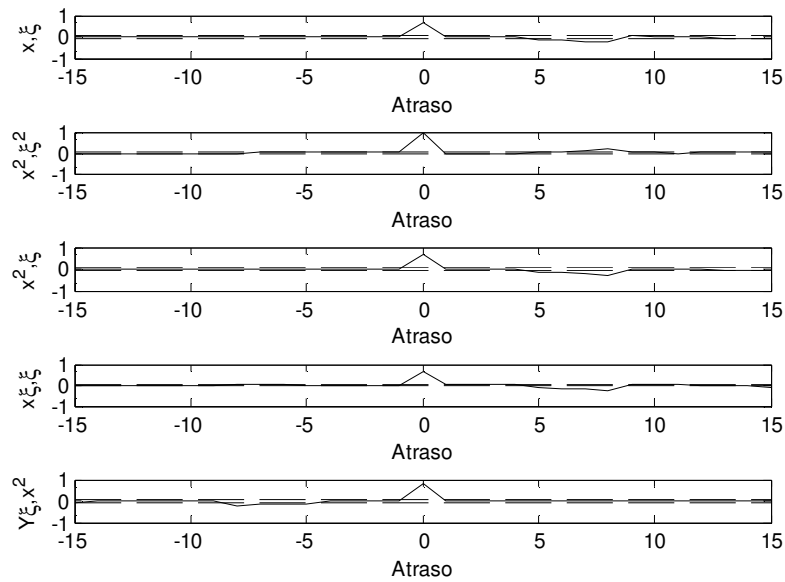


Figura 87: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $e(k - 1)e(k - 8)^3$ da série de Lorenz com as novas entradas adicionadas.

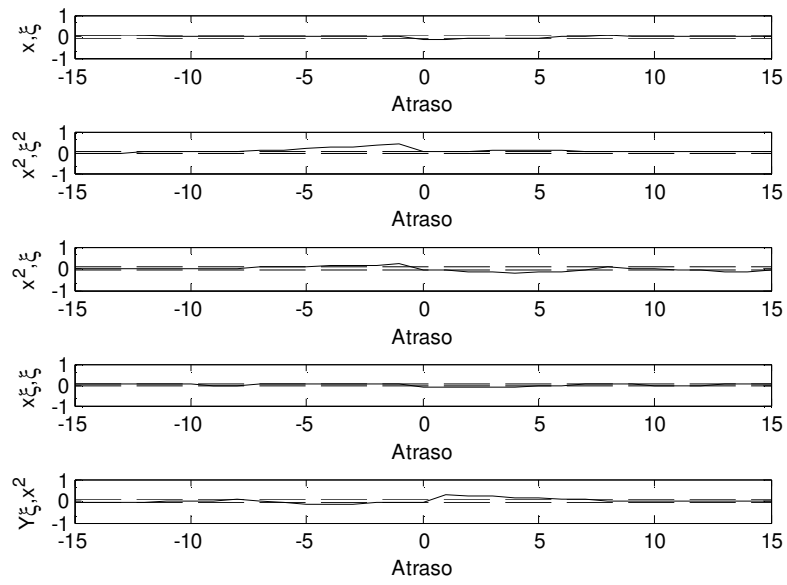


Figura 88: Gráficos de correlações cruzadas e correlações cruzadas de alta ordem para a entrada $x(k-1)e(k-8)$ da série de Lorenz com as novas entradas adicionadas.

