

RODRIGO CLEMENTE THOM DE SOUZA

**PREVISÃO DE SÉRIES TEMPORAIS
UTILIZANDO REDE NEURAL TREINADA POR
FILTRO DE KALMAN E EVOLUÇÃO
DIFERENCIAL**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Engenharia de Produção e Sistemas.

CURITIBA

2008

RODRIGO CLEMENTE THOM DE SOUZA

**PREVISÃO DE SÉRIES TEMPORAIS
UTILIZANDO REDE NEURAL TREINADA POR
FILTRO DE KALMAN E EVOLUÇÃO
DIFERENCIAL**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Engenharia de Produção e Sistemas.

Área de Concentração: Automação e Controle de Processos

Orientador: Prof. Dr. Leandro dos Santos Coelho

CURITIBA

2008

Souza, Rodrigo Clemente Thom de

Previsão de Séries Temporais Utilizando Rede Neural Treinada por Filtro de Kalman e Evolução Diferencial. Curitiba, 2008. 85p.

Dissertação – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Engenharia de Produção e Sistemas.

1. Previsão de Séries Temporais 2. Rede Neural Função de Base Radial 3. Filtro de Kalman Estendido 4. Evolução Diferencial. I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Engenharia de Produção e Sistemas.

Agradecimentos

Agradeço primeiramente a Deus, por ter me concedido as graças necessárias para a realização deste sonho.

Aos meus pais Gracinda Bianco Clemente de Souza e Antonio Roberto de Souza pelo imenso carinho e pelo verdadeiro amor dedicado ao meu crescimento, educação e formação como homem.

À minha esposa Cinthya Covessi Thom de Souza por sua companhia e pelo seu amor imensurável.

Ao meu irmão Roberto Clemente de Souza pelo companheirismo e caráter inigualáveis.

Ao meu orientador Leandro dos Santos Coelho por sua compreensão e pela sua impecável orientação a este trabalho.

Aos professores Wesley Vieira da Silva e Viviana Cocco Mariani pelas atenciosas contribuições oferecidas.

Aos demais parentes, amigos, professores e alunos que ao terem convivido comigo participaram também de alguma forma na conclusão deste trabalho.

Sumário

Agradecimentos.....	i
Sumário.....	ii
Lista de Figuras.....	iv
Lista de Tabelas.....	vi
Lista de Abreviaturas.....	vii
Resumo.....	ix
Abstract.....	x
1. Introdução.....	1
1.1. Justificativa da Pesquisa.....	3
1.2. Definição do Problema de Pesquisa.....	4
1.3. Objetivos da Pesquisa.....	4
1.4. Descrição da Proposta.....	5
1.5. Contribuição da Pesquisa.....	5
1.6. Caracterização da Pesquisa.....	6
1.7. Organização da Dissertação.....	7
2. Séries Temporais.....	8
2.1. Previsão de Séries Temporais.....	9
2.1.1. Modelos Lineares.....	13
2.1.1.1. Modelos Lineares Probabilísticos: ARIMA.....	13
2.1.2. Modelos Não-lineares.....	14
2.1.2.1. Modelos Não-lineares Gerais: RN.....	15
2.1.3. Avaliação de Desempenho dos Modelos.....	16
3. Fundamentação Teórico-Empírica.....	18
3.1. Redes Neurais.....	18

3.1.1. Redes Neurais RBF (Função de Base Radial).....	20
3.1.2. Arquitetura da RN-RBF.....	22
3.1.3. Previsão de Séries Temporais por meio de RN RBF.....	24
3.2. Filtro de Kalman.....	25
3.2.1. Filtro de Kalman Discreto.....	25
3.2.2. Filtro de Kalman Estendido.....	29
3.2.2.1. Treinamento de RN RBF por meio do EKF.....	32
3.3. Meta-Heurísticas.....	35
3.3.1. Evolução Diferencial.....	38
3.3.1.1. Operador de Mutação.....	40
3.3.1.2. Operador de Cruzamento.....	41
3.3.1.3. Operador de Seleção.....	42
3.3.1.4. Parâmetros de Projeto da ED.....	42
3.3.1.5. Otimização do EKF usando Evolução Diferencial.....	43
4. Apresentação e Análise dos Resultados.....	44
4.1. Coleta e Tratamento dos Dados.....	44
4.2. Simulações ARIMA.....	45
4.3. Simulações RN-RBF.....	50
4.4. Análises Comparativas.....	54
5. Conclusão.....	71
5.1. Limitações.....	72
5.2. Recomendações para Elaboração de Trabalhos Futuros.....	72
Referências.....	74

Lista de Figuras

Figura 1.1	Modelos híbridos inteligentes.....	2
Figura 2.1	Filtro linear com $M = 4$ para previsão um passo à frente.....	10
Figura 2.2	Filtro linear com $M = 4$ para previsão dois passos à frente.....	12
Figura 2.3	Modelos baseados na física, lineares gerais e não-lineares gerais.....	16
Figura 3.1	Taxonomia da computação natural.....	18
Figura 3.2	Arquitetura de uma rede neural RBF.....	24
Figura 3.3	Diagrama em blocos do sistema, modelo de medição e filtro de Kalman, onde q^{-1} é o operador atraso.....	28
Figura 3.4	Taxonomia dos métodos meta-heurísticos de otimização.....	38
Figura 3.5	Um exemplo de uma função custo bi-dimensional mostrando suas linhas de contorno e o processo para geração de $v_{i,G+1}$	41
Figura 3.6	Cruzamento (<i>crossover</i>) para $D = 8$ parâmetros.....	42
Figura 4.1	Série temporal do preço da soja adotada neste trabalho.....	46
Figura 4.2	Função de Auto-Correlação Amostral (ACF).....	47
Figura 4.3	Função de Auto-Correlação Parcial (PACF).....	47
Figura 4.4	ACF para os resíduos do modelo ARIMA (1,1,0).....	48
Figura 4.5	ACF para os resíduos do modelo ARIMA (1,1,1).....	48
Figura 4.6	PACF para os resíduos do modelo ARIMA (1,1,1).....	49
Figura 4.7	ACF para os resíduos do modelo ARIMA (1,1,2).....	49
Figura 4.8	PACF para os resíduos do modelo ARIMA (1,1,2)	50
Figura 4.9	Análise comparativa do <i>MSE</i> para os algoritmos de treinamento por descida de encosta e EKF otimizado por AG e por ED.....	55
Figura 4.10	Análise comparativa do tempo de processamento para os algoritmos de treinamento por descida de encosta e EKF otimizado por AG e por ED.....	56
Figura 4.11	Análise comparativa do <i>MSE</i> para os melhores modelos sem aplicação de médias móveis nas RN-RBF para previsão um passo à frente.....	57

Figura 4.12	Análise comparativa do <i>MSE</i> para a RN-RBF treinada por EKF otimizado por ED sem e com duas médias móveis.....	58
Figura 4.13	Análise comparativa do <i>MSE</i> para a RN-RBF treinada por descida de encosta sem e com duas médias móveis.....	59
Figura 4.14	Análise comparativa do <i>MSE</i> para a RN-RBF treinada por EKF otimizado por AG sem e com duas médias móveis.....	60
Figura 4.15	Análise comparativa do <i>MSE</i> para os melhores modelos aplicando-se duas médias móveis nas RN-RBF para previsão um passo à frente.....	61
Figura 4.16	Análise comparativa entre o preço real (desejado) e preço estimado do melhor modelo.....	62
Figura 4.17	Erro de previsão do melhor modelo.....	62
Figura 4.18	<i>MSE</i> para previsão 20 passos à frente.....	63
Figura 4.19	Comparação entre os valores reais e os valores previstos pelos três métodos de RN-RBF com as 20 primeiras amostras de validação.....	64
Figura 4.20	Comparação entre os erros de previsão dos três métodos de RN-RBF com as 20 primeiras amostras de validação.....	64
Figura 4.21	<i>MSE</i> para previsão com todas as amostras de validação (687 passos à frente).....	65
Figura 4.22	Comparação entre os valores reais e os valores previstos pelos três métodos de RN-RBF com todas as amostras de validação (687 passos à frente).....	65
Figura 4.23	Comparação entre os erros de previsão dos três métodos de RN-RBF com todas as amostras de validação (687 passos à frente).....	65
Figura 4.24	Série temporal do preço do café adotada neste trabalho.....	67
Figura 4.25	Análise comparativa do <i>MSE</i> para os algoritmos de treinamento por descida de encosta e EKF otimizado por AG e por ED para previsão um passo à frente dos preços do café.....	68
Figura 4.26	Série temporal do preço do boi gordo adotada neste trabalho.....	69
Figura 4.27	Análise comparativa do <i>MSE</i> para os algoritmos de treinamento por descida de encosta e EKF otimizado por AG e por ED para previsão um passo à frente dos preços do boi gordo.....	70

Lista de Tabelas

Tabela 4.1	Parâmetros da RN-RBF.....	51
Tabela 4.2	Resultados da simulação do algoritmo de treinamento por descida de encosta sem aplicação de médias móveis.....	52
Tabela 4.3	Resultados da simulação do algoritmo de treinamento por EKF otimizado por AG sem aplicação de médias móveis.....	53
Tabela 4.4	Resultados da simulação do algoritmo de treinamento por EKF otimizado por ED sem aplicação de médias móveis.....	54
Tabela 4.5	Melhores modelos sem aplicação de médias móveis na RN-RBF um passo à frente.....	56
Tabela 4.6	Resultados da simulação do algoritmo de treinamento por EKF otimizado por ED com aplicação de duas médias moveis para previsão um passo à frente.....	58
Tabela 4.7	Resultados da simulação do algoritmo de treinamento por descida de encosta com aplicação de duas médias móveis para previsão um passo à frente.....	59
Tabela 4.8	Resultados da simulação do algoritmo de treinamento por EKF otimizado por ED com aplicação de duas médias móveis para previsão um passo à frente.....	60
Tabela 4.9	Melhores modelos aplicando-se duas médias móveis na RN-RBF para previsão um passo à frente.....	61
Tabela 4.10	Análise comparativa do MSE para os algoritmos de treinamento por descida de encosta e EKF otimizado por AG e por ED para previsão um passo à frente dos preços do café.....	67
Tabela 4.11	Análise comparativa do MSE para os algoritmos de treinamento por descida de encosta e EKF otimizado por AG e por ED para previsão um passo à frente dos preços do boi gordo.....	69

Lista de Abreviaturas

ABIOVE	Associação Brasileira das Indústrias de Óleos VEgetais
ACF	Função de Auto-Correlação Amostral
ADF	Dickey-Fuller Aumentado
AG(s)	Algoritmo(s) Genético(s)
AR	Auto Regressivo
ARCH	Auto Regressivo com Heteroscedasticidade Condicional
ARMA	Auto Regressivo com Médias Móveis
ARIMA	Auto Regressivo Integrado com Médias Móveis
BDS	Brock, Dechert e Scheinkman
BM&F	Bolsa de Mercadorias & Futuros
BOVESPA	Bolsa de Valores de São Paulo
CEPEA	Centro de Estudos Avançados em Economia Aplicada
CONCEX	Conselho Nacional do Comércio Exterior
DF	Dickey-Fuller
ED	Evolução Diferencial
EKF	Filtro de Kalman Estendido
ESALQ	Escola Superior de Agricultura “Luiz de Queiroz”
FIR	Resposta de Impulso Finito
GARCH	Auto Regressivo com Heteroscedasticidade Condicional Generalizado
IA	Inteligência Artificial
KF	Filtro de Kalman
MA	Médias Móveis
MATLAB	<i>MATrix LABoratory</i>
MLP	<i>Perceptron</i> Multicamadas
MSE	Erro Quadrático Médio
NARMAX	Auto Regressivo com Médias Móveis Não-Linear com Entradas Exógenas

NP	Número de indivíduos da população
NPR	Nota Promissória Rural
OC	Otimização Combinatória
PACF	Função de Auto-Correlação Parcial
PID	Proporcional, Integral e Derivativo
RBF	Função de Base Radial
RNs	Redes Neurais
SEs	Sistemas Especialistas
SE	Erro Padrão
SNs	Sistemas Nebulosos

Resumo

As Redes Neurais Função de Base Radial (RNs-RBF) são modelos não-lineares que podem ser treinados para implementar um mapeamento (interpolação) de entrada e saída desejado. Esta capacidade de generalização aliada ao baixo custo computacional inerente a RN-RBF possibilita eficiência quando aplicadas à previsão de séries temporais (em especial, séries não-lineares, como é o caso de séries temporais financeiras de *commodities* do agro-negócio). Entretanto, o desempenho apropriado das RNs-RBF depende do número de centros de funções de base radial, de suas formas e do método utilizado para aprendizagem dos mapeamentos de entrada e saída. Vários algoritmos têm sido propostos recentemente na literatura para treinamento de RNs-RBF. A seleção dos pesos da RN-RBF pode ser abordada como um problema de otimização. Diferentes derivações do Filtro de Kalman para tais fins estão disponíveis na literatura, já que a aplicação de tais filtros em sistemas não-lineares se mostrou eficiente para sistemas cuja não-linearidade é não significativa. Para sistemas não-lineares com ruído colorido, o filtro de Kalman pode ser estendido pela linearização da matriz de transição não-linear da descrição em variáveis de estado do sistema em torno dos parâmetros estimados correntes. Este filtro, denominado Filtro de Kalman Estendido (EKF, *Extended Kalman Filter*) atualiza os parâmetros de forma a ser consistente com todos os dados medidos do processo e geralmente converge em poucas iterações. A Evolução Diferencial (ED) é um algoritmo evolutivo que utiliza uma abordagem “gulosa” e menos estocástica para resolver problemas do que fazem os algoritmos evolutivos clássicos, tais como algoritmo genético, programação evolutiva e estratégia evolutiva. Neste contexto, este trabalho propõe o uso do EKF para minimização do erro de treinamento de RN-RBF na previsão de séries temporais financeiras de *commodities* do agro-negócio, e a ED para determinação dos parâmetros de sintonia do EFK.

Palavras-Chave: Previsão de Séries Temporais, Rede Neural Função de Base Radial, Filtro de Kalman Estendido, Evolução Diferencial.

Abstract

Radial Basis Function Neural Networks (RBF-NNs) are nonlinear models that can be trained to implement a desirable input and output mapping (interpolation). This generalization skill and the low computational cost of the RBF-NN enables efficiency when applied to time series forecasting (especially, non-linear, as financial time series of agri-business commodities). The appropriate performance of the RBF-NN depends on the number of centers of the radial basis functions, of its forms, and the method used to learn the mapping of inputs and outputs. Several algorithms have been proposed to the training of the RBF-NN. The selection of the weights of RBF-NN can be understood as an optimization problem. The derivations of the Kalman Filter for such purposes are widely available in the literature, since the implementation of such filters in non-linear was efficient for systems whose non-linearity does not exist. For non linear systems with colored noise, the Kalman filter can be extended by the linearization of the non-linear transition matrix of the description on the status variables of the system around the current estimated parameters. This filter, called Extended Kalman Filter (EKF) updates the parameters in order to be consistent with all the measured data of the process and usually converges in a few iterations. Differential Evolution is an evolutionary algorithm that uses a greedy and less stochastic approach to solve problems than the classic algorithms, such as genetic algorithm, evolutionary programming and evolution strategy. In this context, this work proposes the use of EKF to decrease the training error of RBF-NN in the commodities agri-business financial time series forecast, and the differential evolution to determine the tuning parameters of EKF.

Keywords: Time Series Forecasting, Radial Basis Function Neural Network, Extended Kalman Filter, Differential Evolution.

1. Introdução

Caso seja conhecido o comportamento passado do valor de um sinal até um determinado instante de tempo é possível fazer alguma inferência sobre seus valores futuros. Tal processo de inferência é conhecido como previsão.

A previsão de séries temporais é uma área de estudo relevante, tendo em vista os inúmeros exemplos de séries temporais encontrados em áreas do conhecimento, tais como Física (WEIGEND, 1994), Astrofísica (WEIGEND, 1990), Meteorologia (FERRAZ, 1999), Engenharia Elétrica (MARTIN, 2005), Ciência da Computação (LELAND, 1994) e Ciências Econômicas (MADDALA, 2001), objeto deste estudo.

As pesquisas de previsão de séries temporais financeiras tratam, entre outros temas, do acompanhamento da taxa de câmbio entre moedas (SANTOS, 2005; CASTRO, 2001), previsão de insolvência de empresas (SCARPEL, 2005), análise do mercado de ações (NYGREN, 2004; BRAGA, 2006) e previsão de preços de *commodities* (SOUZA, 2007).

Alheio a isso, a área de finanças pode ser analisada como um sistema complexo formado por elementos que interagem entre si, geralmente de modo não-linear.

Isto significa que as séries temporais financeiras são influenciadas por diversos fatores (por exemplo: preço internacional, condições climáticas e meteorológicas, variáveis macroeconômicas, oferta e demanda, sazonalidade, câmbio e preço de produtos substitutos) que, de modo combinado, fazem com que estas séries apresentem comportamento não-linear.

No âmbito da Inteligência Computacional (IC), a partir dos anos 1990, alguns estudos se voltaram à criação e utilização de modelos híbridos inteligentes (KARR, 1991; MEDSKER, 1992). Tais sistemas eram respaldados em dois ou mais modelos de IC, sob o pretexto de melhorar resultados ou transpor limitações de um modelo de IC em particular.

Exemplificando, os algoritmos genéticos (AGs) foram empregados na especificação de arquiteturas iniciais de redes neurais artificiais, os sistemas especialistas (SEs) foram aplicados a redes neurais artificiais (RNs), e os sistemas nebulosos (SNs) foram empregados no tratamento de incertezas em SEs, entre outras formas de hibridização, tal como evidencia a Figura 1.1.

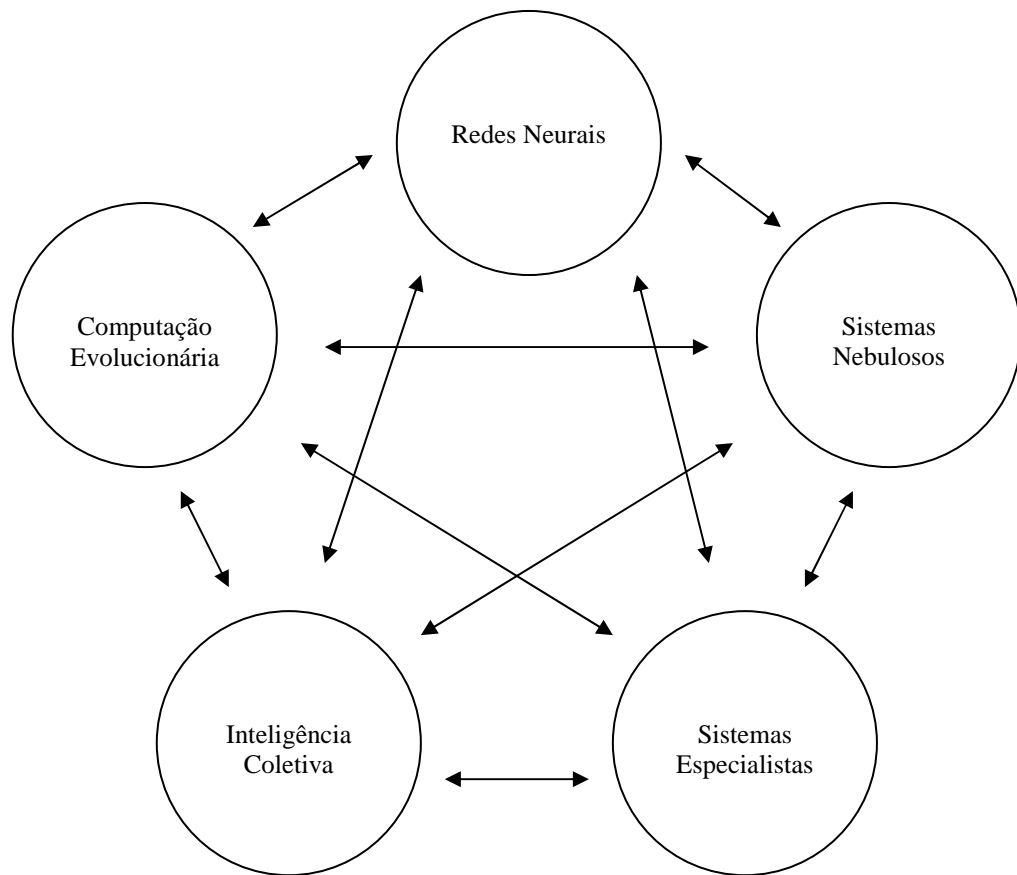


Figura 1.1: Modelos híbridos inteligentes.

Uma rede neural artificial que pode ser aplicada com sucesso à previsão de séries temporais é a Função de Base Radial (RBF, *Radial Basis Function*). Por outro lado, o filtro de Kalman (KF, *Kalman Filter*) é uma ferramenta que possibilita lidar com informações ruidosas do sistema dinâmico, utilizando informações matemáticas para estimar o estado atual do ambiente no qual está inserido. Finalmente, a evolução diferencial (ED) proposta por STORN e PRICE (1995) é um método de otimização da computação evolucionária (ou evolutiva) guloso em termos de busca, de rápida convergência, robusto e de fácil implementação.

Deste modo, supõe-se que a aplicação combinada destas ferramentas para a modelagem e previsão de sistemas com comportamento não-linear possa produzir resultados satisfatórios para a previsão de séries temporais (SOUZA, 2007), ou até mesmo superiores que os das principais ferramentas conhecidas atualmente, tal como o método proposto por BOX e JENKINS (1976).

Uma abordagem é proposta combinando a RN-RBF, uma adaptação do KF e a ED para a previsão de séries temporais. Os resultados desta abordagem são comparados, portanto, com o desempenho de outras abordagens baseadas em modelos lineares e não-lineares.

A partir da análise de um estudo de caso, esta pesquisa pretende prover algumas conclusões sobre a capacidade preditiva do modelo não-linear proposto quando aplicado a séries temporais. A série temporal utilizada no estudo é a série de preços diários da soja no Estado do Paraná.

1.1. Justificativa da Pesquisa

Uma parte do interesse pelas RNs-RBF é proveniente do fato delas possuírem fundamentação estatística e serem consideradas aproximadores universais de funções com dada precisão (CHEN *et al.*, 1991), realizando um mapeamento das entradas de rede de dimensão vetorial m , em uma saída de dimensão vetorial n de qualquer função vetorial contínua.

A RN-RBF apresenta potencialidades de generalização local e seu aprendizado é relativamente rápido. Tais potencialidades, úteis para a previsão de séries temporais, são amplamente difundidas na literatura, destacando-se HUTCHINSON (1994) e KUAN (1995).

Recentemente, a utilização de diferentes abordagens de métodos de treinamento para as RNs-RBF também tem sido muito estudadas. Exemplos de tais estudos são a aplicação de métodos tais como mínimos quadrados ortogonal (BILLINGS, 1998) e agrupamentos de dados (GUERRA, 2006).

YAO (1999) estudou a utilização de algoritmos de computação evolucionária de modo combinado às redes neurais. COELHO e MARIANI (2006) propuseram um controlador PID (proporcional, integral e derivativo) híbrido neuro-genético, GOLDBERG (1989) e DOGANIS *et al.* (2006) detalharam métodos de treinamento de RNs com base em AGs e DU (2006) baseado na Evolução Diferencial (ED). Mais recentemente, outros pesquisadores que contribuíram com a avaliação da ED como método de treinamento para RNs foram BABU (2007) e BASTURK (2008).

Na literatura, abordagens de treinamento de redes neurais usando filtro de Kalman têm sido propostas, tais como as de PUSKORJUS (1994), BIRGMEIER (1995), XIONG (2003), BAI (2004), SCHAUER *et al.* (2005) e ZHANG (2005).

SIMON (2002) aplicou o Filtro de Kalman Estendido (EKF, *Extended Kalman Filter*) na otimização dos parâmetros de uma RN-RBF, visando diminuir o erro de treinamento. Neste mesmo trabalho, o autor propôs inclusive um filtro de Kalman desacoplado para melhorar o desempenho, já que o EKF é uma solução com elevado esforço computacional. CIOCOIU (2002) utilizou um EKF duplo para treinamento de uma RN-RBF.

A contribuição deste trabalho é avaliar as potencialidades de usar uma abordagem de treinamento de RN-RBF baseada em EKF com otimização usando evolução diferencial para previsão de séries temporais do preço da soja.

1.2. Definição do Problema de Pesquisa

Modelos não-lineares híbridos de previsão de séries temporais são capazes de gerar previsões de curto, médio e longo prazo mais precisas que modelos lineares e modelos não-lineares não combinados?

1.3. Objetivos da Pesquisa

Os modelos não-lineares híbridos a serem analisados são as RNs-RBF com treinamento baseado em filtro de Kalman estendido e evolução diferencial. Os modelos não-lineares tradicionais são a própria RN-RBF com treinamento baseado no algoritmo de descida de encosta (gradiente descendente) e a otimização do filtro de Kalman por AG, o mais difundido algoritmo de computação evolutiva.

Já o modelo linear é o modelo de BOX e JENKINS (1976) ou modelo auto-regressivo integrado de média móvel (ARIMA, do inglês, *Auto Regressive Integrated Moving Average*), e são utilizadas séries de preços de fechamento mensal da soja em grão no Estado do Paraná entre os meses de Fevereiro de 1998 e Maio de 2008.

Enquanto parcela destes estudos mostram que modelos não-lineares conseguem uma previsão um passo à frente mais acurada que os modelos lineares (KAASHOEK, 2002; FÉLIX *et al.*, 2003; KUAN, 1995), outra parte dos estudos chega a uma conclusão divergente (STOCK, 1999; NEELY, 2003; SWANSON, 1995) tornando a questão controversa. Este estudo pretende corroborar com a hipótese de superioridade de modelos não-lineares, principalmente, devido ao fato da série em estudo apresentar comportamento não-linear.

Uma vez que parte dos estudos empíricos tem comparado somente os modelos lineares e não-lineares tradicionais, outra idéia subjacente ao estabelecimento do objetivo geral é tentar fornecer mais uma contribuição teórica e empírica à questão relativa ao uso de modelos não-lineares híbridos em detrimento de modelos lineares, bem como de modelos não-lineares tradicionais. Deste modo, este estudo visa também complementar a base de conhecimento científica atual para elucidar esta questão.

Os objetivos específicos desta dissertação são:

- Estudar e implementar modelos estatísticos clássicos de previsão de séries temporais;
- Estudar e implementar, isoladamente e de forma combinada, modelos de RN-RBF, filtro de Kalman e evolução diferencial para previsão de séries temporais financeiras;
- Analisar comparativamente o desempenho dos modelos implementados para previsão de séries temporais financeiras.

1.4. Descrição da Proposta

Nesta dissertação é apresentado um estudo comparativo de desempenho de quatro métodos de previsão de séries temporais financeiras: (i) ARIMA, (ii) RN-RBF com treinamento baseado em descida de encosta, (iii) RN-RBF com treinamento baseado em EKF otimizado por AG e (iv) RN-RBF com treinamento baseado em EKF otimizado por ED. Estes métodos são utilizados na previsão da série não-linear de preços diários da soja.

1.5. Contribuição da Pesquisa

A contribuição deste trabalho é na elucidação da controversa questão a respeito do desempenho de métodos não-lineares frente a métodos lineares na previsão de séries temporais financeiras. Além disso, o uso da meta-heurística de Evolução Diferencial na sintonização dos melhores parâmetros do EKF, algoritmo utilizado para treinamento da RN-RBF, é um diferencial deste trabalho em relação ao que existe catalogado na literatura científica a respeito de métodos de previsão de séries temporais.

1.6. Caracterização da Pesquisa

Segundo JUNG (2004), as pesquisas científicas podem ser classificadas quanto à natureza, objetivos, procedimentos, fonte de referências, além de ambiente e tempo de aplicação.

Em relação à natureza, esta pesquisa enquadra-se como tecnológica (JUNG, 2004), uma vez que conhecimentos básicos são aplicados e novos conhecimentos são gerados como resultado do processo de pesquisa.

Quanto aos objetivos, esta pesquisa apresenta-se como explicativa. As pesquisas explicativas, segundo JUNG (2004), visam à “identificação dos fatores que contribuem para a ocorrência dos fenômenos ou variáveis que afetam o processo”. Assim, nessa pesquisa, o objetivo é corroborar com a hipótese da influência de determinados fatores na previsão do preço da soja por meio da aplicação de técnicas que modelam estes fatores.

Referente aos procedimentos, essa pesquisa caracteriza-se como operacional. Segundo JUNG (2004), a pesquisa operacional “tem por princípio a investigação de forma sistemática e racional dos processos envolvidos” e utiliza ferramentas estatísticas e métodos matemáticos para obtenção da melhor solução. Dessa forma, observa-se que essa pesquisa utiliza análises estatísticas e métodos matemáticos para a avaliação dos resultados.

Com relação à fonte de referências essa pesquisa é bibliográfica, pois segundo JUNG (2004), a pesquisa bibliográfica procura identificar as diferentes contribuições existentes na literatura sobre o tema. Neste estudo, revisa-se a literatura a respeito de métodos lineares e não-lineares de previsão de séries temporais.

O ambiente de aplicação desta pesquisa é laboratorial. A pesquisa em laboratório, segundo JUNG (2004), “caracteriza-se pela possibilidade de se controlar as variáveis que possam interferir no experimento”. Assim, esta pesquisa investiga única e exclusivamente os dados da série de fechamento diário do preço da saca da soja no Estado do Paraná.

Além disso, o fato de utilizar os preços da soja dentro de um intervalo de tempo determinado, no caso entre Fevereiro de 1998 e Maio de 2008, faz com que esta pesquisa seja caracterizada quanto ao tempo de aplicação como longitudinal (JUNG, 2004).

1.7. Organização da Dissertação

A presente dissertação é dividida da seguinte forma: No capítulo 2, a descrição do problema é apresentada, detalhando os principais conceitos relacionados a séries temporais, seus modelos de previsão e critérios de desempenho. No capítulo 3 são apresentadas as fundamentações teóricas e empíricas, onde são detalhados os conceitos de redes neurais RBF, o filtro de Kalman estendido e a meta-heurística de Evolução Diferencial. No capítulo 4 são mostrados os resultados obtidos e as análises comparativas entre os modelos analisados. No capítulo final são apresentadas as conclusões, além limitações da pesquisa e de algumas propostas para continuidade da mesma.

2. Séries Temporais

Uma série temporal é uma coleção de observações realizadas sequencialmente ao longo do tempo. Parte dos procedimentos estatísticos, por exemplo, a análise de dados em secção transversal, foi desenvolvida para analisar observações independentes (SILVA, 2004). Todavia, uma característica importante das séries temporais é que as observações vizinhas são, em geral, dependentes. Portanto, um modo simples de se estudar o comportamento destas séries é a partir da análise e da modelagem destas dependências (EHLERS, 2007).

O estudo de séries temporais requer o uso de técnicas específicas. Algumas características são particulares a este tipo de dados, por exemplo:

- Observações correlacionadas são mais difíceis de analisar e requerem técnicas específicas;
- A ordem temporal das observações deve ser considerada;
- Fatores complicadores, tais como a presença de tendências e a variação sazonal ou cíclica podem ser difíceis de estimar ou remover;
- A seleção de modelos pode ser complicada, e as ferramentas podem ser de difícil interpretação;
- É difícil de lidar com observações perdidas e com dados discrepantes devido à natureza sequencial.

Uma série temporal é dita contínua quando as observações são realizadas continuamente no tempo ou discreta quando as observações são realizadas em tempos específicos, geralmente equiespaçados. Note que estes termos não se referem à variável observada, esta pode assumir valores discretos ou contínuos.

Por outro lado, séries temporais discretas podem surgir de várias formas. As séries contínuas podem ser “discretizadas”, isto é, seus valores são registrados a certos intervalos de tempo. As séries de valores agregados ou acumulados em intervalos de tempo, por exemplo, pode se citar as exportações medidas mensalmente ou a quantidade de chuva medida diariamente. Finalmente, algumas séries são inerentemente discretas, por exemplo, dividendos pagos por uma empresa aos seus acionistas em anos sucessivos.

Em algumas situações, o objetivo da análise de séries temporais pode ser realizar previsões de valores futuros enquanto em outras a estrutura da série ou sua relação com outras séries pode ser o interesse principal. De um modo geral, os principais objetivos em se estudar séries temporais podem ser os seguintes:

- *Descrição*: propriedades da série, por exemplo, o padrão de tendência, existência de variação sazonal, *outliers*, alterações estruturais, entre outros;
- *Explicação*: usar a variação em uma série para explicar a variação em outra série;
- *Controle de processos*: por exemplo, controle estatístico de qualidade;
- *Predição*: prever valores futuros com base em valores passados.

O interesse desta dissertação é neste último objetivo particular, a Predição de Séries Temporais.

A especificação e estimação de modelos lineares de séries temporais, baseados em modelos univariados ARIMA, são procedimentos bem estabelecidos. Entretanto, a teoria econômica frequentemente sugere um relacionamento não-linear entre as variáveis (SKOTT, 1989), e alguns economistas crêem que o sistema econômico como um todo é não-linear (LEE *et al.*, 1993). Dessa forma, é interessante testar se uma determinada série temporal na área econômica pode ser modelada por um modelo linear ou se é melhor reproduzida por um modelo não-linear (SANTOS, 2005).

A maioria dos testes de não-linearidade tem como primeiro passo a extração de uma estrutura linear a partir do uso de um filtro. Tipicamente, um modelo $AR(p)$ é ajustado a série e então o teste é aplicado aos resíduos obtidos. Para otimizar o processo, um valor particular inteiro de p é escolhido (usualmente $p = 1$ ou $p = 2$), entretanto é possível escolher p a partir de algum critério de informação (LEE *et al.*, 1993).

Um teste apropriado para detectar não-linearidades, desenvolvido a partir de pesquisas relacionadas à presença de caos em dados econômicos, é o teste *Brock, Dechert and Scheinkman* (BDS) proposto por BROCK *et al.* (1987). Detalhes referentes a este teste podem ser encontrados em BROCK *et al.* (1987).

2.1. Previsão de Séries Temporais

A previsão linear considera o problema de predição da amostra $y(n)$, subsequente a um conjunto conhecido de amostras consecutivas prévias $\{y(n-1), y(n-2), y(n-3), \dots\}$

pertencentes a uma série temporal discreta, problema este conhecido como previsão de curto prazo, ou previsão um passo à frente (CASTRO, 2001).

Em previsão linear, a estimativa da amostra predita, $\hat{y}(n)$, é expressa como uma combinação linear de M amostras prévias $\{y(n-1), y(n-2), \dots, y(n-M)\}$. Os coeficientes W_k , onde $k = 1, 2, \dots, M$, que ponderam tal combinação linear definem um filtro FIR (*Finite Impulse Response*).

A Figura 2.1 detalha um preditor FIR de ordem M , o qual é mostrado no instante n . Portanto, como o instante é definido não se faz necessária a explicitação do indexador n para as variáveis envolvidas, a menos que n não seja inequivocadamente definido pelo contexto.

Um preditor linear de ordem M utiliza M amostras previamente conhecidas da série temporal para estimar $y(n)$. A função custo J mede o erro médio quadrático entre a estimativa da previsão $\hat{y}(n)$ e o valor efetivamente obtido para a amostra em questão, $y(n)$. O vetor \underline{W} é um vetor de ponderações que define o filtro FIR que tem seus coeficientes determinados de forma a minimizar a função custo J (para qual \underline{W} , J é mínimo?).

Conforme pode ser observado na Figura 2.1, a amostra predita $\hat{y}(n)$ é dada por:

$$\hat{y}(n) = \sum_{k=1}^M w_k y(n-k) = \underline{W}^T \underline{y}, \quad (2.1)$$

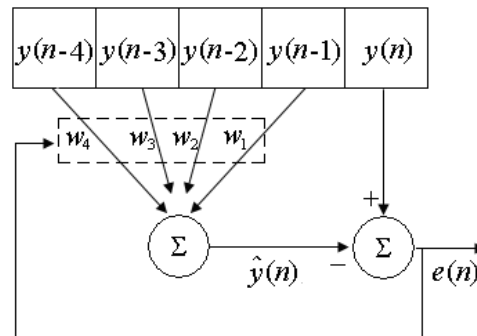


Figura 2.1 – Filtro linear com $M = 4$ para previsão um passo à frente.

Da Figura 2.1, o erro de predição $e(n)$ pode ser expresso por:

$$e(n) = y(n) - \hat{y}(n), \quad (2.2)$$

e a função custo J por:

$$J = E\{e^2\}. \quad (2.3)$$

Isto ocorre porque, na prática, não se conhece o processo aleatório subjacente que determina a série temporal em questão. Portanto, não são conhecidas as funções de

correlações que são realmente envolvidas no processo. Assim, o operador $E\{.\}$ é substituído pela média dos vetores de M componentes envolvidos no cálculo de \mathbf{R} e \underline{P} , média esta realizada sobre o intervalo de N_t amostras totais da série temporal.

Desta maneira, a previsão linear só tem sentido quando o processo aleatório subjacente é estacionário, caso contrário, \mathbf{R} e \underline{P} não são univocamente definidas, mesmo para N_t suficientemente grande. Ou seja, se a série temporal resulta de um processo aleatório não-estacionário, \mathbf{R} e \underline{P} variam ao longo da série. A solução algumas vezes adotada é assumir que a série temporal é estacionária em intervalos e adaptar \mathbf{R} e \underline{P} para cada intervalo. No entanto, o número de amostras em cada intervalo nem sempre é suficiente para expressar com fidelidade a operação $E\{.\}$.

Esta é a razão do uso de modelos de previsão não-linear, os quais, embora apresentem custo computacional maior, contornam a necessidade do conhecimento de um número grande de amostras passadas da série a ser predita, suficientes para que o operador $E\{.\}$ seja aproximado com fidelidade pela média temporal (DELURGIO, 1998).

A previsão linear também pode considerar o problema de predição da amostra $y(n+k)$, referente a um conjunto conhecido de amostras consecutivas prévias $\{y(n-1), y(n-2), y(n-3), \dots\}$ pertencentes a uma série temporal discreta sem que seja conhecido o intervalo de amostras de $y(n)$ a $y(n+k-1)$, problema este conhecido como previsão de longo prazo, ou previsão k passos à frente.

A estimativa da amostra $\hat{y}(n+k)$ é feita com base nos resultados da combinação das previsões das amostras do intervalo desconhecido de $y(n)$ a $y(n+k-1)$. Deste modo, primeiramente é realizada a previsão do valor de $\hat{y}(n)$. Com base no resultado de $\hat{y}(n)$, realiza-se a previsão do valor de $\hat{y}(n+1)$, com base neste, o de $\hat{y}(n+2)$, e assim sucessivamente até encontrar o valor de $\hat{y}(n+k)$.

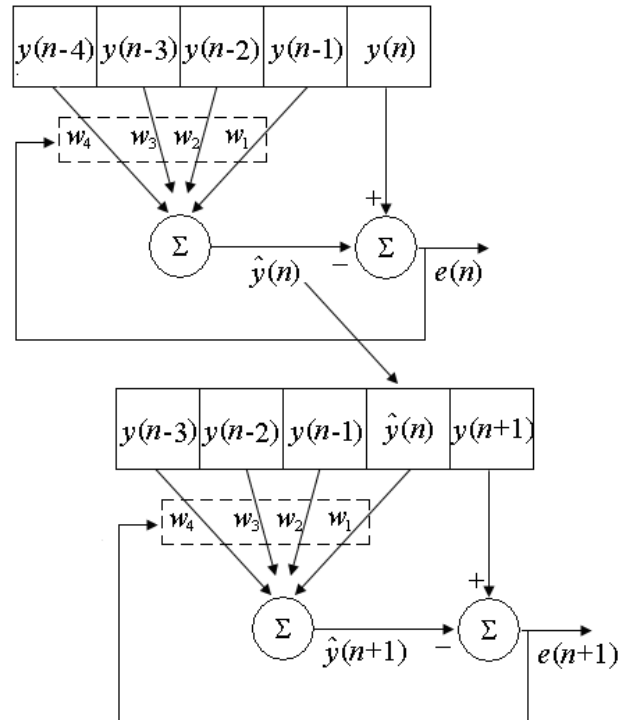


Figura 2.2 – Filtro linear com $M = 4$ para previsão dois passos à frente.

Os erros de cada previsão são propagados e, devido a este fato, o erro final da previsão da amostra $\hat{y}(n)$, denominado $e(n+k)$, é amplificado pelos erros das previsões das amostras anteriores. A Figura 2.2 apresenta um filtro linear com $M = 4$ para previsão de longo prazo com $k = 2$.

As abordagens de previsão de séries temporais encontradas na literatura podem ser organizadas da seguinte forma:

- *Técnicas Descritivas*: Ao se analisar uma ou mais séries temporais a representação gráfica dos dados sequencialmente ao longo do tempo é fundamental e pode revelar padrões de comportamento importantes. A tendência de crescimento (ou decrescimento), padrões cíclicos, alterações estruturais, observações aberrantes, entre outros, são geralmente identificados. Estas técnicas não são foco deste trabalho;
- *Modelos lineares*: incluem modelos probabilísticos, análise espectral, métodos não paramétricos (alisamento ou suavização), modelos de espaço de estados, séries multivariadas, estudos longitudinais e processos de longa dependência;
- *Modelos não-lineares*: englobam modelos não-lineares gerais (redes neurais artificiais, sistemas nebulosos, filtro de Kalman estendido, modelos híbridos), modelos pré-definidos, modelos com volatilidade variável, entre outros.

2.1.1. Modelos Lineares

Considerar-se-á vários tipos de modelagens lineares, tais como análise espectral, métodos não paramétricos (alisamento ou suavização), modelos de espaço de estados, séries multivariadas, estudos longitudinais, processos de longa dependência, modelos auto-regressivos (AR, do inglês, *Auto Regressive*), modelos de média móvel (MA, do inglês, *Moving Average*), modelos combinados ARMA (*Auto Regressive Moving Average*) e ARIMA (*Auto Regressive Integrated Moving Average*), modelos auto-regressivos com heterocedasticidade condicional ARCH (*Auto Regressive Conditional Heteroskedasticity*) e GARCH (*Generalized Auto Regressive Conditional Heteroskedasticity*), entre outros. Dentre estes modelos, destaque para os modelos probabilísticos, em especial os modelos ARIMA com suas etapas, detalhados na subseção a seguir.

2.1.1.1. Modelos Lineares Probabilísticos: ARIMA

Matematicamente um modelo probabilístico, também denominado processo estocástico, pode ser definido como uma coleção de variáveis aleatórias ordenadas no tempo e definidas em um conjunto de pontos n , que pode ser contínuo ou discreto. Nesta dissertação são analisados somente dados discretos.

Diferentemente da maioria dos problemas estatísticos, na análise de séries temporais, embora seja possível variar o tamanho da série observada, usualmente é impossível fazer mais do que uma observação em cada tempo. Assim, tem-se apenas uma realização do processo estocástico e uma única observação da variável aleatória no tempo n denotada $y(n)$.

Uma série temporal é qualquer conjunto de observações ordenadas no tempo e é composta por quatro elementos (MORETTIN, 1987):

- *Tendência*: verifica o sentido de deslocamento da série ao longo de vários anos;
- *Ciclo*: movimento ondulatório que ao longo de vários anos tende a ser periódico;
- *Sazonalidade*: movimento ondulatório de curta duração, em geral, inferior a um ano associada, em muitos casos, a mudanças climáticas;
- *Ruído aleatório (ou erro)*: compreende a variabilidade intrínseca aos dados e não pode ser modelado.

Um dos principais modelos para previsão de séries temporais é o modelo ARIMA, também conhecido como modelo Box-Jenkins (BOX e JENKINS, 1976). A aplicação de tal

modelo é baseada em um ciclo iterativo, no qual a escolha do modelo é realizada com base nos próprios dados. Todo este processo ocorre a partir das seguintes etapas (WERNER, 2003):

1. *Identificação*: consiste em descobrir qual dentre as várias versões dos modelos de Box-Jenkins descreve o comportamento da série;
2. *Estimação*: consiste em estimar os parâmetros ϕ e Φ do componente auto-regressivo, os parâmetros θ e Θ do componente de médias móveis e a variância de ε_t ;
3. *Verificação*: consiste em avaliar se o modelo estimado é adequado para descrever o comportamento dos dados. Caso o modelo não seja adequado, o ciclo é repetido, voltando-se à fase de identificação. Um procedimento utilizado é identificar não só um único modelo, mas alguns modelos que são, então, estimados e verificados;
4. *Previsão*: Quando se obtém um modelo satisfatório, passa-se para a última etapa da metodologia de Box-Jenkins, que constitui o objetivo principal da metodologia: realizar previsões.

Os testes utilizados neste trabalho são o teste de raízes unitárias de Dickey-Fuller Aumentado, em inglês *Augmented Dickey-Fuller* (ADF), para análise de estacionariedade e o teste da Função de Auto-Correlação, em inglês *Autocorrelation Function* (ACF), para análise de resíduos. Detalhes referentes aos testes abordados podem ser encontrados em MAKRIDAKIS (1998).

2.1.2. Modelos Não-Lineares

Apesar de que muitos modelos estudados dizem respeito ao comportamento linear das variáveis em estudo, não há razão para crer que os processos na natureza apresentem na sua maioria um comportamento linear (CHEN, 1991).

As razões, pelas quais os estudos prendem-se mais à modelagem linear, provavelmente estão relacionadas à facilidade computacional e a formulação matemática mais simples, o que não ocorre para a modelagem de fenômenos não-lineares, observados na natureza (MARTIN, 2005).

Uma representação geral de um modelo não-linear é dada por (2.4):

$$y = \varphi(x, \delta) + \varepsilon, \quad (2.4)$$

onde x e δ são variáveis explicativas, ε é o componente de distúrbio estocástico e φ é uma função que associa variáveis dependentes à variável y (SANTOS, 2005).

Considerar-se-á vários tipos de modelagens não-lineares, conforme mencionado na seção 2.2. Dentre estes modelos, destaque para os modelos não-lineares gerais, em especial os modelos de RN, que são introduzidos na seção a seguir e apresentados mais detalhadamente na seção 3.1.

2.1.2.1. Modelos Não-Lineares Gerais: RN

Um modelo não-linear geral é usualmente utilizado para se reproduzir fenômenos não-lineares. O termo “geral” refere-se ao fato de não se necessitar um prévio conhecimento detalhado sobre o sistema. Um dos modelos não-lineares gerais utilizados são as redes neurais artificiais, um tipo de modelagem inspirada no comportamento biológico dos neurônios do cérebro humano.

Por meio de uma representação pictórica, a Figura 2.3 descreve como selecionar e as motivações para uso de cada modelo de representação, análise e modelagem de fenômenos. Os modelos baseados na Física são os mais intuitivos e podem ser determinados pelas leis e fenômenos físicos. Contudo, nem sempre é simples, ou até mesmo possível, se obter todo o conhecimento detalhado requerido sobre o fenômeno para sua modelagem.

Esta dificuldade contribui para a escolha dos modelos lineares gerais. Todavia, conforme explicado na seção 2.2.2, uma parte considerável dos fenômenos observados na natureza apresenta comportamento não-linear. A impossibilidade de reprodução de tais fenômenos por modelos lineares implica na utilização da última classe descrita na Figura 2.3, a dos modelos não-lineares gerais.

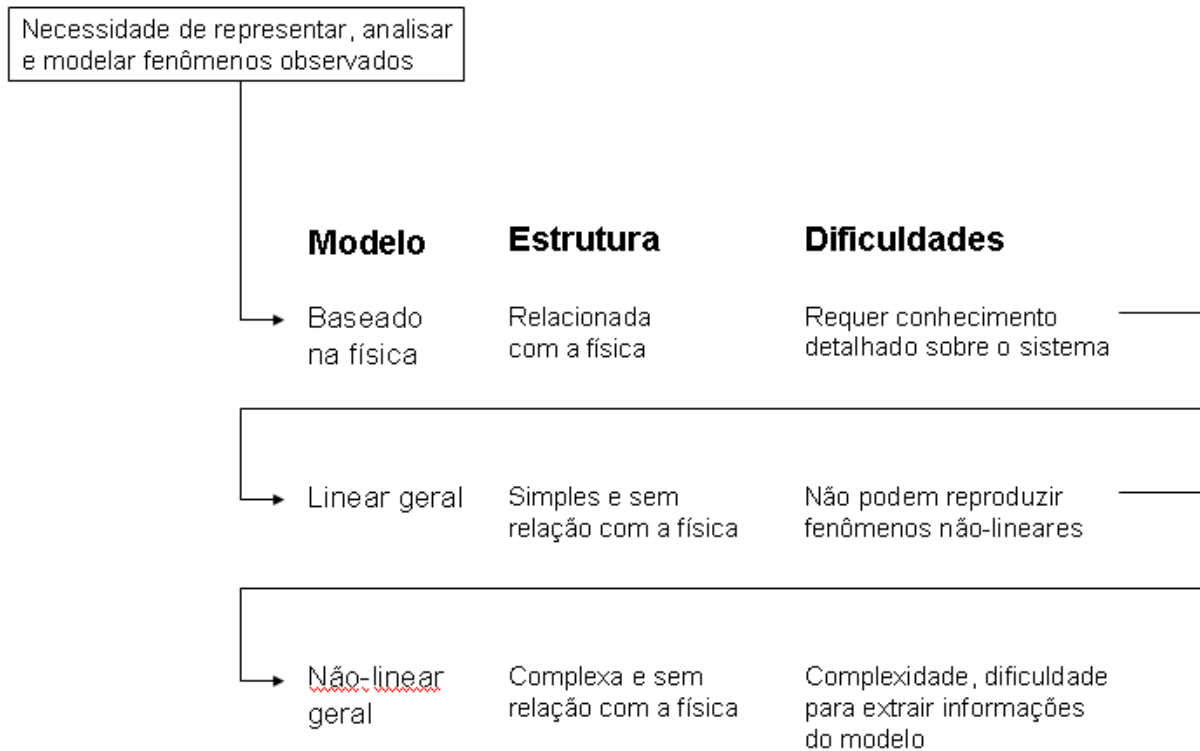


Figura 2.3: Modelos baseados na física, lineares gerais e não-lineares gerais.

Fonte: Adaptado de AGUIRRE (2000).

As RNs destacam-se como modelos não-lineares gerais com capacidade de aprender padrões complexos presentes em conjuntos de dados de determinados fenômenos (por exemplo, séries temporais) e, por esta razão, trazem uma grande contribuição ao estudo de previsão de séries temporais (HAYKIN, 2001a).

2.1.3. Avaliação de Desempenho dos Modelos

Para avaliação da qualidade da previsão é adotado o erro médio quadrático, em inglês *Mean Square Error (MSE)*, e o coeficiente de determinação múltipla, também denominado de R^2 , dois critérios considerados referências pela comunidade de pesquisadores da área de previsão de séries temporais, tal que:

$$MSE = \frac{\sum_{n=1}^N [y(n) - \hat{y}(n)]^2}{N}, \quad (2.5)$$

onde $y(n)$ é a saída real do sistema, $\hat{y}(n)$ é a saída estimada e N é o número de amostras medidas. O R^2 é dado por:

$$R^2 = 1 - \frac{\sum_{n=1}^N [y(n) - \hat{y}(n)]^2}{\sum_{n=1}^N [y(n) - \bar{y}(n)]^2}, \quad (2.6)$$

onde $y(n)$ é a saída real do sistema, $\hat{y}(n)$ é a saída estimada e $\bar{y}(n)$ é a média das N amostras medidas (média da saída real).

Os valores de MSE e R^2 são sempre números reais, sendo que um MSE igual a 0 (zero) ou um R^2 igual a 1 (um) indicam uma exata adequação do modelo para os dados medidos do sistema, mas valores de MSE menores que 0,01 e de R^2 maiores que 0,9 podem ser considerados suficientes para a maioria das aplicações práticas em previsão de séries temporais (NAGELKERKE, 1991).

O escopo deste trabalho limita-se a avaliar os valores do MSE e do R^2 .

3. Fundamentação Teórico-Empírica

3.1. Redes Neurais

Segundo CASTRO (2007), as Redes Neurais Artificiais (RNs) – em inglês *Artificial Neural Networks* – são uma área da Computação Natural (*Natural Computation*) que trata da solução de problemas a partir de uma mimetização simplificada do cérebro humano em sistemas computacionais. O trabalho pioneiro nesta área é atribuído a MCCULLOCH e PITTS (1943), que introduziram o primeiro modelo lógico-matemático de um neurônio.

A Computação Natural e suas divisões são apresentadas na Figura 3.1:

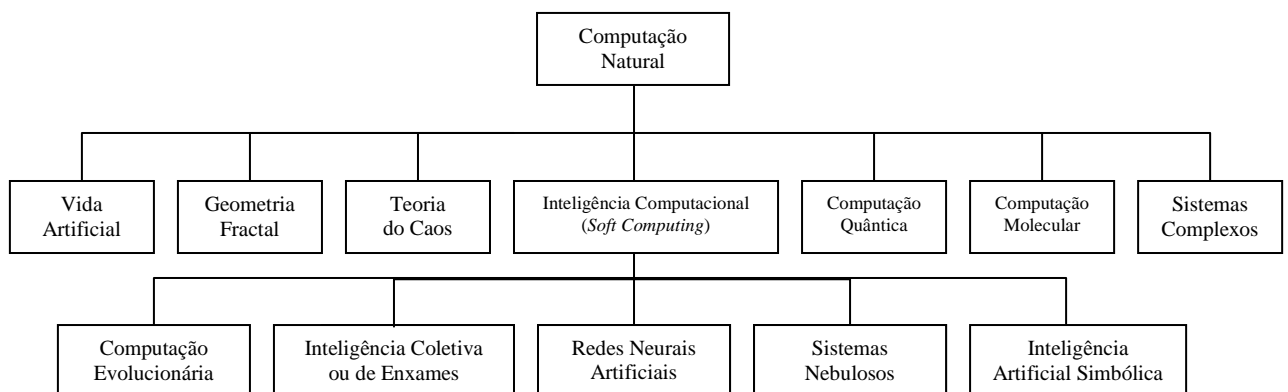


Figura 3.1: Taxonomia da computação natural.

As RNs são métodos computacionais que possuem a capacidade de se adaptar e de aprender a realizar uma certa tarefa, ou comportamento, a partir de um conjunto de exemplos dados.

Na sua forma mais geral, uma RN é uma estrutura computacional projetada para modelar a maneira como o cérebro humano realiza uma tarefa particular ou função de interesse; a rede é normalmente implementada utilizando-se componentes eletrônicos ou é simulada por programação em um computador digital (HAYKIN, 2001a). A RN pode ser caracterizada como uma ferramenta alternativa de computação que procura imitar o funcionamento do cérebro humano na solução de problemas de reconhecimento de padrões.

As RNs são modelos de aproximação de funções que aprendem a relação entre variáveis independentes e dependentes, à semelhança da regressão ou outras abordagens tradicionais.

A principal diferença entre redes neurais e as abordagens estatísticas, tais como os modelos ARIMA, é que as redes neurais não estabelecem hipóteses ou suposições sobre a distribuição ou propriedades dos dados e, desta forma, tendem a ser mais úteis em situações práticas. As RNs constituem também uma abordagem inerentemente não-linear, fornecendo precisão quando modelam dados de padrões complexos. Existem vários tipos de RNs, cada uma com diferentes objetivos, topologias e algoritmos de aprendizagem (SMITH, 2000).

Uma rede neural é formada pela interconexão de unidades de processamento denominadas neurônios, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Assim, elas devem exibir características básicas similares ao comportamento humano, tais como:

- **Aprendizado:** a RN aprende por experiência, a partir de treinamento baseado na apresentação de exemplos;
- **Associação:** a RN é capaz de fazer associações entre padrões diferentes;
- **Generalização:** a RN é capaz de generalizar por exemplos anteriores, ou seja, responder corretamente a uma entrada nunca vista antes por similaridade aos padrões já apresentados.

O aprendizado ou treinamento tem como finalidade efetuar os ajustes necessários nos parâmetros da RN, representados pelos pesos sinápticos. O objetivo desta etapa é capacitar a RN a associar um dado de entrada com um grau de semelhança aos exemplos já apresentados.

De uma forma geral, a teoria de RN pode ser dividida em três tópicos básicos:

- **Neurônio:** elemento básico, inspirado no neurônio biológico e constituído de três partes fundamentais: pesos sinápticos, que ponderam os valores das entradas do neurônio, regra de propagação, que define como as entradas são combinadas no neurônio - em geral é um somatório ponderado - e a função de ativação, que determina o efeito que o resultado da regra de propagação tem sobre o nível de ativação do neurônio;
- **Topologia:** modo de conexão entre os diferentes neurônios que formam a RN. O encadeamento de vários destes neurônios formam as camadas. Em geral, uma RN possui três ou mais camadas de neurônios, sendo que a primeira recebe o nome de camada de entrada, a última é a camada de saída e a(s) camada(s) intermediária(s) de

camada(s) escondida(s) ou oculta(s). Uma RN com a topologia (2-4-4-1) indica que ela possui a camada de entrada com 2 elementos de processamento, duas camadas intermediárias (ocultas), cada qual com 4 elementos de processamento e a camada de saída com apenas 1 elemento de processamento. O fluxo dos dados conforme o algoritmo *back-propagation* ocorre em dois sentidos: para frente e para trás;

- *Aprendizado*: processo pelo qual a RN aprende por meio dos padrões a ela apresentados. Os procedimentos de aprendizado (ou treinamento) que levam as RNs a aprender determinadas tarefas podem ser classificados em duas categorias: treinamento supervisionado, na qual a RN é alimentada por dados de entrada e seus respectivos valores de saída (valores desejados), e treinamento não supervisionado, que não requer valores de saída, na qual a rede agrupa os valores de maior semelhança.

As RNs, em especial as com aprendizado supervisionado, têm a capacidade de mimetizar processos estocásticos associados a conjuntos de dados, por meio do aprendizado. Assim como na forma convencional de um filtro linear adaptativo, as RN têm capacidade de, a partir da informação de uma resposta desejada tentar aproximar um sinal alvo durante o processo de aprendizado. Esta aproximação é obtida a partir do ajuste, de forma sistemática, de um conjunto de parâmetros livres provê um mecanismo para armazenar o conteúdo de informação subjacente presente nos dados que são apresentados à rede na fase de treinamento.

Os dois principais tipos de RNs supervisionadas são as RNs-MLP (*Multilayer Perceptron*) treinadas pelo algoritmo *back-propagation* e as redes RN-RBF. Entretanto, quando se trata de aprendizado continuado, como no caso da previsão de séries temporais, as RNs-MLP se mostram menos adequadas porque o custo computacional de treino de uma RN-MLP é geralmente superior ao de uma RN-RBF (HAYKIN, 2001a; HAYKIN, 1996).

Já as RNs-RBF, estudadas neste trabalho e detalhadas a seguir, possuem características especiais que as capacitam a aprender rapidamente padrões complexos e tendências presentes nos dados, além de se adaptar rapidamente a eventuais mudanças nos padrões destes dados.

3.1.1. Redes Neurais RBF (Função de Base Radial)

As RNs-RBF são uma abordagem de redes neurais supervisionadas com origem na área de Análise Numérica com o propósito de aproximação de funções (HARDY, 1971). Foram originalmente desenvolvidas para interpolação de dados em espaços multi-

dimensionais, e devido ao seu sucesso neste tipo de operação, podem ser consideradas aproximadores universais de funções.

O processo de aprendizagem deste tipo de rede funciona como a resolução de um problema de ajuste (aproximação) de curvas – em inglês *curve fitting*. Basicamente, o aprendizado acontece ao se tentar ajustar sucessivamente uma superfície não-linear que melhor se aproxime ao conjunto de dados de treino, em um espaço multi-dimensional, de acordo com algum critério estatístico de “melhor ajuste” pré-estabelecido.

Ao final deste processo, um combinador linear na camada de saída atua para que as ativações das funções de base radial gerem uma saída linear, aos moldes dos dados de entrada. A generalização ocorre durante a interpolação de pontos não pertencentes ao conjunto de dados de treino, mas que estejam em sua vizinhança em uma superfície multi-dimensional.

Os neurônios da camada escondida de uma rede neural do tipo RBF são um conjunto de funções que constitui uma base arbitrária no espaço por eles formado, em cujo espaço busca-se obter a expansão dos vetores de entrada. Os dados representados por meio de redes neurais RBF são, portanto, expandidos com referência a um conjunto finito de funções de ativação, denominadas funções de base radial – das quais a RN-RBF deriva seu nome (HAYKIN, 2001a).

A consequência imediata do uso de funções de ativação de base radial está na forma como as entradas são processadas pelos neurônios da camada intermediária. Ao invés da ativação interna de cada neurônio da camada intermediária se dar pelo emprego do produto escalar entre o vetor de entradas e o vetor de pesos, como no caso do *perceptron*, ela é obtida a partir de uma norma ponderada da diferença entre ambos os vetores.

Cada uma destas funções é centrada em uma coordenada específica do espaço multi-dimensional dos pontos que compõem o espaço de entrada. Cada coordenada particular caracteriza-se por definir o centro de uma – entre várias possíveis – região de maior aglomeração de dados, ou *cluster*, do espaço de dados de entrada (BISHOP, 1997).

Uma função de ativação de base radial é caracterizada por apresentar uma resposta que decresce (ou cresce) monotonicamente com distância a um ponto central. O centro e a taxa de decrescimento (ou crescimento) em cada direção são alguns dos parâmetros a serem definidos. Estes parâmetros devem ser constantes caso o modelo de regressão seja tomado como linear nos parâmetros.

Uma função de base radial monotonicamente decrescente típica é a função Gaussiana. Neste caso, os elementos do vetor de entrada são responsáveis pela taxa de decrescimento da Gaussiana junto a cada coordenada do espaço de entrada, e o argumento da função

exponencial é uma norma ponderada da diferença entre o vetor de entrada e o centro da função de base radial.

Contudo, a função adotada nas RNs-RBF estudadas neste trabalho é a multi-quadrática inversa (Equação 3.1), onde ζ corresponde as entradas da RN-RBF e σ define a distância Euclidiana média (raio médio) que mede o espalhamento dos dados representados pela função de base radial $\phi(\zeta)$ em torno de seu centro, tal que:

$$\phi(\zeta) = \frac{1}{\sqrt{\zeta^2 + \sigma^2}}, \quad (3.1)$$

Os raios de cada uma das funções de base radial de uma mesma rede RBF podem assumir diferentes valores, no entanto, para redes RBF reais (contínuas), o mesmo raio utilizado para cada neurônio não-linear já permite que a rede uniformemente aproxime qualquer função contínua, desde que haja número suficiente de funções de base radial. Na prática, o valor do raio das funções de base radial afeta as propriedades numéricas dos algoritmos de aprendizado, mas não afeta a capacidade geral de aproximação das RNs-RBF (MULGREW, 1996).

As RNs-RBF foram interpretadas por BROOMHEAD (1998) como sendo estimadores de mínimos quadrados (*Least Squares Estimator*) (MULGREW, 1996; BISHOP, 1997). Tal interpretação resultou em uma significativa redução de custo computacional e no aumento da capacidade de generalização das RNs-RBF, o que possibilita a sua aplicação a uma vasta gama de problemas em processamento digital de sinais, em especial, a Previsão de Séries Temporais.

3.1.2. Arquitetura da RN-RBF

A arquitetura de uma RN-RBF é composta por três camadas (VAN GORP, 2000), sendo:

- Uma camada de entrada (nós sensores) responsável somente pela aquisição dos sinais de entrada $\underline{u}(n) \in \mathfrak{R}^M$ do ambiente externo;
- Uma camada escondida (ou oculta) intermediária, sempre única, que possui apenas funções de ativação de base radial, e responsável pela transformação não-linear do espaço das entradas no espaço da camada escondida – geralmente de alta

dimensionalidade (quanto maior a dimensão do espaço escondido melhor a aproximação); e

- Uma camada de saída responsável pela produção dos resultados, composta pelas combinações lineares das ativações das funções de base radial (saída esta sempre linear, portanto). A saída y é, em outras palavras, a soma das saídas de cada uma das Gaussianas ponderadas pelos respectivos pesos sinápticos w_k , de tal forma que a combinação linear é expressa por:

$$\hat{y}(n) = \sum_{k=0}^{K-1} w_k \varphi_k(\underline{u}, \underline{t}_k, \sigma_k^2). \quad (3.2)$$

Na Equação 3.2, o termo $\varphi_k(\underline{u}, \underline{t}_k, \sigma_k^2)$ é a k -ésima função de base radial. Note que φ_k computa o quadrado da distância Euclidiana $D_k^2 = \|\underline{u} - \underline{t}_k\|^2$ entre um vetor de entrada \underline{u} e o centro \underline{t}_k da k -ésima função de base radial. O sinal de saída produzido pelo k -ésimo neurônio escondido é, portanto, devido à função $\exp(\cdot)$ e ao operador $(\cdot)^2$, uma função não-linear da distância D_k . O fator de escala w_k representa o peso do caminho que conecta o k -ésimo neurônio escondido ao nó de saída da rede. À Equação 3.2 pode, em alguns casos, ser ainda acrescentado um termo constante de polarização ou *bias*.

A transformação não-linear acima referida é definida pelo conjunto de funções de base radial φ_k e a transformação linear é definida pelo conjunto de pesos w_k , $k = 0, 1, \dots, K - 1$. Conforme mostrado na Figura 3.2, o mapeamento não-linear é expresso por funções de ativação Gaussianas, da forma

$$\varphi_k(n) = \varphi_k(\underline{u}(n), \underline{t}_k(n), \sigma_k^2(n)) = \exp\left[-\frac{1}{\sigma_k^2(n)} \left\| \underline{u}(n) - \underline{t}_k(n) \right\|^2\right] \quad (3.3)$$

onde $\underline{u}(n) \in \mathfrak{R}^M$ representa o vetor de entrada u no instante n , $\underline{t}_k(n) \in \mathfrak{R}^M$ representa o vetor centro da k -ésima função de base radial $k = 0, 1, \dots, K - 1$, K é o número de funções de base radial, e $\sigma_k^2(n) \in \mathfrak{R}$ é a variância associada a cada uma das funções no instante n .

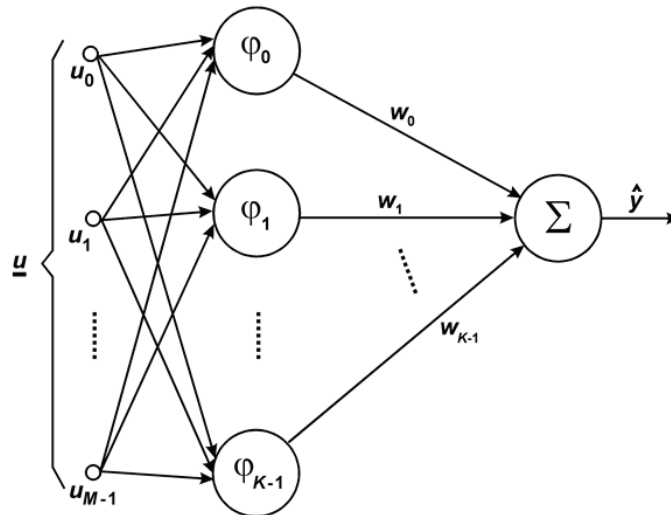


Figura 3.2: Arquitetura de uma rede neural RBF.

O procedimento para a implementação de uma rede RBF compreende a determinação, por meio de um processo de aprendizagem, dos valores adequados aos parâmetros livres da RBF, que são as variâncias σ_k^2 , os centros \underline{t}_k e os pesos sinápticos w_k . O aprendizado consiste em determinar estes parâmetros de tal forma que, dado um conjunto de estímulos \underline{u} na entrada, as saídas y se aproximem o máximo possível do conjunto de valores desejado.

3.1.3. Previsão de Séries Temporais por meio de RN-RBF

O procedimento de análise da previsão de séries temporais por meio de RN-RBF é dividido nas seguintes etapas (SOUZA, 2007):

1. *Identificação*: consiste em escolher a estrutura da RN-RBF para representar a série a ser predita, incluindo o método de treinamento e otimização de parâmetros da RN;
2. *Estimação*: consiste em estimar os parâmetros do modelo matemático;
3. *Verificação*: consiste na validação do modelo matemático com valores diferentes dos utilizados na etapa de estimação, avaliando se o modelo é adequado ou não para descrever o comportamento da série temporal;
4. *Previsão*: Quando se obtém um modelo satisfatório, o último passo é a realização de previsões.

3.2. Filtro de Kalman

Em 1960, Rudolf Emil Kalman publicou o seu famoso artigo descrevendo uma solução recursiva para o problema da filtragem linear de dados discretos (KALMAN, 1960). O Filtro de Kalman é um estimador para aquilo que é denominado “problema Gaussiano-linear-quadrático” (PEREIRA, 2000), que é o problema da estimação dos estados instantâneos de um sistema linear dinâmico perturbado por ruído Gaussiano branco (média zero e variância σ) usando-se medições linearmente relacionadas aos estados e também corrompidas por ruído branco (GREWAL, 1993).

O Filtro de Kalman é representado por um conjunto de equações matemáticas que provê uma solução computacional eficiente para o método dos mínimos quadrados (WELCH, 1998). Tal filtro permite a estimação dos estados passados, presentes e futuros de um sistema sendo que para isso não é necessário um conhecimento preciso de seu modelo. Neste caso, o termo “filtro” significa a separação entre “sinal” e “ruído”.

O Filtro de Kalman é um algoritmo linear, recursivo, não-polarizado e de variância mínima para a estimação dos estados desconhecidos de um sistema dinâmico a partir de medições discretas corrompidas por ruído. Estados, neste contexto, se referem a qualquer quantidade de interesse envolvida no processo dinâmico (PEREIRA, 2000).

3.2.1. Filtro de Kalman Discreto

O Filtro pressupõe que o sistema em questão pode ser descrito por uma equação de diferenças linear da forma:

$$x(n) = A(n-1)x(n-1) + B(n-1)u(n-1) + w(n-1), \quad (3.4)$$

onde $x \in \mathfrak{R}^k$ é o vetor de estados; $u \in \mathfrak{R}^l$ é o vetor das entradas de controle; a matriz $k \times k$, A , é a matriz de transição de estado; B , $k \times l$, é a matriz de transição de entrada e $w \in \mathfrak{R}^n$ representa o ruído do processo. Os índices n e $n-1$ representam os instantes de tempo atual e anterior respectivamente. Assume-se também que as medições são linearmente relacionadas aos estados por uma equação da forma:

$$y(n) = C(n)x(n) + v(n), \quad (3.5)$$

onde $y \in \mathfrak{R}^m$ é o vetor de medições; a matriz C , $m \times k$ é a matriz de observação e $v \in \mathfrak{R}^m$ é o vetor de erros de medição (CHEN, 1984). Considera-se também, que w e v são variáveis aleatórias Gaussianas não correlacionadas de média zero:

$$E[w(n)] = E[v(n)] = 0, \quad \forall n \quad (3.6)$$

com covariância:

$$E[w(n) w^T(i)] = \delta_{ni} Q(n), \quad (3.7)$$

$$E[v(n) v^T(i)] = \delta_{ni} R(n), \quad (3.8)$$

onde Q e R são as matrizes de covariância dos ruídos de processo e medição, respectivamente.

A variável w é usada para descrever tanto os erros ocorridos no processo quanto as incertezas no modelo de estados e v representa os erros de medição e os erros no modelo de observação.

O objetivo do modelo é encontrar uma estimação para o vetor $x(n)$, representada por $\hat{x}(n)$, que minimiza o erro médio quadrático, tal que:

$$E\{[x(n) - \hat{x}(n)]^T [x(n) - \hat{x}(n)]\}. \quad (3.9)$$

Definindo $e(n) \triangleq x(n) - \hat{x}(n)$ como sendo o vetor dos resíduos entre os estados reais e os estimados vê-se que a função custo a ser minimizada é exatamente o somatório quadrático dos resíduos para cada instante de amostragem, tal que:

$$J = \sum_{i=1}^n e(i) = e^T e = \|e\|^2. \quad (3.10)$$

Nota-se que ao minimizar a função custo, minimiza-se também a covariância do resíduo:

$$P(n) = E[e(n)e^T(n)]. \quad (3.11)$$

Um vetor de estados estimados *a priori* no instante n pode ser definido como $\hat{x}^-(n) \in \mathfrak{R}^k$. Desta forma, tem-se:

$$\hat{x}^-(n) = A(n-1) \hat{x}^-(n-1) + B(n-1) u(n-1), \quad (3.12)$$

e de forma análoga a $\hat{x}^-(n)$:

$$e^-(n) \triangleq x(n) - \hat{x}^-(n) \quad (3.13)$$

e

$$P^-(n) = E[e^-(n) e^{-T}(n)]. \quad (3.14)$$

Na derivação das equações do Filtro de Kalman deseja-se encontrar uma equação que calcule o estado $\hat{x}(n)$ a partir de uma combinação linear dos estados estimados, *a priori*, $\hat{x}^-(n)$ e do vetor de medições $y(n)$:

$$\hat{x}(n) = K^l(n) \hat{x}^-(n) + K(n)y(n). \quad (3.15)$$

A partir da condição de ortogonalidade entre os resíduos e os estados imposta pelo método dos mínimos quadrados observa-se que:

$$K^l(n) = I - K(n)C(n). \quad (3.16)$$

Desta forma:

$$\hat{x}(n) = \hat{x}^-(n) + K(n) \cdot [y(n) - C(n) \hat{x}^-(n)], \quad (3.17)$$

onde a matriz $m \times k$, $K(n)$, deve ser escolhida para ser a matriz de ganho que minimiza a covariância do resíduo. O termo que multiplica $K(n)$ é denominado inovação.

Para encontrar o ganho $K(n)$ deve ser realizada a substituição para o resíduo $e(n)$. Derivando esta nova equação em relação a $K(n)$, igualando o resultado a zero e isolando $K(n)$ encontra-se que:

$$K(n) = P^-(n)C^T(n) \cdot [C(n)P^-(n)C^T(n) + R(n)]^{-1}. \quad (3.18)$$

A Equação 3.18 é a principal equação do Filtro de Kalman. O único termo ainda não determinado é $P^-(n)$ ($R(n)$ deve ser estimado). Subtraindo $x(n)$ e levando em consideração as definições de resíduo tem-se:

$$e^-(n) = A(n-1)e(n-1) - w(n-1), \quad (3.19)$$

pós-multiplicando por e^{-T} e tomando o valor esperado:

$$P^-(n) = A(n-1)P(n-1)A^T(n-1) + Q(n-1). \quad (3.20)$$

A matriz $P(n)$ pode ser obtida a partir da Equação 3.20, onde $x(n)$ deve ser subtraído em ambos os lados da equação. Tomando-se o valor esperado, após de alguma manipulação tem-se que:

$$P(n) = [I - K(n)C(n)]P^-(n). \quad (3.21)$$

Desta forma foram encontradas todas as equações necessárias para a implementação do filtro de Kalman discreto.

Na Figura 3.3 é mostrado o diagrama de blocos de um sistema que utiliza o Filtro de Kalman, por meio do qual pode-se perceber a relação existente entre o Filtro e o sistema cujos estados estão sendo estimados.

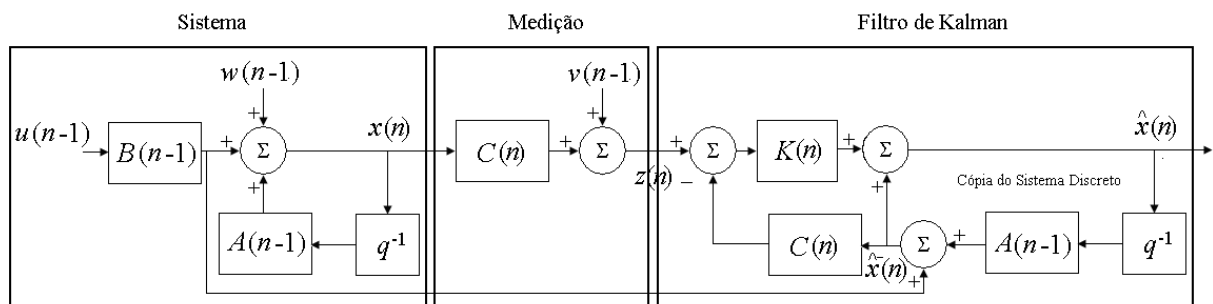


Figura 3.3: Diagrama em blocos do sistema, modelo de medição e filtro de Kalman, onde q^{-1} é o operador atraso.

A partir das equações apresentadas, pode-se resumir o algoritmo do Filtro de Kalman pelas seguintes etapas:

i. Escolher os valores iniciais $\hat{x}(0)$ e $P(0)$ e faça $n = 1$. Se não houver nenhum conhecimento sobre as condições iniciais do sistema, a covariância do erro é suficiente para favorecer a convergência;

ii. Calcular os estados estimados *a priori*, onde:

$$\hat{x}^-(n) = A(n-1) \hat{x}(n-1) + B(n-1) u(n-1);$$

iii. Obter o novo vetor de medições $y(n)$;

iv. Estimar as matrizes de covariância dos erros de processo de medição, $Q(n-1)$ e $R(n)$ respectivamente;

v. Calcular a covariância dos resíduos de estimação *a priori*:

$$P^-(n) = A(n-1)P(n-1)A^T(n-1) + Q(n-1);$$

vi. Calcular o ganho:

$$K(n) = P^-(n)C^T(n) \cdot [C(n)P^-(n)C^T(n) + R(n)]^{-1};$$

vii. Estimar o vetor de estados:

$$\hat{x}(n) = \hat{x}^-(n) + K(n) \cdot [y(n) - C(n) \hat{x}^-(n)];$$

viii. Calcular a covariância do erro:

$$P(n) = [I - K(n)C(n)] P^-(n);$$

ix. Incrementar n e voltar à etapa ii até que um critério de parada definido previamente seja atingido.

O algoritmo do Filtro de Kalman é geral e direto, e sua implementação computacional requer alguma prática e conhecimento prévio sobre o sistema. Desta forma, além de se obter e sintonizar o modelo do processo deve-se descobrir se o conjunto de sensores disponíveis é suficiente ou não para observar todos os estados de interesse.

3.2.2. Filtro de Kalman Estendido

O Filtro de Kalman estima os estados de um sistema discreto que é regido por uma equação de diferenças linear. Mas o que acontece no caso de séries temporais financeiras, por meio das quais o processo a ser estimado é não-linear (CHEN, 2003)? O Filtro de Kalman que lineariza o sistema em torno das médias e variâncias correntes é chamado de Filtro de Kalman Estendido, em inglês, *Extended Kalman Filter* (EKF). Em outras palavras, o EKF é uma versão do Filtro de Kalman Discreto aplicado a sistemas não-lineares.

É possível linearizar as funções em torno da estimação corrente usando as derivadas parciais do processo e das funções de medição para calcular as estimações mesmo em face de relações não-lineares.

Assume-se que o processo possui os estados $x \in \mathfrak{R}^k$, mas a equação que governa a dinâmica do sistema é uma equação de diferenças não-linear da forma:

$$x(n) = f(x(n-1), u(n-1), w(n-1), n-1) \quad (3.22)$$

e a função de medição é:

$$y(n) = h(x(n), v(n), n), \quad (3.23)$$

onde $f(\cdot)$ e $h(\cdot)$ são funções não-lineares. Para as variáveis w e v são válidas as observações feitas na seção de Filtro de Kalman Discreto.

Como não é possível conhecer os valores individuais das variáveis w e v em cada intervalo de amostragem é aceitável aproximar o vetor de estados e o vetor de medições respectivamente por:

$$\tilde{x}(n) = f(\hat{x}(n-1), u(n-1), 0, n-1), \quad (3.24)$$

$$\tilde{y}(n) = h(\tilde{x}(n), 0, n), \quad (3.25)$$

onde $\hat{x}(n-1)$ é o vetor de estados estimados no passo anterior. Para estimar os estados do processo as Equações 3.24 e 3.25 devem ser linearizadas em torno de \tilde{x} e \tilde{y} . Esta linearização pode ser representada por:

$$x(n) \approx \tilde{x}(n) + A(n-1) \cdot [x(n-1) - \hat{x}(n-1)] + W(n-1)w(n-1) \quad (3.26)$$

$$y(n) \approx \tilde{y}(n) + C(n) \cdot [x(n) - \tilde{x}(n)] + V(n)v(n), \quad (3.27)$$

onde:

$A(n-1)$ é a matriz Jacobiana das derivadas parciais de $f(\cdot)$ em relação a x , avaliada em torno de $\hat{x}(n)$:

$$A(n-1) \approx \left. \frac{\partial f(x, u, w, n)}{\partial x} \right|_{\tilde{x}(n), u(n-1), 0, n-1}. \quad (3.28)$$

$W(n-1)$ é a matriz Jacobiana das derivadas parciais de $f(\cdot)$ em relação a w , avaliada em torno de $\hat{x}(n)$:

$$W(n-1) \approx \left. \frac{\partial f(x, u, w, n)}{\partial w} \right|_{\tilde{x}(n), u(n-1), 0, n-1}. \quad (3.29)$$

$C(n)$ é a matriz Jacobiana das derivadas parciais de $h(\cdot)$ em relação a x , avaliada em torno de $\tilde{x}(n)$:

$$C(n) \approx \left. \frac{\partial h(x, v, n)}{\partial x} \right|_{\tilde{x}(n), 0, n} . \quad (3.30)$$

$V(n)$ é a matriz Jacobiana das derivadas parciais de $h(\cdot)$ em relação a v , avaliada em torno de $\tilde{x}(n)$:

$$V(n) \approx \left. \frac{\partial h(x, v, n)}{\partial v} \right|_{\tilde{x}(n), 0, n} . \quad (3.31)$$

Definindo o resíduo de predição e medição como sendo, respectivamente:

$$\tilde{e}_x(n) \triangleq x(n) - \tilde{x}(n), \quad (3.32)$$

$$\tilde{e}_y(n) \triangleq y(n) - \tilde{y}(n) . \quad (3.33)$$

As Equações 3.32 e 3.33, respectivamente, podem ser reescritas como:

$$\tilde{e}_x(n) \approx A(n-1) \cdot [x(n-1) - \hat{x}(n-1)] + \xi(n-1) , \quad (3.34)$$

$$\tilde{e}_y(n) \approx C(n)\tilde{e}_x(n) + \eta(n) , \quad (3.35)$$

onde ξ e η representam novas variáveis aleatórias independentes de média zero e matrizes de covariância WQWT e VQVT, respectivamente.

As Equações 3.34 e 3.35 são lineares, o que motiva o uso de $\tilde{e}_y(n)$ e de um Filtro de Kalman Discreto para estimar o resíduo de predição $\tilde{e}_x(n)$. Esta estimação, chamada de $\tilde{e}_y(n)$, pode ser usada para obter uma estimativa dos estados do processo não-linear original, tal que:

$$\hat{x}(n) = \tilde{x}(n) - \hat{e}(n) . \quad (3.36)$$

A equação do filtro de Kalman discreto usado para estimar $\hat{e}(n)$ é:

$$\hat{e}(n) = K(n)\tilde{e}_y(n) . \quad (3.37)$$

Substituindo as equações como se deseja tem-se que a equação para o Filtro de Kalman Estendido é:

$$\hat{x}(n) = \tilde{x}(n) + K(n)\tilde{e}_y(n) \quad (3.38)$$

$$= \tilde{x}(k) + K(n) \cdot [y(n) - \tilde{y}(n)] , \quad (3.39)$$

onde $K(n)$ pode ser calculado pela Equação 3.21.

O conjunto completo de equações do EKF pode ser obtido como adaptação daquelas mostradas na seção sobre Filtro de Kalman Discreto. Note que nas equações do filtro estendido, $\tilde{x}(n)$ corresponde ao $\hat{x}(n)$ e conseqüentemente a matriz $P(n)$ é então denominada de $\tilde{P}(n)$ apenas para manter a consistência da nomenclatura.

O algoritmo para o EKF pode ser resumido nas seguintes etapas:

- i. Escolher os valores iniciais $\hat{x}(0)$ e $P(0)$ e faça $n = 1$;
- ii. Calcular a predição dos estados estimados:

$$\tilde{x}(n) = f(\hat{x}(n-1), u(n-1), 0, n-1);$$
- iii. Obter o novo vetor de medições $y(n)$;
- iv. Estimar as matrizes de covariância dos erros de processo e medição, $Q(n-1)$ e $R(n)$, respectivamente;
- v. Calcular as matrizes Jacobianas $A(n-1)$, $W(n-1)$, $C(n)$ e $V(n)$;
- vi. Calcular a covariância dos resíduos de estimação *a priori*:

$$\tilde{P}(n) = A(n-1)P(n-1)A^T(n-1) + W(n-1)Q(n-1)W^T(n-1);$$
- vii. Calcular o ganho:

$$K(n) = \tilde{P}(n)C^T(n).[C(n)\tilde{P}(n)C^T(n) + V(n)R(n)V^T(n)]^{-1};$$
- viii. Estimar o vetor de estados:

$$\hat{x}(n) = \tilde{x}(n) + K(n).[y(n) - h(\tilde{x}(n), 0, n)];$$
- ix. Calcular a covariância do erro:

$$P(n) = [I - K(n)C(n)]\tilde{P}(n);$$
- x. Incrementar n e voltar à etapa ii até que um critério de parada definido previamente seja atingido.

GREWAL (1993) faz vários comentários a respeito da implementação do EKF a fim de reduzir o esforço computacional que, a princípio, é grande.

3.2.2.1. Treinamento de RN-RBF por meio do EKF

Conforme explicado na seção anterior, as RNs-RBF são ferramentas eficientes em solucionar problemas de processamento de sinal, classificação de padrões, em especial a previsão de séries temporais (SIMON, 2002). Vários algoritmos têm sido propostos para a escolha dos protótipos RBF e treinamento da rede.

A seleção dos protótipos e dos pesos sinápticos da RN-RBF podem ser analisadas como um problema de identificação de sistemas. Derivações do EKF para tais fins estão amplamente disponíveis na literatura, já que a aplicação de tais filtros em sistemas não-lineares se mostrou eficiente para sistemas cuja não-linearidade é pouco significativa (KUCHNER, 1967; GORDON *et al.*, 1993).

Para sistemas não-lineares com ruído colorido, o filtro de Kalman pode ser estendido pela linearização da matriz de transição não-linear da descrição em variáveis de estado do sistema em torno dos parâmetros estimados correntes (HAYKIN, 2001a). Neste contexto, o filtro de Kalman estendido atualiza os parâmetros de forma a ser consistente com todos os dados medidos do processo e geralmente converge em poucas iterações (SIMON, 2002; WELCH, 1995). Deste modo, este trabalho utiliza o EKF para otimizar o procedimento de aprendizagem da RN-RBF, minimizando seu erro de treinamento.

A formulação dos filtros de Kalman está fundamentada na representação em espaço de estados de um sistema dinâmico linear discreto de dimensão finita, ou seja, o filtro de Kalman é descrito em termos de equações a diferenças, porém se o sistema for não-linear, pode-se estender o uso do filtro de Kalman com um procedimento de linearização de um sistema da forma

$$\theta_{n+1} = f(\theta_n) + \omega_n, \quad (3.40)$$

$$y_n = h(\theta_n) + v_n, \quad (3.41)$$

onde o vetor θ_n é o estado do sistema no instante de tempo N , ω_n é o ruído provocado pela própria dinâmica do processo, y_n é o vetor de observação, v_n é o ruído de observação (ou de medida), e $f(\cdot)$ e $h(\cdot)$ são funções não-lineares.

A idéia básica do filtro de Kalman estendido consiste em linearizar o sistema a cada instante de tempo em torno do estado estimado recente. Se as não-linearidades são suficientemente suaves, pode-se fazê-lo aplicando a aproximação de Taylor de primeira ordem nas funções não-lineares v e $h(\cdot)$ em torno do estado estimado $\hat{\theta}$ para obter

$$f(\theta_n) = f(\hat{\theta}_n) + F_n(\theta_n - \hat{\theta}_n) + \text{termos de ordem elevada} \quad (3.42)$$

$$h(\theta_n) = h(\hat{\theta}_n) + H_n^T(\theta_n - \hat{\theta}_n) + \text{termos de ordem elevada}, \quad (3.43)$$

sendo adotada a seguinte notação para as derivadas parciais:

$$F_n = \frac{\partial f(\theta)}{\partial \theta} \Big|_{\theta = \hat{\theta}_n}, \quad (3.44)$$

$$H_n^T = \frac{\partial h(\theta)}{\partial \theta} \Big|_{\theta = \hat{\theta}_n}. \quad (3.45)$$

Desconsiderando os termos de ordem mais elevada, um sistema linear aproximado é obtido, onde

$$\theta_{n+1} = F_n \theta_n + \omega_n + \phi_n, \quad (3.46)$$

$$y_n = H_n^T \theta_n + v_n + \varphi_n. \quad (3.47)$$

Neste caso, ϕ_n e φ_n são definidos por

$$\phi_n = f(\hat{\theta}_n) - F_n \hat{\theta}_n, \quad (3.48)$$

$$\varphi_n = h(\hat{\theta}_n) - H_n^T \hat{\theta}_n. \quad (3.49)$$

A obtenção da aproximação linear do sistema é o primeiro passo para utilização do filtro de Kalman. O problema do filtro de Kalman consiste em resolver simultaneamente e de maneira ótima as equações de processo e de medida para um estado desconhecido.

Pode-se formular o problema de filtragem de um sistema linear achando um valor ótimo de $\hat{\theta}_n$, tal que o sistema deve satisfazer as restrições impostas pelos teoremas do estimador médio condicional e o do princípio da ortogonalidade. Finalmente, pode-se mostrar que a estimação desejada $\hat{\theta}_n$ pode ser obtida recursivamente pelas equações do filtro de Kalman:

$$\hat{\theta}_n = f(\hat{\theta}_{n-1}) + K_n [y_n - h(\hat{\theta}_{n-1})], \quad (3.50)$$

$$K_n = P_n H_n (R + H_n^T P_n H_n)^{-1}, \quad (3.51)$$

$$P_{n+1} = F_n (P_n - K_n H_n^T P_n) F_n^T + Q, \quad (3.52)$$

onde K_n é o ganho de Kalman e P_n é a matriz de covariância do erro de estimação do estado θ_k .

As matrizes Q e R são as matrizes de covariância dos ruídos artificiais ω_n e v_n , respectivamente. Como o sistema descrito é uma aproximação de um sistema não-linear, o filtro não é ótimo e a estimação é somente uma aproximação da média condicional.

O problema de otimização da matriz de pesos sinápticos W e as coordenadas dos vetores protótipos v_j sob o ponto de vista da estimação ótima de parâmetros com filtro de Kalman de uma RN-RBF pode ser tratado como um problema de minimização do tipo mínimo quadrados ponderado. Neste caso, para aplicar o EKF, faz-se com que os elementos da matriz de pesos W e os elementos dos vetores de protótipos v_j constituam o estado do sistema discreto não-linear em que o filtro de Kalman é aplicado.

Um dos problemas da aplicação do EKF na otimização da RN-RBF é o alto custo computacional utilizado por ele na solução de problemas. Dentre as abordagens encontradas

na literatura para diminuir este custo, SIMON (2002), pioneiro nas pesquisas de treinamento de RN-RBF com EKF, propõe um filtro de Kalman desacoplado para este fim e que melhora substancialmente o desempenho da solução.

Todavia, outra dificuldade encontrada – e menos estudada – acaba por ser a determinação dos parâmetros de sintonia do EKF, os valores de P_0 , Q e R . O presente trabalho propõe que estes parâmetros sejam otimizados por meio do uso de uma heurística de otimização ainda pouco pesquisada da área de computação evolucionária: a Evolução Diferencial, detalhada na seção a seguir.

3.3. Meta-Heurísticas

O objetivo da Otimização Combinatória (OC) é encontrar a melhor configuração de um projeto com mais eficiência e menor custo operacional. A aplicação de otimização em vários campos da ciência tem crescido consideravelmente nas últimas décadas. Cada um dos inúmeros métodos de otimização pode alcançar melhor resultado dependendo do tipo de problema a que são aplicados. A escolha do método depende de uma série de características do problema a ser otimizado, principalmente do comportamento da função que o representa (VANDERPLAATS, 1999).

Quanto à relação entre as variáveis de decisão na função objetivo e nas restrições, tais métodos podem ser classificados em métodos de programação linear e de programação não-linear. Sendo que os não-lineares podem ser divididos em métodos determinísticos e métodos heurísticos (estocásticos). Os métodos determinísticos são baseados no cálculo de derivadas ou em aproximações destas, necessitando de informações do vetor gradiente, seja procurando o ponto onde ele se anula ou usando a direção para a qual aponta. Para ARANTES (2006a), estes métodos produzem melhores resultados para funções contínuas, convexas e unimodais (funções que possuem apenas um ponto de mínimo ou de máximo).

O termo “heurística” tem origem na palavra grega *heuriskein*, que significa “encontrar”/“descobrir”. Os métodos heurísticos utilizam apenas as informações da função a ser otimizada, que pode ser de difícil representação, não-linear, descontínua, não diferenciável, multimodal (possui muitos pontos de mínimo ou de máximo). Estes métodos buscam a solução ótima a partir de regras de probabilidade operando de maneira “aleatória orientada”.

Uma heurística também pode visar a obtenção de boas soluções (soluções aproximadas), mas com um custo computacional aceitável (muitas vezes baseada em procedimentos simples e intuitivos). Neste caso, não oferece garantias de otimalidade, nem de admissibilidade, e eventualmente, não pode sequer garantir informação de qualidade a respeito de uma dada solução.

Uma meta-heurística é uma heurística para propósitos gerais. Alguns exemplos de meta-heurísticas são os “clássicos” Busca Local e Busca Tabu, e métodos inspirados nas “leis da natureza”. Esta última categoria baseia-se em disciplinas como a física – método do *Simulated Annealing* (AARTIS, 1989), que se utiliza de conceitos da termodinâmica, e Computação Quântica (HIRVENSALO, 2000), baseado em conceitos da mecânica quântica; a química, por meio da Computação Molecular (GRAMB *et al.*, 2001) e a biologia, por meio da Inteligência Coletiva (BONABEAU *et al.*, 1999) e da Computação Evolucionária (BÄCK *et al.*, 2000).

Os principais métodos de otimização baseados na biologia (meta-heurísticas bio-inspiradas) podem ser divididos conforme abaixo:

- Redes Neurais Artificiais (ROSENBLATT, 1962; HAYKIN, 2001a; HAYKIN, 2001b) (*Artificial Neural Networks*) – sistema com capacidade de aprendizado (previamente descritos na seção 3.1 deste capítulo).
- Sistemas Nebulosos (ZADEH, 1965; PEDRYCZ, 1998) (*Fuzzy Systems*) – tratam e realizam inferências com relação a informações vagas, imprecisas e/ou incertas com base na lógica multivalorada. Podem ser classificados como de inspiração tanto biológica/cognitiva (a mente humana consegue lidar com informações incertas) quanto física (algumas de suas propriedades, como o princípio da incerteza de Heizenberg e o da polivalência de Lukasiewicz são alguns dos principais fundamentos da mecânica quântica). Podem ser sub-divididos conforme abaixo:
 - Lingüístico;
 - Interpolativo;
 - Relacional.
- Inteligência Coletiva ou de Enxames (*Swarm Intelligence*) – busca comportamento coletivo inteligente a partir de sistemas descentralizados formados por uma população de indivíduos simples interagindo localmente entre si e com seu ambiente. Destacam-se nesta categoria os seguintes algoritmos de otimização:
 - Nuvem ou Enxame de Partículas (*Particle Swarm Optimization*);
 - Colônia de Formigas (BONABEAU *et al.*, 1999) (*Ant Colony Systems*);

- Sistemas Imunológicos Artificiais (DASGRUPTA, 1999) (*Artificial Immuno Systems*).
- Computação Evolutiva (*Evolutionary Computation*) – utiliza-se dos princípios da evolução natural de Darwin ao aplicar algoritmos de seleção, cruzamento e mutação em uma população, na qual cada indivíduo representa um ponto de busca no espaço das soluções candidatas de um dado problema. Uma classificação destes algoritmos pode ser resumida conforme abaixo:
 - Algoritmos Genéticos (HOLLAND, 1962) (*Genetic Algorithms*) – propõe a implementação computacional dos conceitos elementares da genética para solução de problemas de otimização;
 - Programação Evolutiva (FOGEL, 1960) (*Evolutionary Programming*) – visa basicamente prever o comportamento de máquinas de estados finitas;
 - Estratégias Evolutivas (RECHENBERG, 1973) (*Evolution Strategies*) – enfatizam a auto-adaptação. O papel da recombinação é aceito, mas como operador secundário;
 - Programação Genética (KOZA, 1992) (*Genetic Programming*) – tem atuação na evolução de programas;
 - Sistemas Classificadores (BAGLEY, 1967) (*Classifier Systems*) – são sistemas de produção utilizados em uma parte do algoritmo global; e
 - Evolução Diferencial (STORN e PRICE, 1995) (*Differential Evolution*) – abordagem que é mais detalhada a seguir.

A Figura 3.4 apresenta uma taxonomia dos métodos de otimização de problemas contínuos.

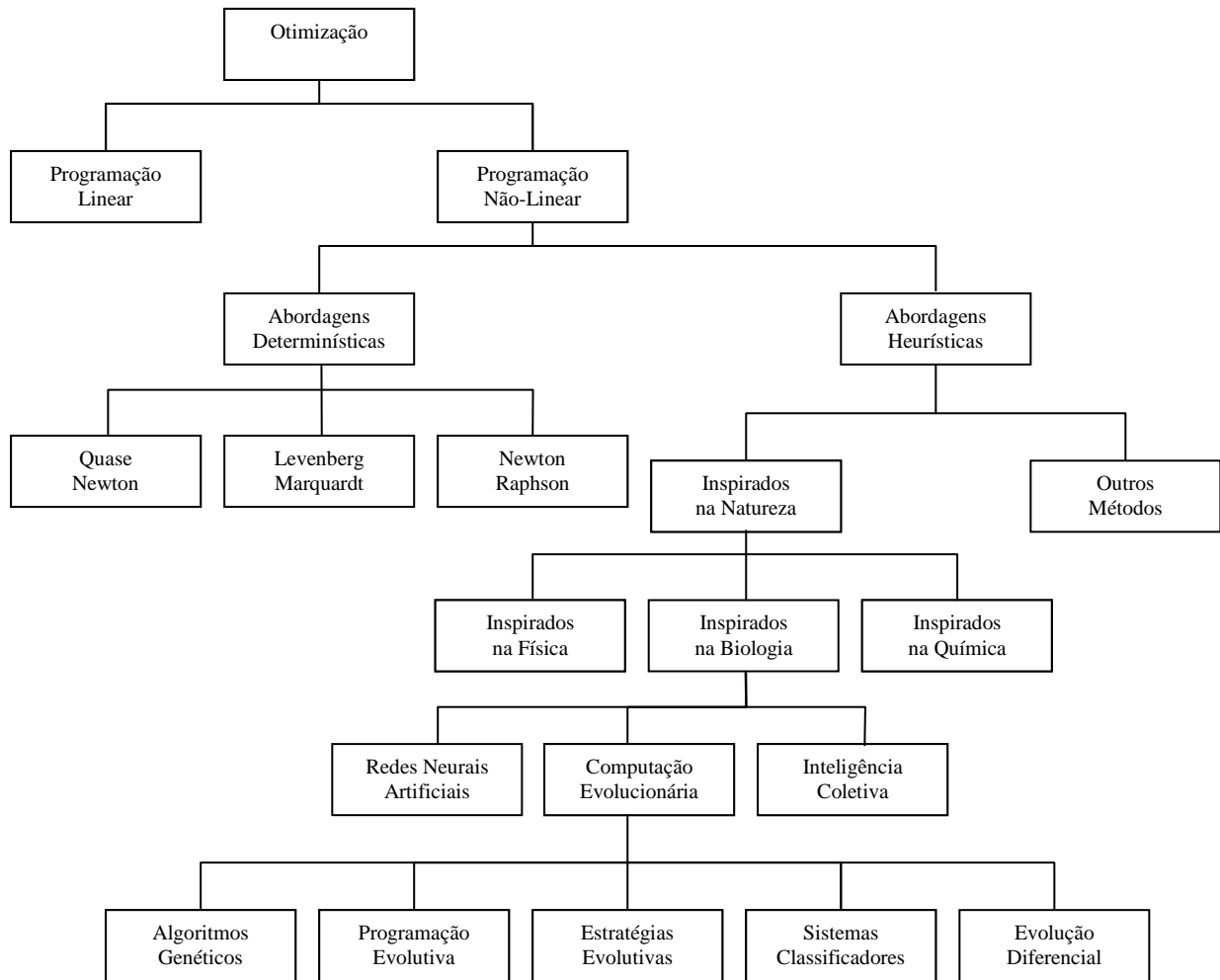


Figura 3.4: Taxonomia dos métodos meta-heurísticos de otimização.

3.3.1. Evolução Diferencial

Em 1995, na tentativa de se resolver o problema de ajuste polinomial de Chebychev, Rainer Storn e Kenneth Price descreveram uma solução adaptativa simples e eficiente para este problema e outros problemas de otimização não-linear contínua (STORN e PRICE, 1995). A solução proposta era baseada nos conceitos de computação evolutiva, sendo que sua idéia principal era gerar novos indivíduos pela adição da diferença vetorial ponderada entre dois indivíduos aleatórios da população a um terceiro indivíduo. Por este motivo, a solução foi denominada *Differential Evolution*, em português, Evolução Diferencial (ED) (ARANTES, 2006b).

A ED tem sido aplicada com sucesso na solução de diversos tipos de problemas, com destaque para problemas computacionais, como reconhecimento de imagens (DU, 2006), e

problemas de engenharia, tais como projetos ótimos de filtros biológicos de gases (BATH, 2006) e projeto ótimo de estruturas de concreto e aço (HRSTKA, 2003).

Este princípio de usar diferenças de vetores para perturbar a população (indivíduos) resulta em um método com rápida convergência, de fácil implementação, “guloso” e robusto. Sua escolha para problemas de otimização é baseada em suas características:

- é um algoritmo de busca estocástica que é motivado por mecanismos de seleção natural;
- é menos susceptível a mínimos (ou máximos) locais, pois busca a solução ótima global pela manipulação de uma população de soluções candidatas, ou seja, busca um número de diferentes áreas simultaneamente no espaço de busca;
- não requer informação de derivadas para o cálculo da função de aptidão;
- manipula diretamente os números de ponto flutuante (fenótipo), diferente dos algoritmos genéticos canônicos ou binários que manipulam cadeias de bits (genótipo), o que diminui o custo computacional necessário para transformação de uma representação genotípica em fenotípica;
- geralmente não necessita de populações grandes para funcionar eficientemente.

Na ED clássica, cada variável (indivíduo) é representada por um valor real (ponto flutuante). O algoritmo começa criando uma população inicial escolhida aleatoriamente, que deve cobrir todo o espaço de busca. Geralmente, é criada por uma distribuição de probabilidade uniforme, quando não há nenhum conhecimento sobre o problema.

A idéia principal da evolução diferencial é gerar novos indivíduos, denotados vetores modificados ou doadores, pela adição da diferença ponderada entre dois indivíduos randômicos da população a um terceiro indivíduo. Esta operação é denominada mutação.

As componentes do indivíduo doador são misturadas com as componentes de um indivíduo escolhido aleatoriamente (chamado vetor alvo), para resultar no vetor tentativa (ou vetor experimental). O processo de misturar os parâmetros é referido freqüentemente como "cruzamento" na comunidade dos algoritmos evolutivos (ARANTES, 2006).

Se o valor da função objetivo do vetor experimental for menor do que o valor da função para o vetor alvo, então o vetor experimental substitui o vetor alvo na geração seguinte. Esta última operação é chamada seleção. O procedimento é finalizado a partir de algum critério de parada.

A Evolução Diferencial é um método de busca direto paralelo que utiliza vetores de parâmetros multi-dimensionais NP :

$$x_{i,G}, i = 1, 2, \dots, NP, \quad (3.53)$$

onde G é a geração e NP é o número de indivíduos da população, sendo que NP não muda durante o processo de minimização e o vetor inicial de população é escolhido aleatoriamente e deve cobrir completamente o espaço de parâmetros.

A ED gera novos vetores de parâmetros a partir da adição de diferenças ponderadas entre dois vetores de população a um terceiro vetor. A esta operação dá-se o nome de mutação. Os parâmetros do vetor alterado são então misturados com os parâmetros de outro vetor predeterminado, o vetor alvo, para rendimento o então denominado vetor experimental.

A mistura de parâmetros é frequentemente referida como cruzamento (em inglês, *crossover*) e é explicado com detalhes na seção 3.3.1.2. Se o vetor experimental rende um valor funcional de custo menor que o vetor alvo, o vetor tentativa substitui o vetor alvo na geração seguinte. Esta última operação é chamada seleção.

A estratégia básica da ED e seus operadores são descritos de um modo mais detalhado a seguir.

3.3.1.1. Operador de Mutação

Para cada vetor alvo $x_{i,G}$; $i = 1, 2, 3, \dots, NP$, um vetor mutante é gerado conforme

$$v_{i,G+1} = x_{i,G} + F \cdot (x_{2,G} - x_{3,G}) \quad (3.54)$$

com índices aleatórios $x_{1,G}, x_{2,G}, x_{3,G} \in \{1, 2, \dots, NP\}$, inteiros, mutuamente diferentes e $F > 0$. Os inteiros escolhidos aleatoriamente $x_{1,G}, x_{2,G}, x_{3,G}$ são também escolhidos para ser diferentes do índice i executado, tal que NP deve ser maior ou igual a quatro para permitir para esta condição. F é um fator constante real $\in [0, 2]$ que controla a amplificação de uma variação diferencial $(x_{2,G} - x_{3,G})$.

A Figura 3.5 mostra um exemplo bi-dimensional que ilustra os diferentes vetores que atuam na geração de $v_{i,G+1}$.

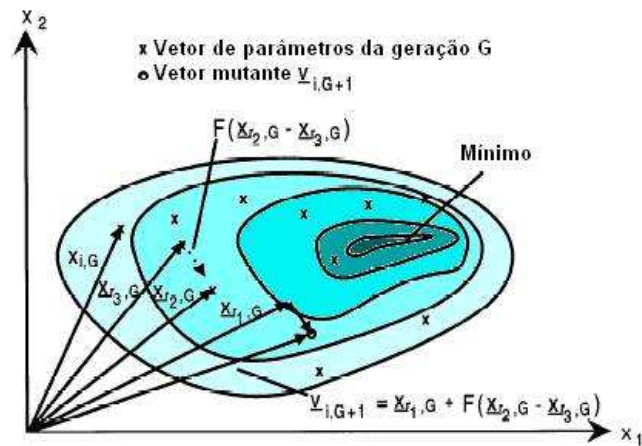


Figura 3.5: Um exemplo de uma função custo bi-dimensional mostrando suas linhas de contorno e o processo para geração de $v_{i,G+1}$.

Fonte: Adaptado de STORN e PRICE (1997)

3.3.1.2. Operador de Cruzamento

Para aumentar a diversidade dos vetores de parâmetros perturbados, é introduzido o cruzamento. Para este fim, o vetor experimental:

$$u_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, \dots, u_{Di,G+1}), \quad (3.55)$$

é formado, onde

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{se } randb(j) \leq CR \\ x_{ji,G} & \text{se } randb(j) > CR \end{cases} \quad \text{ou} \quad \begin{cases} j = rnbr(i) \\ e \quad j \neq rnbr(i) \end{cases}, \quad (3.56)$$

onde $j = 1, 2, \dots, D$.

Em (3.56), $randb(j)$ é a j -ésima avaliação de um gerador uniforme de números aleatórios com resultado $\in [0, 1]$. A probabilidade de cruzamento $CR \in [0, 1]$ é determinada pelo usuário e é escolhido aleatoriamente um índice $rnbr(i) \in 1, 2, \dots, D$ que assegura que $u_{i,G+1}$ traz ao menos um parâmetro de $v_{i,G+1}$.

A Figura 3.6 fornece um exemplo do mecanismo de cruzamento para vetores com 8 dimensões.

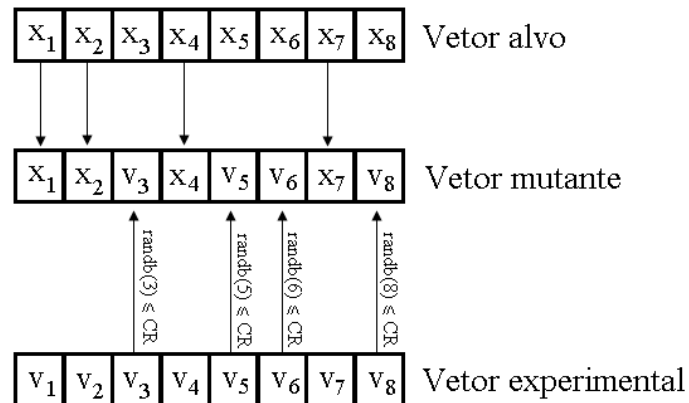


Figura 3.6: Cruzamento (*crossover*) para $D = 8$ parâmetros.

Fonte: Adaptado de STORN e PRICE (1997)

3.3.1.3. Operador de Seleção

Para decidir se deve ou não tornar-se um membro da geração $G+1$, o vetor tentativa $u_{i,G+1}$ é comparado ao vetor alvo $x_{i,G}$ usando o critério *greedy*. Se o vetor $u_{i,G+1}$ alcançar um valor de função custo menor que $x_{i,G}$, então $x_{i,G+1}$ é conjunto para $u_{i,G+1}$, senão o antigo valor $x_{i,G}$ é mantido.

3.3.1.4. Parâmetros de projeto da ED

O seguinte conjunto de regras da linguagem tem surgido como ajuda para a escolha das variáveis de controle F (fator de valor constante e real), CR (probabilidade de cruzamento) e NP (tamanho da população), conforme STORN e PRICE (1997):

- A população inicial deve ser gerada o mais próximo possível da superfície da função objetivo;
- Frequentemente a probabilidade de cruzamento $CR \in [0; 1]$ deve ser considerada menor do que um (por exemplo 0,3). Caso não ocorra convergência, uma $CR \in [0,8; 1]$ pode ajudar;
- Para muitas aplicações NP de $10 * D$ (onde D é igual à dimensão ou ao número de variáveis) é uma boa escolha. F é, normalmente, escolhido no intervalo $[0,5; 1,0]$;
- Quanto maior for o tamanho da população escolhida, menor é o valor de F ;
- Tem-se uma boa taxa de convergência quando os parâmetros do melhor membro da população variam muito de geração para geração, especialmente durante o início do

processo de minimização. Mesmo se o seu valor da função objetivo decrescer lentamente;

- Não há necessariamente um desempenho inapropriado, quando o valor da função objetivo do melhor membro da população apresentar platôs durante o processo de minimização. Entretanto, isto indica que a minimização leva um tempo maior até encontrar o mínimo global (ou próximo dele) ou que o aumento do tamanho da população pode ser benéfico para a convergência;
- O valor da função objetivo do melhor membro da população não pode cair de forma brusca. Caso aconteça, a otimização está em um mínimo local;
- A escolha apropriada da função objetivo é crucial. Quanto maior a inclusão do conhecimento sobre o problema na função objetivo, maior a possibilidade de uma convergência suave e adequada (FALCONE, 2004).

3.3.1.5. Otimização do EKF usando de Evolução Diferencial

Na seção anterior foi mostrado como o EKF possibilita lidar com informações ruidosas do sistema dinâmico, utilizando informações matemáticas para estimar o estado atual do ambiente no qual está inserido. Contudo, uma dificuldade no projeto do filtro de Kalman foi o projeto das matrizes P , Q e R .

Neste contexto, este trabalho propõe a aplicação da heurística ED para otimizar os termos da diagonal principal das matrizes de P , Q e R .

As diversas implementações da ED são representadas na forma “ED/ $\alpha/\beta/\delta$ ”, onde:

- α especifica o modo de seleção dos genitores para a operação de mutação (vetor base). Geralmente utiliza-se *rand* (seleção aleatória) ou *best* (o vetor se menor custo da população corrente);
- β representa o número de diferenças (vetor de diferenças) para perturbar os indivíduos do vetor base. Em geral, uma ou duas diferenças é suficiente para aumentar satisfatoriamente a diversidade da população;
- δ denota o mecanismo de cruzamento, sendo mais comum o uso do *bin* (controlado por experimentos com distribuição binomial).

4. Apresentação e Análise dos Resultados

Foram implementados os modelos para previsão de séries temporais e realizadas análises comparativas entre tais modelos a fim de se analisar os resultados obtidos. Para uma previsão mais eficiente, os modelos devem receber valores obedecendo a determinados limites. Valores variando em uma ordem de magnitude muito grande podem inclusive inviabilizar o uso de determinados modelos.

Devido a estas preocupações, os valores de entrada devem ser “transformados” para permanecer dentro de limites. A este processo de transformação dá-se o nome de normalização. No presente trabalho a normalização foi realizada aplicando-se o logaritmo natural na série estudada.

Para a implementação dos testes de raízes unitárias foi utilizado o módulo PCGIVE da ferramenta GIVEWIN 2.20 (OXMETRICS, 2003), para implementação dos modelos ARIMA foi utilizada a ferramenta STATISTICA 7.1.280.0 (STATSOFT, 2005) e para implementação dos modelos de Redes Neurais foi utilizado o *software* MATLAB versão 7.0 (MATHWORKS, 2008).

4.1 Coleta e Tratamento dos Dados

O Indicador de preço de soja CEPEA/ESALQ (Centro de Estudos Avançados em Economia Aplicada/Escola Superior de Agricultura “Luiz de Queiroz”), antes denominado ESALQ/BM&F (Escola Superior de Agricultura “Luiz de Queiroz”/Bolsa de Mercadorias & Futuros), é uma média aritmética ponderada dos preços observados no Paraná para soja em grão a granel, tipo exportação, conforme padrão CONCEX: até 14% de umidade, até 2% de impurezas, e limites máximos de 8% para grãos avariados (até 5% de ardidos, aqueles com pelo menos um quarto de sua superfície com descolorações) e 30% de grãos quebrados (CEPEA, 2008)

O Indicador da Soja é divulgado diariamente pelo CEPEA desde agosto de 1997 e é utilizado por agentes e instituições que lidam com o mercado físico como uma das principais referências de preços. Apesar de o Indicador considerar apenas os preços do Paraná, um

Estado considerado importante na produção, industrialização e exportação de soja, a equipe CEPEA consulta agentes também nos Estados do Rio Grande do Sul, Mato Grosso, Mato Grosso do Sul, Goiás, Minas Gerais e São Paulo (CEPEA, 2008).

Para a obtenção do Indicador da Soja, o Estado do Paraná foi dividido em cinco regiões: Porto de Paranaguá, Ponta Grossa, Sudoeste, Oeste e Norte. Em cada região é consultada diariamente uma amostra representativa de compradores, vendedores, corretores e comerciantes que fornecem os preços, livres de impostos, que compõem o Indicador deste dia. Os cálculos do desconto de pagamento à vista são feitos a partir da NPR (Nota Promissória Rural) coletada diariamente junto a vários bancos (CEPEA, 2008).

O Indicador da Soja é determinado pela média aritmética geral das médias aritméticas simples regionais ponderadas pelas participações relativas das regiões na capacidade estática total de esmagamento do Estado do Paraná, no ano imediatamente anterior. Por exemplo, para 1997, as ponderações regionais para o indicador de preços de soja foram de: Porto de Paranaguá (19,13%), Ponta Grossa (25,49%), Norte (34,72%), Oeste (9,73%) e Sudoeste (10,93%). O percentual de participação de cada região é determinado pela sua capacidade instalada de esmagamento (CEPEA, 2008).

Essas participações são atualizadas, desde que a média móvel dos últimos dois anos referente à capacidade estática de processamento do estado do Paraná, com dados fornecidos pela ABIOVE (Associação Brasileira das Indústrias de Óleos Vegetais), apresente mudanças estatisticamente significativas. Em caso de não existirem preços em certa região, esta localidade não é considerada e sua participação é distribuída proporcionalmente entre as demais para cálculo do Indicador (CEPEA, 2008).

4.2 Simulações ARIMA

Esta série de preços diários da soja apresenta tendência crescente, como pode ser verificado na Figura 4.1. Embora a primeira metade dos dados de validação (500 primeiras amostras) mostre alguma estabilidade com um pequeno viés de alta, sua variância é pequena se comparada ao restante da série, onde ocorrem os três principais picos da série (amostras 1332, 1644 e 2627) e seu maior vale (entre o segundo e o terceiro picos mencionados).

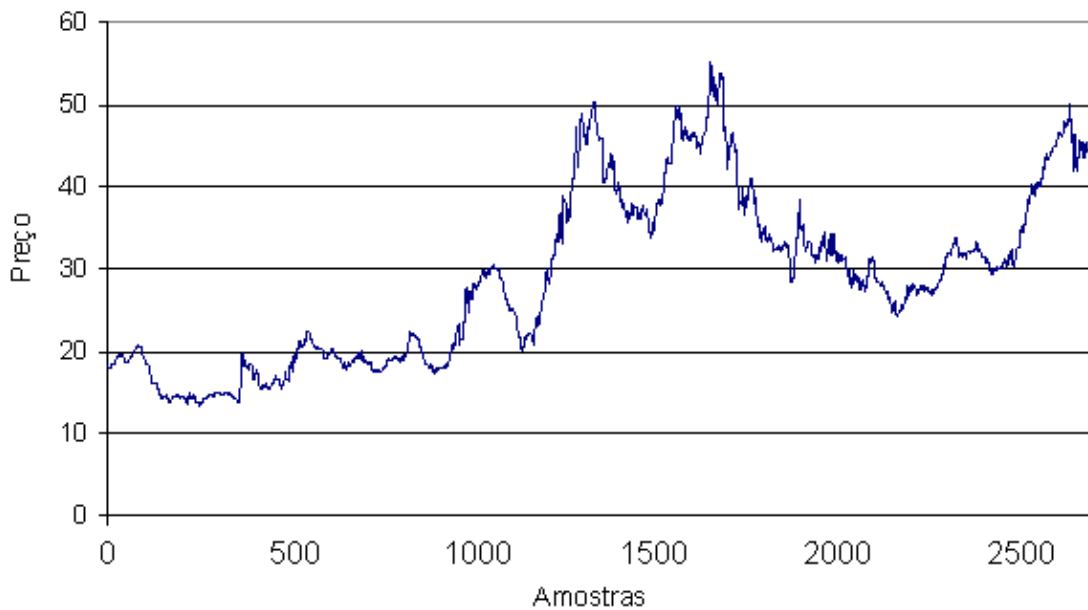


Figura 4.1 – Série temporal do preço da soja adotada neste trabalho.

Para verificação da estacionariedade da série foi realizado o teste de hipótese de Dickey-Fuller Aumentado (teste ADF). O resultado foi que a hipótese de não-estacionariedade foi aceita em nível de 5% de significância. Assim, em virtude da evidência de não-estacionariedade, o processo natural é a tomada de uma diferença na série. Utilizando a primeira diferença e reaplicando o teste ADF verificou-se que a série é estacionária. Assim, o teste indica que a ordem de integração deve ser igual a um, ou seja, a série temporal é caracterizada como integrada de primeira ordem ($d = 1$).

Visando a definição do modelo ARIMA mais apropriado foram realizados os testes ACF e PACF. A Figura 4.2 apresenta a função ACF, e o PACF mostrado na Figura 4.3 apresenta somente a primeira defasagem significativa, indicando um modelo AR(1) (WERNER, 2003).

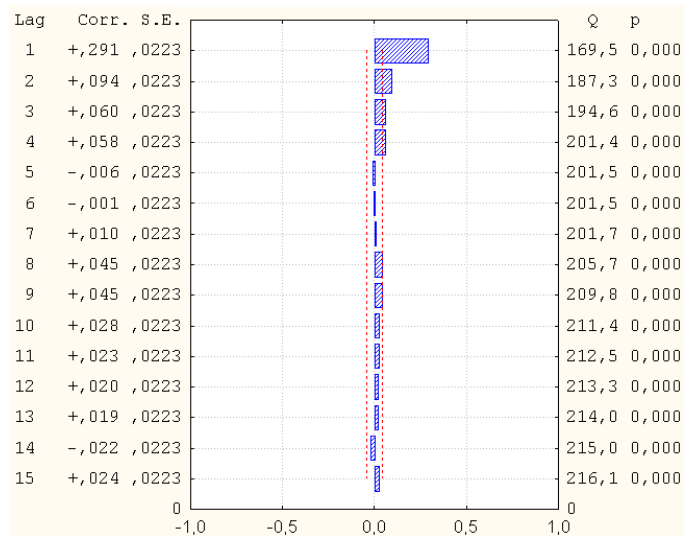


Figura 4.2 – Função de Auto-Correlação Amostral (ACF).

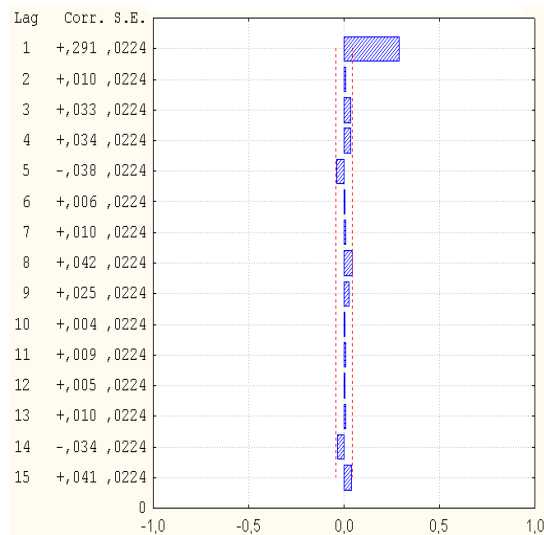


Figura 4.3 – Função de Auto-Correlação Parcial (PACF).

Portanto, a identificação do modelo por meio da análise combinada de todos os testes (DF, ACF e PACF) indica que o modelo pode ser um ARIMA (1,1,0).

Os coeficientes estimados foram iguais a $\phi_1 = 0,29089$ e $\mu = 0,00027$. Entretanto, a Figura 4.4 mostra que a verificação do modelo por meio da análise de resíduos a partir da análise ACF indica a existência de auto-correlações significativas, isto é, os resíduos não apresentam comportamento aleatório. Neste caso, deve-se identificar um novo modelo, até que os ruídos não revelem um padrão específico, em outras palavras, o resíduo deve ser caracterizado como ruído branco, ou seja, média igual a zero.

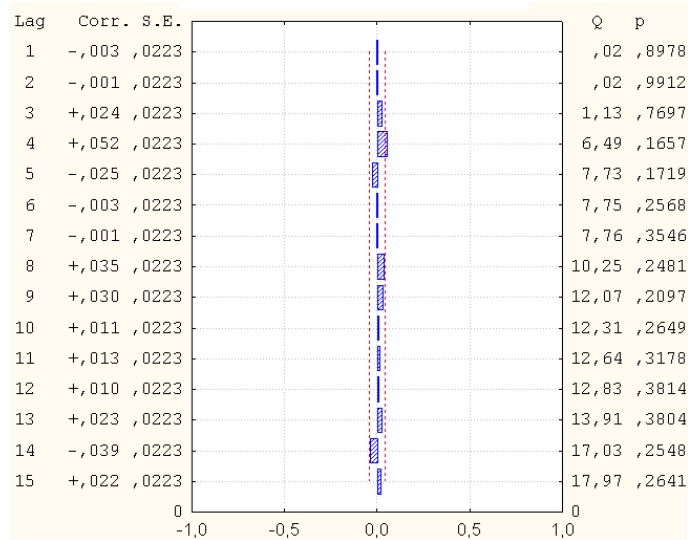


Figura 4.4 – ACF para os resíduos do modelo ARIMA (1,1,0).

A nova tentativa é a escolha de um modelo ARIMA (1,1,1). Para este modelo, os coeficientes estimados foram iguais a $\phi_1 = 0,33182$, $\theta_1 = 0,04479$ e $\mu = 0,00027$. Nesta ocasião, a Figura 4.5 mostra que a função ACF para análise de resíduos indica a ausência de autocorrelações significativas, entretanto, por meio da função PACF disposta na Figura 4.6 os ruídos continuam apresentando um padrão.

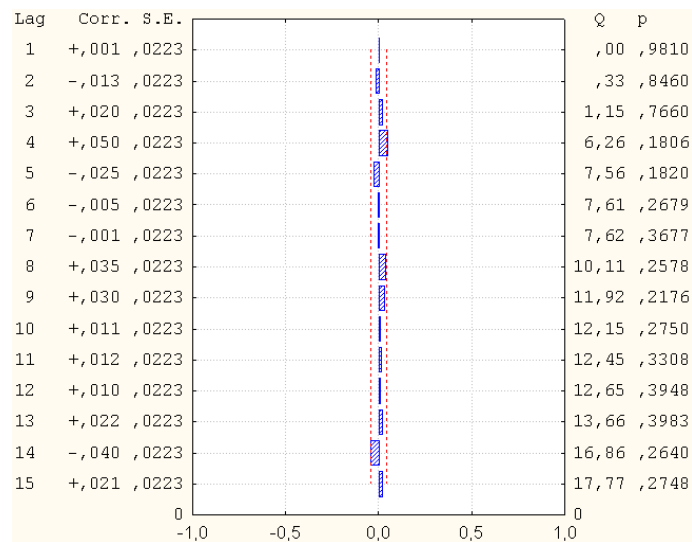


Figura 4.5 – ACF para os resíduos do modelo ARIMA (1,1,1).

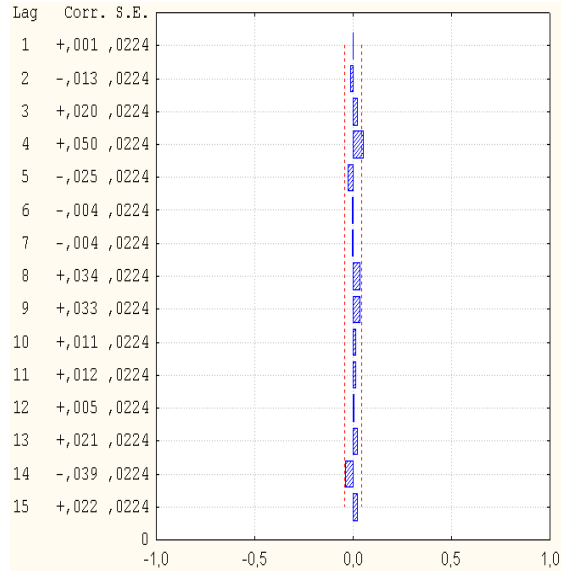


Figura 4.6 – PACF para os resíduos do modelo ARIMA (1,1,1).

Ao se escolher um modelo ARIMA (1,1,2), tanto a função ACF quanto a PACF, apresentados nas Figuras 4.7 e 4.8, respectivamente, sinalizam possuírem um comportamento tipo ruído branco. Para este modelo, os coeficientes estimados foram iguais a $\phi_1 = 0,65367$, $\theta_1 = 0,36743$, $\theta_2 = 0,10433$ e $\mu = 0,00027$. O MSE para este modelo na previsão de curtíssimo prazo (um passo à frente) foi de $1,6 \times 10^{-4}$.

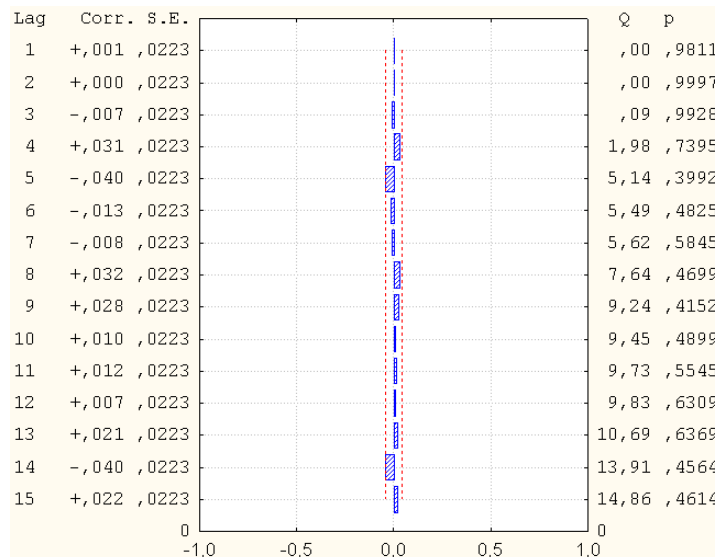


Figura 4.7 – ACF para os resíduos do modelo ARIMA (1,1,2).

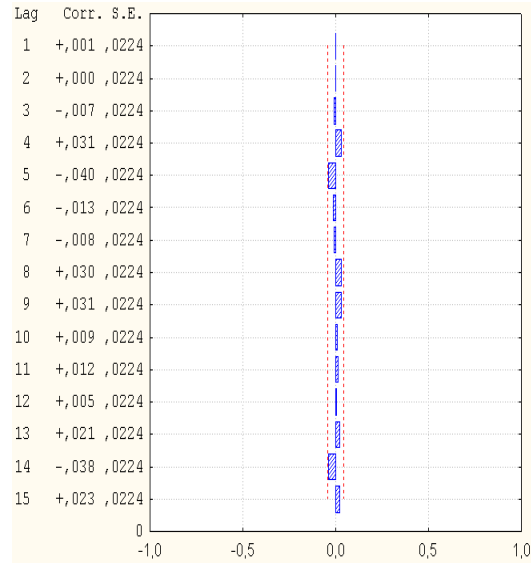


Figura 4.8 – PACF para os resíduos do modelo ARIMA (1,1,2).

4.3 Simulações RN-RBF

Os dados utilizados para a fase de treinamento (estimação do modelo) são as 2000 primeiras amostras, ou seja, compreende o período de 29/07/1997 até 16/08/2005. Os dados utilizados na fase de teste (validação do modelo) são as 687 amostras restantes, de 17/08/2005 até 26/05/2008. Todos estes dados estão disponíveis no *site* do CEPEA (2008).

Foram utilizados quatro neurônios na camada de entrada, representando os preços da soja com 1, 2, 3 e 4 defasagens $y(n-1)$, $y(n-2)$, $y(n-3)$ e $y(n-4)$ e um neurônio na camada de saída $\hat{y}(n)$, representando o preço estimado da soja no tempo n , a ser comparado com o preço real da soja $y(n)$ no mesmo tempo n .

A camada oculta de uma RN-RBF pode ser generalizada por meio de uma função da seguinte forma (SIMON, 2002):

$$g(v) = [g_0(v)]^{\frac{1}{1-p}}. \quad (4.1)$$

O parâmetro p deve ser um real maior que 1 para que a função geradora $g_0(v)$ satisfaça as seguintes condições:

$$g_0(v) > 0 \quad \forall v \in (0, \infty), \quad (4.2)$$

$$g'_0(v) > 0 \quad \forall v \in (0, \infty), \quad (4.3)$$

$$\frac{p}{p-1} [g'_0(v)]^2 - g_0(v)g''_0(v) > 0 \quad \forall v \in (0, \infty). \quad (4.4)$$

As condições 4.3 e 4.4 asseguram que $g'(v) < 0$ e $g''(v) > 0 \quad \forall v \in (0, \infty)$. O parâmetro p deve ser um real menor que 1 para que a função geradora $g_0(v)$ satisfaça as seguintes condições:

$$g_0(v) > 0 \quad \forall v \in (0, \infty), \quad (4.5)$$

$$g'_0(v) < 0 \quad \forall v \in (0, \infty), \quad (4.6)$$

$$\frac{p}{p-1} [g'_0(v)]^2 - g_0(v)g''_0(v) < 0 \quad \forall v \in (0, \infty). \quad (4.7)$$

As condições 4.6 e 4.7 novamente garantem que $g'(v) < 0$ e $g''(v) > 0 \quad \forall v \in (0, \infty)$. A função multiquadrática inversa (Equação 4.8)

$$g(v) = (v^2 + \beta^2)^{\frac{-1}{2}}, \quad (4.8)$$

onde β é uma constante real, satisfaz estas condições, por este motivo ela é utilizada como função de ativação neste trabalho.

Para efeitos de simulação, as RNs-RBF foram parametrizadas conforme mostrado na Tabela 4.1.

Tabela 4.1 – Parâmetros da RN-RBF.

ε	γ	p	η
10^{-3}	1	2	10^{-3}

Os parâmetros da função geradora γ e p foram fixados em 1 e 2, respectivamente. O parâmetro ε (erro-delta de *threshold* para parada do treinamento) foi fixado em 10^{-3} . Para o treinamento por descida de encosta (gradiente), o valor de η (tamanho do passo) foi fixado em 10^{-3} . O número de centros de funções de base radial foi variado entre 5 e 25.

Os resultados obtidos para previsão “um passo à frente” por meio do treinamento com o algoritmo de “descida de encosta” são mostrados na Tabela 4.2.

Tabela 4.2 – Resultados da simulação do algoritmo de treinamento por descida de encosta sem aplicação de médias móveis para previsão um passo à frente.

Número de centros	MSE	R ²	Tempo de Simulação [segundos]
5	1,066 x 10 ⁻³	0,99331	90
6	8,55 x 10 ⁻⁴	0,99463	46
7	7,94 x 10 ⁻⁴	0,99502	202
8	1,471 x 10 ⁻³	0,99076	40
9	não convergiu		
10	1,308 x 10 ⁻³	0,99178	167
11	1,517 x 10 ⁻³	0,99047	180
12	7,9 x 10 ⁻⁴	0,99504	290
13	4,98 x 10 ⁻⁴	0,99688	301
14	1,155 x 10 ⁻³	0,99275	310
15	1,544 x 10 ⁻³	0,99031	168
16	2,23 x 10 ⁻³	0,986	456
17	5,25 x 10 ⁻⁴	0,9967	350
18	4,164 x 10 ⁻³	0,97386	474
19	1,172 x 10 ⁻³	0,99264	141
20	3,007 x 10 ⁻³	0,98112	162
21	5,685 x 10 ⁻³	0,9643	94
22	2,211 x 10 ⁻³	0,98612	643
23	1,951 x 10 ⁻³	0,98775	447
24	6,354 x 10 ⁻³	0,96011	225
25	4,166 x 10 ⁻³	0,97384	40

Para o treinamento por EKF, os valores das matrizes de covariância foram otimizados inicialmente pelo algoritmo de AG com representação binária, o algoritmo de computação evolutiva mais conhecido. Os espaços de busca iniciais utilizados foram:

- P (erro de estimação do estado) $\in [10^0, 10^2]$;
- Q (ruído da dinâmica do processo) $\in [10^0, 10^2]$;
- R (ruído de observação) $\in [10^{-6}, 10^{-4}]$.

Os demais parâmetros do algoritmo de otimização por AG foram:

- População: 6 indivíduos;
- Critério de parada: 3 gerações;
- Probabilidade de cruzamento: 80%;
- Probabilidade de mutação: 5%;
- Estratégia de Algoritmo Genético: elitista.

Os resultados obtidos para o treinamento com o EKF otimizado por AG são mostrados na Tabela 4.3.

Tabela 4.3 – Resultados da simulação do algoritmo de treinamento por EKF otimizado por AG sem aplicação de médias móveis para previsão um passo à frente.

Número de centros	MSE	R^2	Tempo de Simulação [segundos]
5	$1,079 \times 10^{-3}$	0,99323	285
6	$5,1 \times 10^{-4}$	0,9968	287
7	$4,19 \times 10^{-4}$	0,99737	289
8	$2,58 \times 10^{-4}$	0,99838	293
9	$3,36 \times 10^{-4}$	0,99789	297
10	$2,71 \times 10^{-4}$	0,9983	301
11	$3,14 \times 10^{-4}$	0,99803	305
12	$2,25 \times 10^{-4}$	0,99859	309
13	$2,83 \times 10^{-4}$	0,99822	313
14	$2,15 \times 10^{-4}$	0,99865	317
15	$2,01 \times 10^{-4}$	0,99874	321
16	$2,15 \times 10^{-4}$	0,99865	327
17	$2,42 \times 10^{-4}$	0,99848	331
18	$1,99 \times 10^{-4}$	0,99875	336
19	$2,06 \times 10^{-4}$	0,99871	340
20	$1,95 \times 10^{-4}$	0,99878	345
21	$1,82 \times 10^{-4}$	0,99886	350
22	$2,09 \times 10^{-4}$	0,99869	356
23	$1,93 \times 10^{-4}$	0,99879	366
24	$1,9 \times 10^{-4}$	0,99881	368
25	$1,67 \times 10^{-4}$	0,99895	374

Na segunda forma de treinamento por EKF, os valores das matrizes de covariância foram otimizados pelo algoritmo de ED utilizando os seguintes espaços de busca iniciais:

- P (erro de estimação do estado) $\in [10^0, 10^2]$;
- Q (ruído da dinâmica do processo) $\in [10^0, 10^2]$;
- R (ruído de observação) $\in [10^{-6}, 10^{-4}]$.

Os demais parâmetros do algoritmo de otimização por ED foram:

- População: 6 indivíduos;
- Critério de parada: 3 gerações;
- Probabilidade de cruzamento: 80%;
- Estratégia de Evolução Diferencial: rand/1/bin.

Os resultados obtidos para o treinamento com o EKF otimizado por ED são mostrados na Tabela 4.4.

Tabela 4.4 – Resultados da simulação do algoritmo de treinamento por EKF otimizado por ED sem aplicação de médias móveis para previsão um passo à frente.

Número de centros	<i>MSE</i>	<i>R</i> ²	Tempo de Simulação [segundos]
5	$1,077 \times 10^{-3}$	0,99324	57
6	$5,1 \times 10^{-4}$	0,9968	58
7	$3,74 \times 10^{-4}$	0,99765	58
8	$2,28 \times 10^{-4}$	0,99857	59
9	$3,11 \times 10^{-4}$	0,99805	60
10	$2,56 \times 10^{-4}$	0,99839	61
11	$2,87 \times 10^{-4}$	0,9982	62
12	$1,9 \times 10^{-4}$	0,99881	63
13	$2,11 \times 10^{-4}$	0,99868	64
14	$1,85 \times 10^{-4}$	0,99884	65
15	$1,85 \times 10^{-4}$	0,99884	66
16	$1,83 \times 10^{-4}$	0,99885	67
17	$2,01 \times 10^{-4}$	0,99874	68
18	$1,65 \times 10^{-4}$	0,99896	69
19	$1,76 \times 10^{-4}$	0,9989	70
20	$1,58 \times 10^{-4}$	0,99901	71
21	$1,63 \times 10^{-4}$	0,99897	72
22	$1,64 \times 10^{-4}$	0,99897	73
23	$1,64 \times 10^{-4}$	0,99897	75
24	$1,57 \times 10^{-4}$	0,99901	75
25	$1,52 \times 10^{-4}$	0,99905	77

4.4 Análises Comparativas

Uma análise comparativa inicial somente entre os modelos de RN-RBF, mostra que o desempenho do treinamento por EKF otimizado por ED é amplamente superior tanto em termos do menor erro quadrático médio (Figura 4.9) quanto em termos de esforço computacional (Figura 4.10). Os resultados obtidos para o treinamento com o EKF por AG e ED são mostrados na Tabela 4.3 e 4.4, respectivamente.

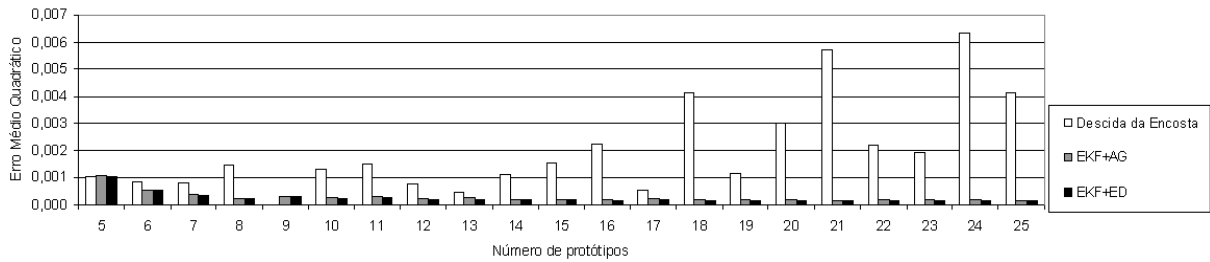


Figura 4.9 – Análise comparativa do *MSE* para os algoritmos de treinamento por descida de encosta e EKF otimizado por AG e por ED.

A partir da análise do *MSE*, o treinamento por descida de encosta apresentou um desempenho razoável com poucos centros (protótipos) de funções de base radial. Somente em 5 casos (6, 7, 12, 13 e 17 centros) o *MSE* foi inferior a 10^{-3} .

Embora oscile muito, tal algoritmo tende a perder desempenho à medida que se aumenta o número de centros. Isto se desconsiderando ainda o caso em que o modelo não convergiu (configuração com 9 centros).

O melhor desempenho para este algoritmo ficou por conta da configuração com 13 centros, por meio da qual o *MSE* foi de 10^{-4} .

Contudo, os algoritmos de treinamento por EKF apresentam comportamento oposto em termos de tendência. Conforme se aumenta o número de centros, os desempenhos tornam-se superiores.

Além disso, exceto para a configuração com 5 centros (em ambos os casos de otimização, por AG e por ED), todas as outras simulações mostraram um *MSE* menor do que 10^{-3} , sendo piores que o descida de encosta somente nesta situação (configuração com 5 centros), nas quais mesmo assim o desempenho não mostrou-se tão inferior.

Quando comparados entre si, a otimização por ED mostra-se ligeiramente superior a por AG em todos os casos. Em termos absolutos, os melhores desempenhos em termos de *MSE* para a otimização por AG foi de $1,7 \times 10^{-4}$ e por ED foi de $1,5 \times 10^{-4}$. Em ambos os casos, a melhor configuração foi para 25 centros e os melhores valores para as matrizes de covariância do EKF (otimizado por ED) foram: $P = 30,0649$, $Q = 28,9516$ e $R = 2,9084 \times 10^{-5}$.

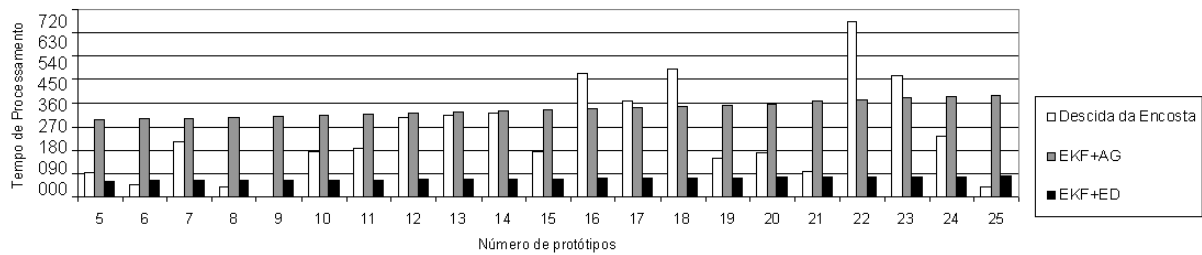


Figura 4.10 – Análise comparativa do tempo de processamento para os algoritmos de treinamento por descida de encosta e EKF otimizado por AG e por ED.

Tal como ocorre em termos da análise do *MSE*, o tempo de processamento do treinamento por descida de encosta também apresentou grande oscilação, mas com relativa tendência crescente à medida que se aumenta o número de centros.

Para o EKF otimizado por AG o custo computacional é praticamente estável (sempre dentro do intervalo de 4 e 7 minutos) com uma leve, porém permanente tendência crescente. Para o EKF otimizado com ED o custo computacional segue o mesmo padrão que o otimizado por AG, porém sempre com tempo de processamento inferior a 1 minuto e meio.

Comparando-se os três algoritmos de treinamento da RN-RBF, o tempo de processamento da descida da encosta foi menor somente em três situações (configuração com 6, 8 e 25 centros). Em todas as outras situações o EKF otimizado por ED foi menor. A Tabela 4.5 e a Figura 4.11 apresentam as melhores configurações para cada modelo analisado.

Tabela 4.5 – Melhores modelos sem aplicação de médias móveis nas RNs-RBF para previsão um passo à frente.

	Número de centros	<i>MSE</i>	<i>R</i> ²
ARIMA (1,1,2)	N/A	$1,6 \times 10^{-4}$	0,99896
RN-RBF (descida de encosta)	13	5×10^{-4}	0,99688
RN-RBF (EKF+AG)	25	$1,7 \times 10^{-4}$	0,99895
RN-RBF (EKF+ED)	25	$1,5 \times 10^{-4}$	0,99905

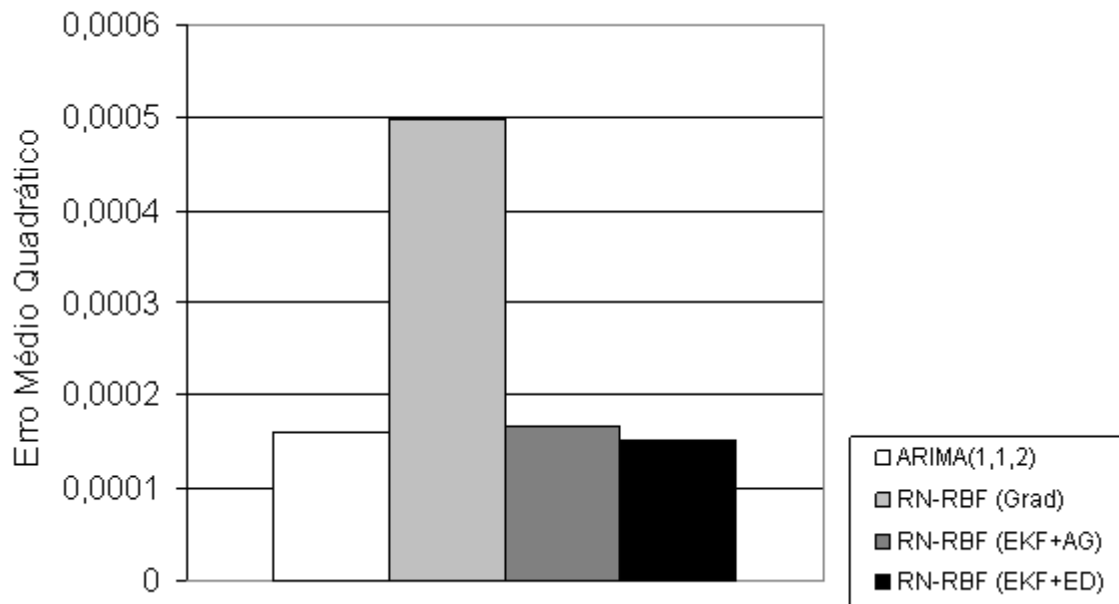


Figura 4.11 – Análise comparativa do MSE para os melhores modelos sem aplicação de médias móveis nas RNs-RBF para previsão um passo à frente.

Nesta primeira análise, a RN-RBF treinada com EKF otimizada por ED demonstrou desempenho superior para previsão de séries temporais financeiras em relação à treinada por descida de encosta em termos de *MSE*, mas um desempenho quase equivalente aos modelos ARIMA (1,1,2) e de RN-RBF treinada por EKF otimizada por AG, porém ainda assim superior.

Em uma nova análise, ao ser aplicada uma transformação dos dados com médias móveis (duas médias móveis foram suficientes, para ser análoga à aplicada no modelo ARIMA), o desempenho da RN-RBF com EKF otimizada por ED torna-se ainda melhor, conforme apresentado na Figura 4.12. A Tabela 4.6 apresenta o resultado das simulações da RN-RBF com EKF otimizada por ED com médias móveis.

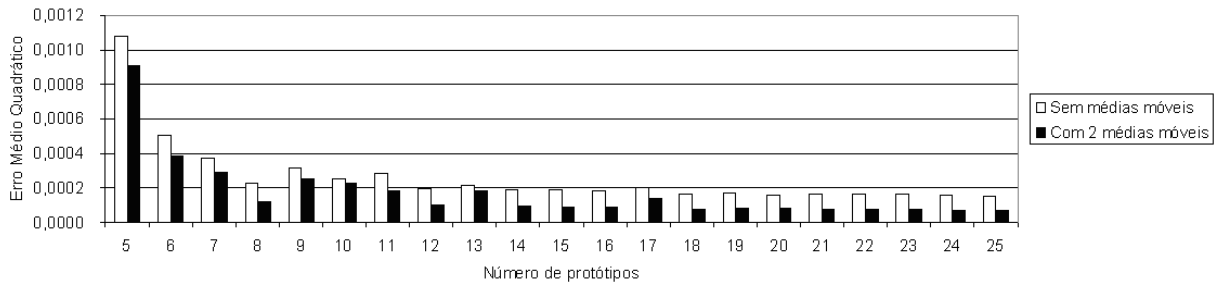


Figura 4.12 – Análise comparativa do *MSE* para a RN-RBF treinada por EKF otimizado por ED sem e com duas médias móveis.

Tabela 4.6 – Resultados da simulação do algoritmo de treinamento por EKF otimizado por ED com aplicação de duas médias móveis para previsão um passo à frente.

Número de centros	<i>MSE</i>	R^2	Tempo de Simulação [segundos]
5	$9,0587 \times 10^{-4}$	0,99431	59
6	$3,8416 \times 10^{-4}$	0,99759	59
7	$2,9640 \times 10^{-4}$	0,99814	61
8	$1,1792 \times 10^{-4}$	0,99926	61
9	$2,5052 \times 10^{-4}$	0,99843	61
10	$2,2954 \times 10^{-4}$	0,99856	62
11	$1,7918 \times 10^{-4}$	0,99887	63
12	$1,0110 \times 10^{-4}$	0,99936	64
13	$1,7794 \times 10^{-4}$	0,99888	65
14	$9,306 \times 10^{-5}$	0,99942	66
15	$8,966 \times 10^{-5}$	0,99944	66
16	$8,535 \times 10^{-5}$	0,99946	67
17	$1,3680 \times 10^{-4}$	0,99914	68
18	$7,651 \times 10^{-5}$	0,99952	69
19	$7,844 \times 10^{-5}$	0,99951	70
20	$8,120 \times 10^{-5}$	0,99949	71
21	$7,425 \times 10^{-5}$	0,99953	73
22	$7,010 \times 10^{-5}$	0,99956	74
23	$7,046 \times 10^{-5}$	0,99956	75
24	$6,785 \times 10^{-5}$	0,99957	76
25	$6,558 \times 10^{-5}$	0,99959	78

As melhores soluções dos treinamentos por descida de encosta e por EKF otimizado por AG também foram superior com as médias móveis, sendo que o treinamento por descida de encosta ainda assim não igualou o desempenho do modelo ARIMA (1,1,2), mas o EKF otimizado por AG foi superior.

Para o treinamento por descida de encosta, o *MSE* da melhor configuração (23 centros) foi igual a $3,8 \times 10^{-4}$, conforme observa-se na Tabela 4.7. Em contrapartida, além da configuração com 9 centros, que já sem as médias móveis não havia convergido, após a

aplicação das médias móveis a configuração com 25 centros também não convergiu (vide Tabela 4.7 e Figura 4.13).

Tabela 4.7 – Resultados da simulação do algoritmo de treinamento por descida de encosta com aplicação de duas médias móveis para previsão um passo à frente.

Número de centros	MSE	R^2	Tempo de Simulação [segundos]
5	$9,052 \times 10^{-4}$	0,99431	67
6	$8,767 \times 10^{-4}$	0,99449	57
7	$6,948 \times 10^{-4}$	0,99564	197
8	$1,1203 \times 10^{-3}$	0,99296	44
9	não convergiu		
10	$9,385 \times 10^{-4}$	0,99411	205
11	$1,6908 \times 10^{-3}$	0,98938	367
12	$8,489 \times 10^{-4}$	0,99467	111
13	$6,95 \times 10^{-4}$	0,99563	219
14	$8,86 \times 10^{-4}$	0,99443	356
15	$1,329 \times 10^{-3}$	0,99165	256
16	$1,19 \times 10^{-3}$	0,99253	474
17	$2,464 \times 10^{-3}$	0,98452	174
18	$5,0597 \times 10^{-3}$	0,96822	530
19	$7,8822 \times 10^{-3}$	0,95049	272
20	$1,4102 \times 10^{-3}$	0,99114	221
21	$1,332 \times 10^{-3}$	0,99163	559
22	$4,0223 \times 10^{-3}$	0,97474	378
23	$3,797 \times 10^{-4}$	0,99762	476
24	$2,3693 \times 10^{-3}$	0,98512	311
25	não convergiu		

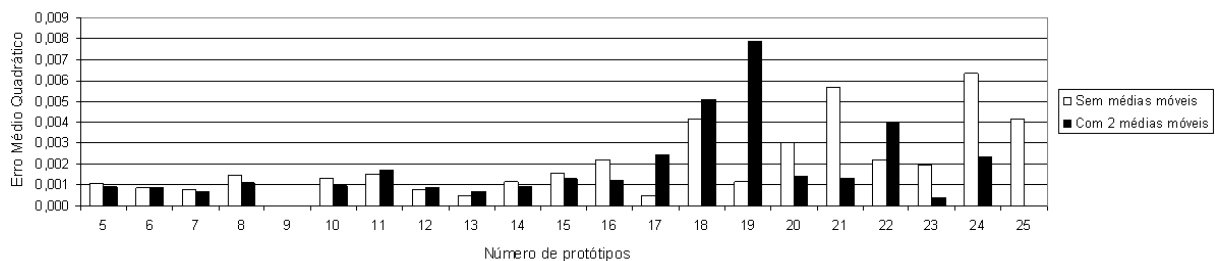


Figura 4.13 – Análise comparativa do MSE para a RN-RBF treinada por descida de encosta sem e com duas médias móveis.

Contudo, para o EKF otimizado por AG, o MSE da melhor configuração (25 centros) foi igual a $1,1 \times 10^{-4}$, conforme verifica-se na Tabela 4.8 e na Figura 4.14.

Tabela 4.8 – Resultados da simulação do algoritmo de treinamento por EKF otimizado por AG com aplicação de duas médias móveis para previsão um passo à frente.

Número de centros	<i>MSE</i>	<i>R</i> ²	Tempo de Simulação [segundos]
5	9,087 x 10 ⁻⁴	0,99429	291
6	3,856 x 10 ⁻⁴	0,99758	291
7	3,478 x 10 ⁻⁴	0,99782	300
8	2,125 x 10 ⁻⁴	0,99867	300
9	2,661 x 10 ⁻⁴	0,99833	303
10	3,135 x 10 ⁻⁴	0,99803	311
11	2,556 x 10 ⁻⁴	0,99839	301
12	1,686 x 10 ⁻⁴	0,99894	308
13	2,2 x 10 ⁻⁴	0,99862	314
14	1,527 x 10 ⁻⁴	0,99904	320
15	1,555 x 10 ⁻⁴	0,99902	326
16	1,424 x 10 ⁻⁴	0,99911	331
17	1,714 x 10 ⁻⁴	0,99892	335
18	1,463 x 10 ⁻⁴	0,99908	340
19	1,375 x 10 ⁻⁴	0,99914	345
20	1,494 x 10 ⁻⁴	0,99906	350
21	1,223 x 10 ⁻⁴	0,99923	356
22	1,243 x 10 ⁻⁴	0,99922	366
23	1,396 x 10 ⁻⁴	0,99912	365
24	1,341 x 10 ⁻⁴	0,99916	371
25	1,082 x 10⁻⁴	0,99932	376

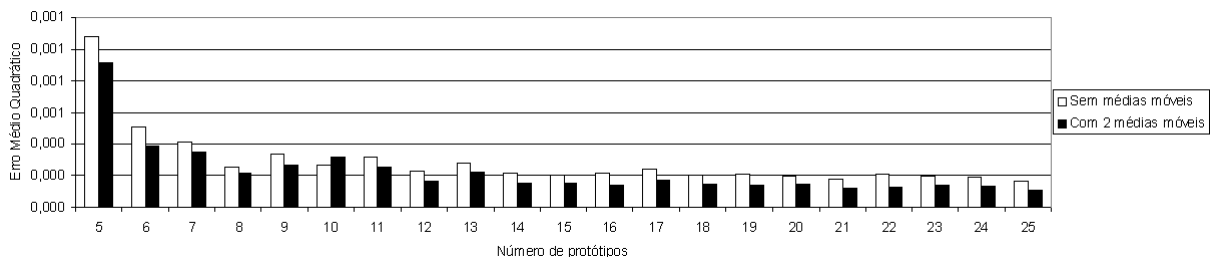


Figura 4.14 – Análise comparativa do *MSE* para a RN-RBF treinada por EKF otimizado por AG sem e com duas médias móveis.

A RN-RBF treinada com EKF otimizado por ED quando aplicadas as duas médias móveis demonstrou desempenho superior para previsão de séries temporais financeiras em relação aos modelos ARIMA (1,1,2), RN-RBF treinada por descida de encosta e RN-RBF treinada por EKF otimizada por AG em termos de *MSE* (e em termos de *R*²) conforme pode ser comprovado pela Tabela 4.9 e pela Figura 4.15.

A melhor configuração do EKF otimizado por ED (o melhor modelo com a aplicação de duas médias móveis) foi para 25 centros. As matrizes de covariância foram: $P = 77,1409$, $Q = 27,6108$ e $R = 6,5265 \times 10^{-5}$.

Tabela 4.9 – Melhores modelos aplicando-se duas médias móveis nas RNs-RBF para previsão um passo à frente.

	Número de centros	MSE	R^2
ARIMA (1,1,2)	N/A	$1,6 \times 10^{-4}$	0,99896
RN-RBF (descida de encosta)	23	$3,8 \times 10^{-4}$	0,99762
RN-RBF (EKF+AG)	25	$1,1 \times 10^{-4}$	0,99932
RN-RBF (EKF+ED)	25	7×10^{-5}	0,99959

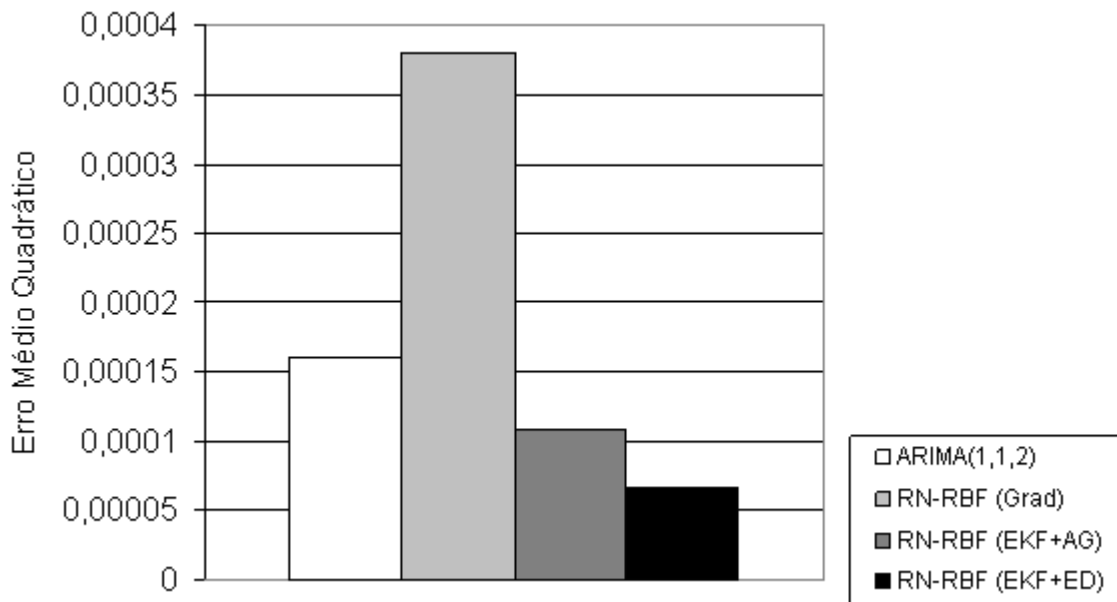


Figura 4.15 – Análise comparativa do MSE para os melhores modelos aplicando-se médias móveis nas RNs-RBF para previsão um passo à frente.

A Figura 4.16 apresenta a comparação entre o preço real (desejado) da soja e o preço estimado do melhor modelo (RN-RBF treinada por EKF otimizado com ED e médias móveis) e a Figura 4.17 apresenta o erro de previsão para o mesmo modelo.

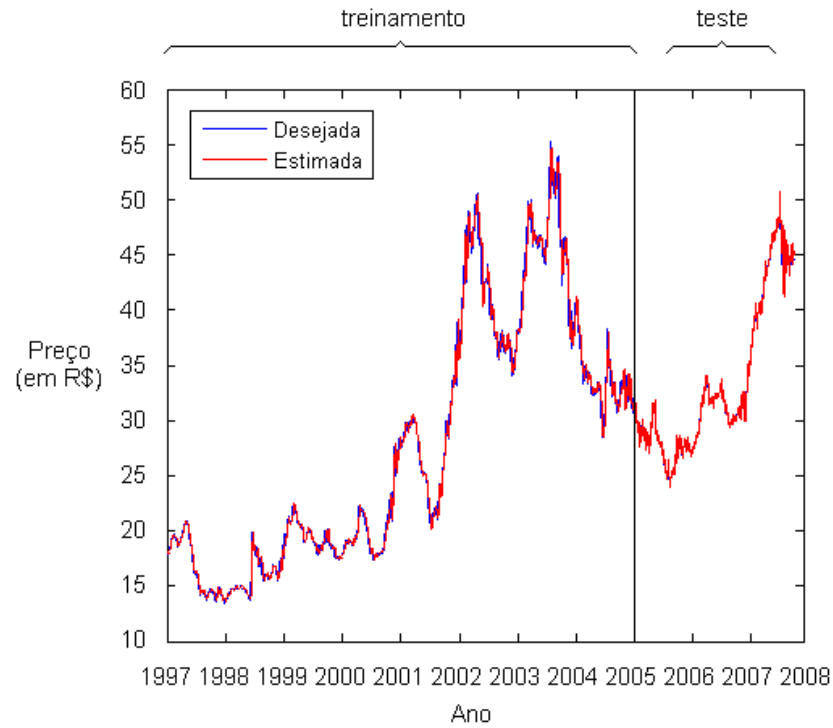


Figura 4.16 – Análise comparativa entre o preço real (desejado) e preço estimado do melhor modelo.

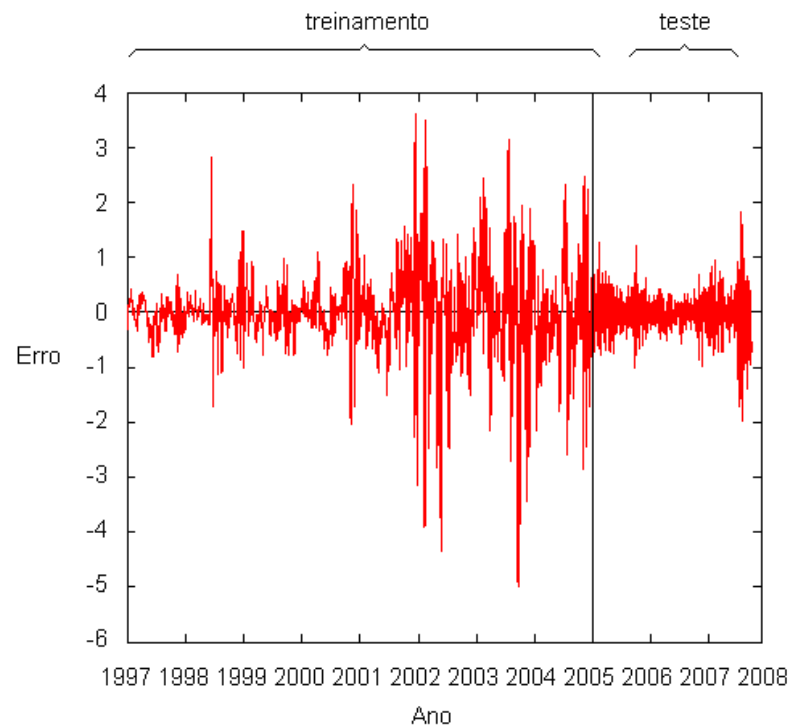


Figura 4.17 – Erro de previsão do melhor modelo.

Para se vislumbrar o desempenho dos modelos de RN em previsões de médio e longo prazo, foram realizadas também simulações de previsão “fora da amostra” com 20 e com 687 passos à frente (dados de validação). Nesta avaliação, foram utilizadas as melhores configurações de cada modelo de RN:

- Treinamento por Descida da Encosta com 23 centros e duas médias móveis;
- Treinamento por EKF otimizado por AG com 25 centros e duas médias móveis;
- Treinamento por EKF otimizado por ED com 25 centros e duas médias móveis.

O desempenho do treinamento por EKF foi novamente superior, com ligeira vantagem para o otimizado por ED para a previsão com 20 passos à frente (vide Figura 4.18).

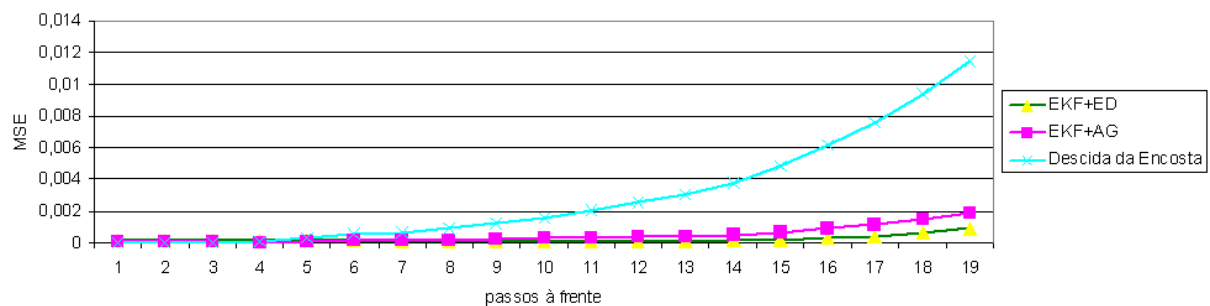


Figura 4.18 – *MSE* para previsão 20 passos à frente.

A partir da Figura 4.19, que apresenta a comparação entre os valores reais e os valores previstos pelos três métodos de RN-RBF com as 20 primeiras amostras de validação, se pode observar que a curva do algoritmo de descida da encosta segue acentuadamente na direção contrária do preço real.

O mesmo ocorre com as curvas dos algoritmos de EKF, porém de modo mais suave, principalmente para o otimizado por ED. A Figura 4.20, que apresenta o erro de previsão de cada método reforça esta conclusão.

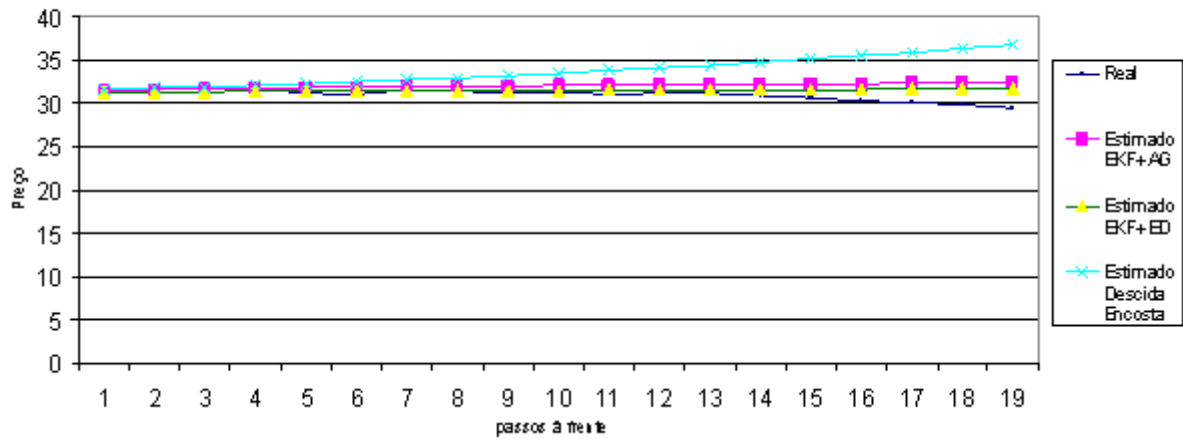


Figura 4.19 – Comparação entre os valores reais e os valores previstos pelos três métodos de RN-RBF com as 20 primeiras amostras de validação.

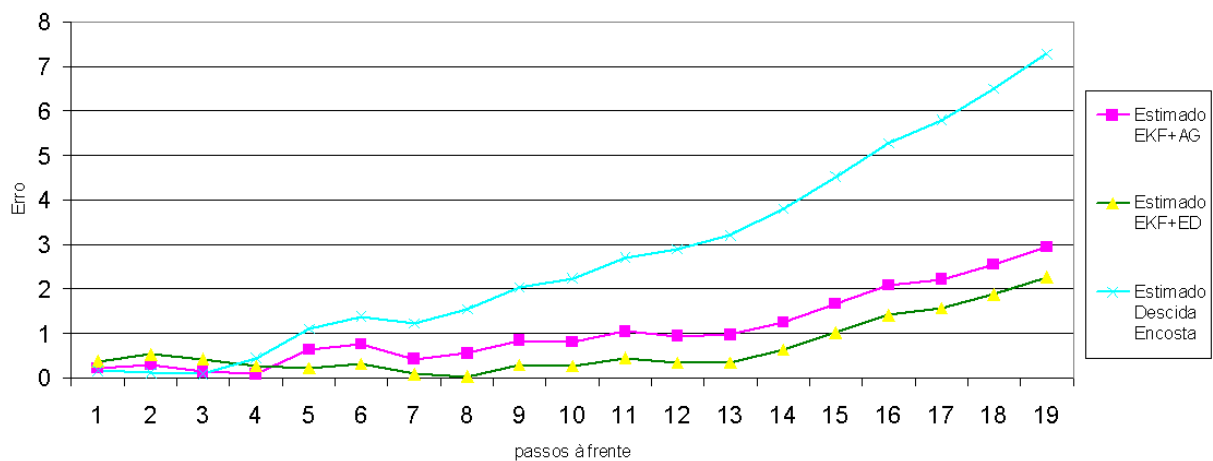


Figura 4.20 – Comparação entre os erros de previsão dos três métodos de RN-RBF com as 20 primeiras amostras de validação.

Para a previsão com 687 passos à frente, o EKF otimizado por AG demonstrou-se melhor conforme pode ser verificado pela Figura 4.21, todavia nesta situação os dados tornam-se espúrios, pois o erro acumulado que se propaga é muito alto.

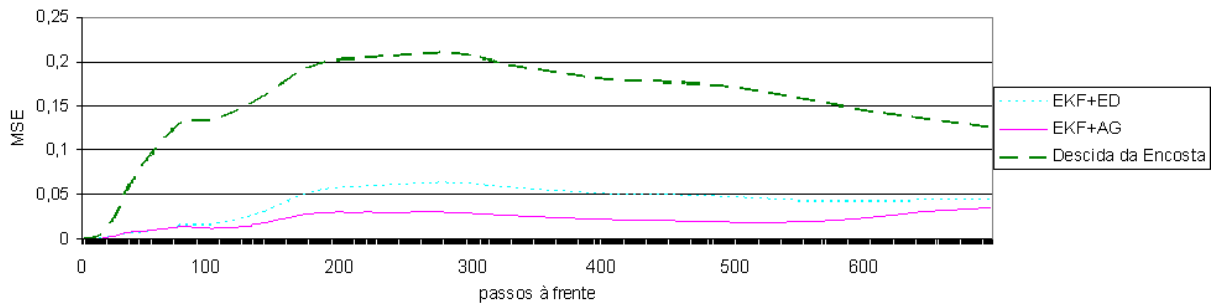


Figura 4.21 – MSE para previsão com todas as amostras de validação (687 passos à frente).

As Figuras 4.22 e 4.23 ilustram, respectivamente, a comparação entre os valores reais e previstos por meio de cada método de RN-RBF e seus erros de previsão para todas as amostras de validação.

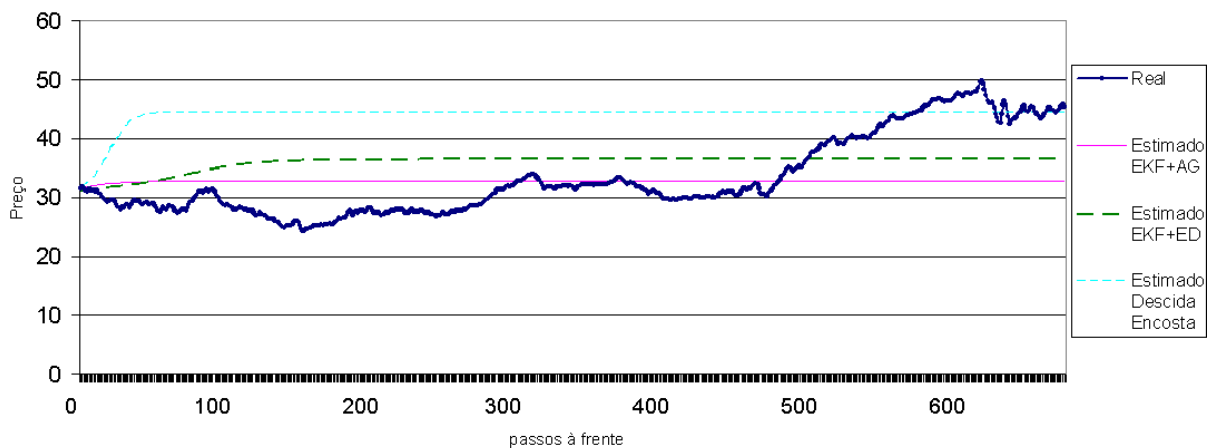


Figura 4.22 – Comparação entre os valores reais e os valores previstos pelos três métodos de RN-RBF com todas as amostras de validação (687 passos à frente).

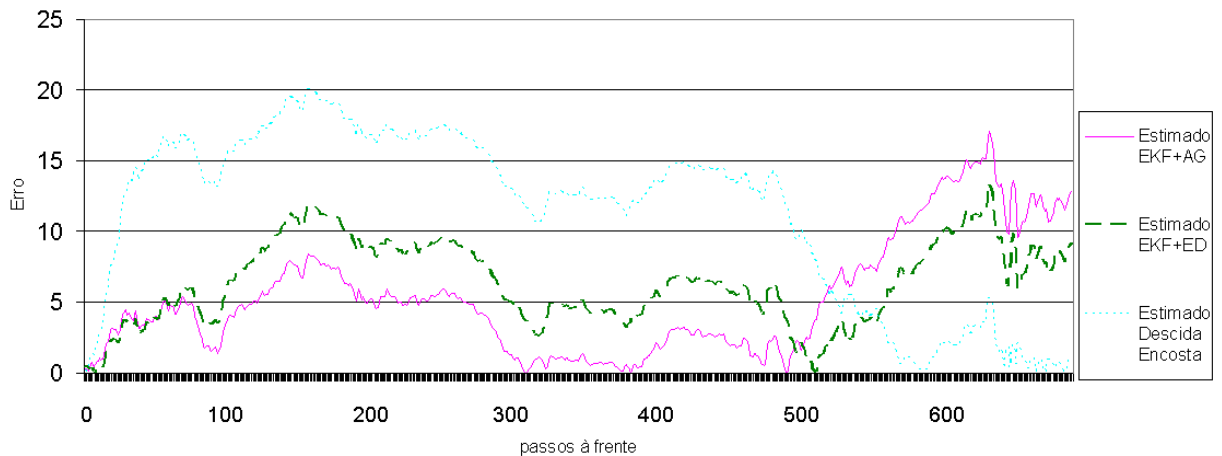


Figura 4.23 – Comparação entre os erros de previsão dos três métodos de RN-RBF com todas as amostras de validação (687 passos à frente).

O modelo otimizado por ED mostrou-se eficiente principalmente na previsão de curto prazo (um passo à frente). Todavia, sua aplicação somente a uma série temporal (série da soja) não é suficiente para generalizá-lo como um bom modelo para previsão de séries temporais de preços diários de *commodities* do agro-negócio.

Deste modo, os modelos de RN-RBF foram comparados também para duas outras séries para previsão de curto prazo. As séries escolhidas foram as de preços diários do café e do boi gordo. Estas séries e respectivas análises comparativas são apresentadas a seguir.

O Indicador de preço do café CEPEA/ESALQ é uma média aritmética ponderada dos preços do café arábica observados no Cerrado e Sul de Minas Gerais, Mogiana (SP), Paulista (SP) e noroeste do Paraná. A importância de cada região na ponderação é determinada pelo volume produzido em cada uma (CEPEA, 2008).

O Indicador é de preços diários, não considera impostos e refere-se a negociações no mercado físico de lotes (entre empresas). Os preços para valor a vista são tomados para sacas de 60 kg (R\$/saca) acrescidas dos custos de frete para a cidade de São Paulo. Os cálculos do desconto dos valores a prazo são feitos a partir da NPR (Nota Promissória Rural) coletada diariamente junto a vários bancos e os agentes consultados na pesquisa para formação do Indicador são cooperativas, corretores, torrefadores e exportadores (CEPEA, 2008).

A série de preços do café utilizada neste trabalho se inicia em 2 de Setembro de 1996 e termina em 21 de Setembro de 2004, possuindo 2000 amostras de estimação. Esta série apresenta grande volatilidade e, tal como a série da soja, também possui três grandes picos (amostras 180, 804 e 1928), porém distribuídos de forma diferente, como pode ser verificado na Figura 4.24.

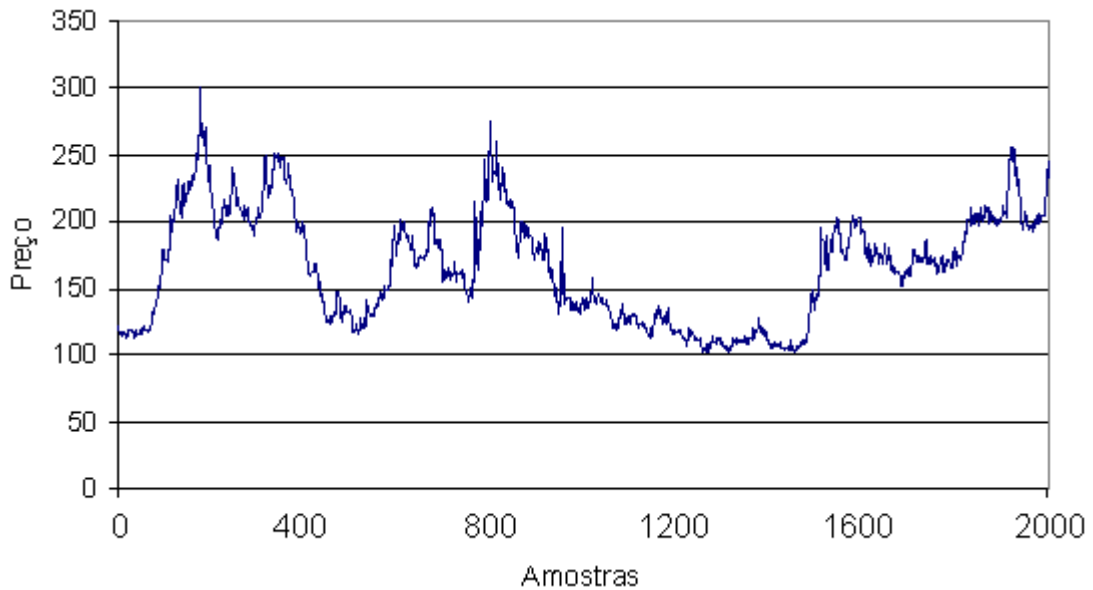


Figura 4.24 – Série temporal do preço do café adotada neste trabalho.

Em termos de *MSE*, novamente os algoritmos de treinamento por EFK mostraram-se superiores ao algoritmo de descida da encosta, com vantagem para o EFK otimizado por ED, como já havia sido verificado na série da soja. Esta análise comparativa é apresentada na Tabela 4.10 e na Figura 4.25.

Um detalhe importante é que também para a série do café, quanto maior o número de centros, melhor o desempenho das RNs-RBF treinadas por EKF, comportamento semelhante ao ocorrido nas simulações para a série da soja.

Tabela 4.10 – Análise comparativa do *MSE* para os algoritmos de treinamento por descida de encosta e EKF otimizado por AG e por ED para previsão um passo à frente dos preços do café.

Número de centros	Descida da Encosta	AG	ED
5	$8,043 \times 10^{-2}$	$5,883 \times 10^{-4}$	$5,874 \times 10^{-4}$
10	$1,047 \times 10^{-2}$	$4,856 \times 10^{-4}$	$4,357 \times 10^{-4}$
15	$8,679 \times 10^{-2}$	$3,151 \times 10^{-4}$	$2,069 \times 10^{-4}$
20	$1,571 \times 10^{-3}$	$2,559 \times 10^{-4}$	$1,995 \times 10^{-4}$
25	$3,948 \times 10^{-3}$	$2,398 \times 10^{-4}$	$1,938 \times 10^{-4}$

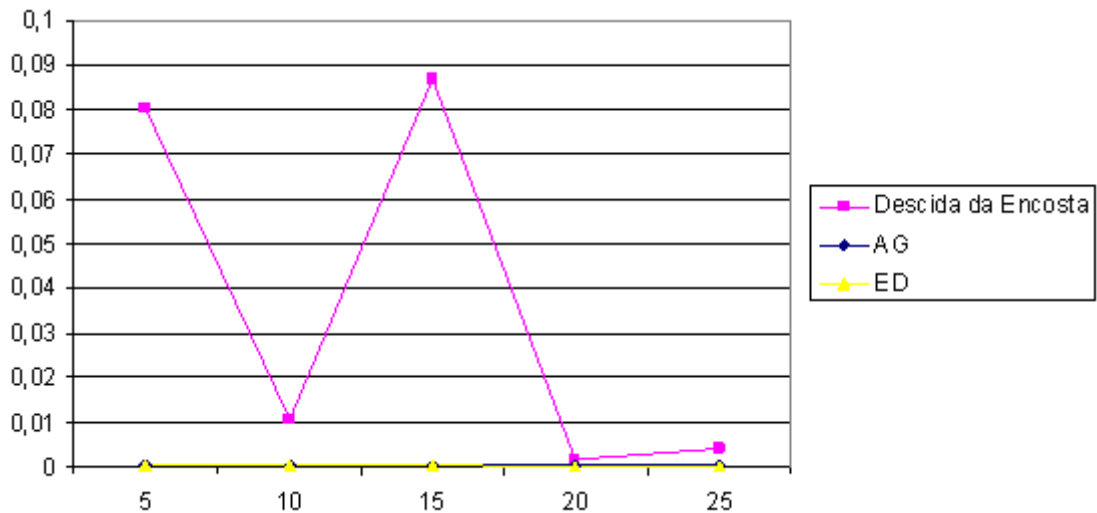


Figura 4.25 – Análise comparativa do *MSE* para os algoritmos de treinamento por descida de encosta e EKF otimizado por AG e por ED para previsão um passo à frente dos preços do café.

Para as estimações do preço do boi gordo foi utilizado o Indicador ESALQ/BM&F/BOVESPA. Este Indicador é uma média aritmética ponderada dos preços do boi gordo, castrado, comum, rastreado e com carcaça convexa acima de 450 kg observados no Estado de São Paulo, dividido em 4 regiões com base nos dados de volume de abate dos frigoríficos amostrados atualizado mensalmente. As regiões são Presidente Prudente, Araçatuba, Bauru/Marília e São José do Rio Preto (CEPEA, 2008).

Os preços para valor a vista referem-se ao arroba do boi gordo (R\$/arroba) a retirar na propriedade rural com desconto do produtor de 2,3% em impostos. Os agentes consultados na pesquisa para formação do Indicador são frigoríficos, pecuaristas, escritórios de compra e venda de gado e leiloeiras (CEPEA, 2008).

A série de preços do boi gordo utilizada neste trabalho se inicia em 23 de Julho de 1997 e termina em 5 de Agosto de 2005, possuindo também 2000 amostras de estimação. Esta série apresenta clara tendência crescente, como pode ser verificado na Figura 4.26.

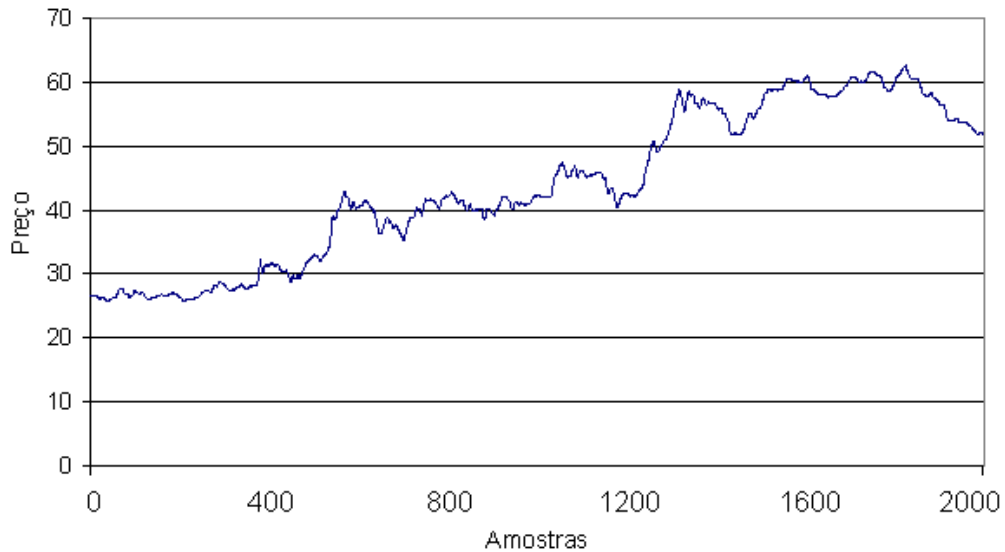


Figura 4.26 – Série temporal do preço do boi gordo adotada neste trabalho.

A Tabela 4.11 e a Figura 4.27 mostram que o *MSE* para os algoritmos de treinamento por EKF foram outra vez superiores ao algoritmo de descida da encosta. Novamente dentre as otimizações do EKF, a por ED mostrou-se mais eficiente que a por AG, como já havia sido verificado nas séries da soja e do café. Também como nas outras séries, a série do boi gordo apresentou o melhor desempenho para 25 centros de funções radiais de base.

Tabela 4.11 – Análise comparativa do *MSE* para os algoritmos de treinamento por descida de encosta e EKF otimizado por AG e por ED para previsão um passo à frente dos preços do café.

Número de centros	Descida da Encosta	AG	ED
5	$4,673 \times 10^{-3}$	$1,079 \times 10^{-4}$	$1,076 \times 10^{-4}$
10	$4,575 \times 10^{-3}$	$6,513 \times 10^{-5}$	$1,731 \times 10^{-5}$
15	$1,004 \times 10^{-3}$	$3,099 \times 10^{-5}$	$1,095 \times 10^{-5}$
20	$1,297 \times 10^{-3}$	$3,807 \times 10^{-5}$	$1,079 \times 10^{-5}$
25	$2,512 \times 10^{-3}$	$2,087 \times 10^{-5}$	$7,462 \times 10^{-6}$

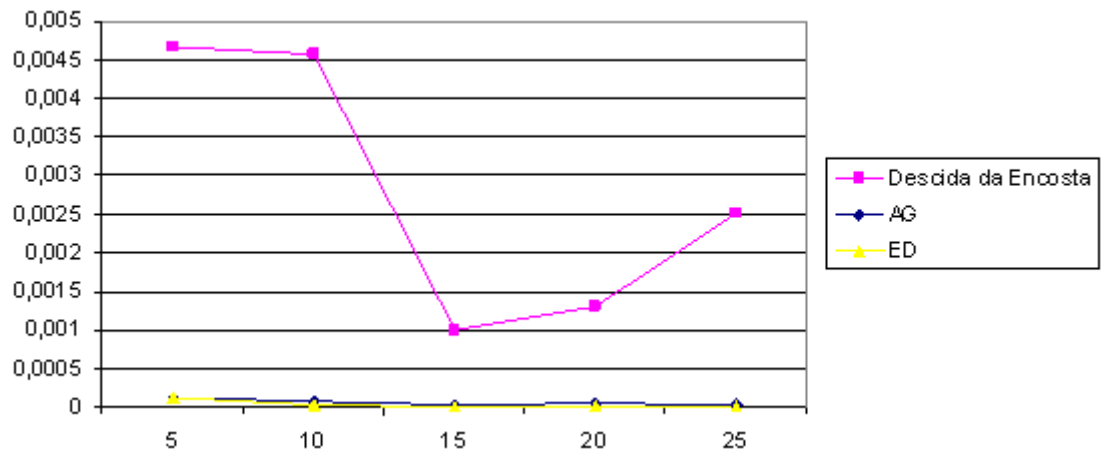


Figura 4.27 – Análise comparativa do MSE para os algoritmos de treinamento por descida de encosta e EKF otimizado por AG e por ED para previsão um passo à frente dos preços do boi gordo.

5. Conclusão

Recentemente, novos métodos têm surgido e obtido relativo sucesso na previsão de séries temporais, sobretudo aqueles que consideram em seus modelos o caráter não-linear presente nas estruturas de tais séries. Dentre tais métodos destacam-se as redes neurais.

Ainda assim, o sucesso da utilização de redes neurais para previsão de séries temporais depende de sua arquitetura e de seus algoritmos de aprendizagem. Conforme relatam alguns pesquisadores (SIMON, 2002; BIRGMEIER, 1995), as diversas formas de filtros de Kalman têm-se demonstradas aptas para treinamento de redes neurais, sobretudo RNs-RBF.

Uma variação do filtro de Kalman aplicada a sistemas não-lineares, como são os casos da maioria das séries temporais financeiras, é o EKF, eficiente para sistemas cuja não-linearidade é pouco significativa.

A vantagem no uso deste filtro se deve a vários fatores, destacando-se entre eles o tempo de processamento. Todavia, uma grande dificuldade para a utilização do filtro de Kalman é a escolha de seus parâmetros de sintonia (P_0 , Q e R). Este trabalho propôs a utilização da meta-heurística de Evolução Diferencial (STORN e PRICE, 1995) para esta seleção.

Devido a crescente importância da precificação futura de *commodities*, em especial a soja, a série diária de preços deste ativo no Estado do Paraná foi escolhida para a aplicação do algoritmo proposto como método de previsão de séries temporais.

Os resultados deste algoritmo mostraram-se superiores em termos de *MSE* à otimização por AG com representação binária (algoritmo de computação evolucionária mais comumente utilizado), ao treinamento da RN-RBF por descida de encosta (algoritmo freqüentemente utilizado para treinamento destas redes), e ao ARIMA (método linear mais popular na previsão de séries temporais) utilizando-se o mesmo critério de desempenho em termos do *MSE*.

5.1 Limitações

Os resultados deste trabalho são específicos para as séries utilizadas como objeto desta pesquisa. Portanto, não necessariamente estes resultados podem ser generalizados, uma vez que os fatores que influenciam o mercado das *commodities* estudadas neste trabalho são diferentes dos fatores relacionados a outros tipos de séries temporais, sejam de outras *commodities*, outros ativos financeiros ou séries não-financeiras.

Isto não quer dizer que o modelo proposto não consiga produzir bons resultados para outras séries temporais. O estudo pode servir de base para novos estudos empíricos. Outra limitação desta pesquisa traduz-se na melhor parametrização dos modelos. Neste trabalho, os modelos foram parametrizados com base em algumas simulações prévias dos modelos e na intuição e experiência do autor.

5.2 Recomendações para Elaboração de Trabalhos Futuros

Este trabalho apresentou uma nova combinação de métodos de previsão de séries temporais do preço da soja. Os métodos combinados foram RN-RBF, EKF e ED. Tal combinação foi comparada com métodos tradicionais, variações e hibridizações reconhecidos na literatura. Como trabalho futuro sugere-se comparar esta combinação com outros modelos e variações existentes na literatura.

Podem ser realizadas novas análises comparativas levando-se em conta outros critérios diferentes do MSE , R^2 e tempo de processamento, tais como: critérios computacionais, critérios econométricos, entre outros.

Uma oportunidade seria a aplicação dos métodos estudados neste trabalho a outros tipos de *commodities* do agronegócio ou *commodities* metálicas, outros tipos de ativos financeiros (ações, debêntures, opções de compra e venda, dentre outros), e até mesmo séries temporais não-financeiras, como séries climáticas, meteorológicas, meteo-oceanográficas ou de sinais biológicos.

Com o objetivo de se analisar mais profundamente outras variáveis do mercado financeiro (por exemplo, a volatilidade), além da previsão de preços poderiam ser estudadas a previsão da variância dos preços, ou outras variáveis.

Outra sugestão seria o uso de outras heurísticas e meta-heurísticas para a seleção dos parâmetros de sintonia do EKF, tais como computação quântica, computação molecular,

sistemas imunológicos artificiais, sistemas nebulosos, ou simplesmente o uso de outras estratégias de evolução diferencial que não a rand/1/bin.

Simulações para previsão de médio prazo seria um caminho promissor, tendo em vista a baixa perda de desempenho do modelo de RN-RBF treinada por EKF otimizado por ED quando de sua utilização na previsão “20 passos à frente”. Outras simulações sugeridas para este modelo incluem alterações nas quantidades de médias móveis aplicadas às séries, nas formas de normalização das séries, nos parâmetros geradores das funções e em outros parâmetros do modelo.

Ao invés da aplicação deste modelo somente para previsão, seria interessante aplicá-los também na identificação de sistemas, por exemplo: sistemas mecânicos, térmicos, hidráulicos e elétricos.

Referências

ABELÉM, A. J. G. **Redes neurais artificiais na previsão de séries temporais**. Dissertação (Mestrado em Engenharia Elétrica) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, 1994.

AGUIRRE, L. A. **Introdução à identificação de sistemas: técnicas lineares e não-lineares aplicadas a sistemas reais**. Editora da UFMG, Belo Horizonte, MG, 2000.

ARANTES, M. B.; OLIVEIRA, G. T. S.; SARAMAGO, S. F. P. **Evolução diferencial aplicada à solução de alguns problemas de engenharia de produção**. FAMAT em Revista, Vol. 1, No. 06, p. 48-61, 2006.

ARANTES, M. B.; SARAMAGO, S. F. P. **Aplicação da evolução diferencial na otimização de problemas de engenharia** - In: VI Encontro Interno de Iniciação Científica, 2006, Uberlândia, MG.

BABU, B. V.; MUNAWAR, S. A. **Differential evolution strategies for optimal design of shell-and-tube heat exchangers**. Chemical Engineering Science, Vol. 62, No. 14, p. 3720-3739, 2007.

BAI, L.; GUO, P. **A novel approach to stellar recognition by combining EKF and RBF-net**. Advances in Neural Networks, International Symposium on Neural Networks, Dalian, China, Lecture Notes in Computer Sciences, Vol. 31, No. 73, p. 1002-1007, Springer, London, UK, 2004.

BASTURK, A.; GUNAY, E. **Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm**. Expert Systems with Applications, 2008.

BHAT, T. R.; VENKATARAMANI, D.; RAVI, V.; MURTY, C. V. S. **An improved differential evolution method for efficient parameter estimation in biofilter modeling.** Biochemical Engineering Journal, Vol. 28, No. 2, p. 167-176, 2006.

BILLINGS, S. A.; XONG, X. **Dual-orthogonal radial basis function networks for nonlinear time series prediction.** Neural Networks, Vol. 11, No. 3, p. 479-493, 1998.

BIRGMEIER, M. **A fully Kalman-trained radial basis function network for nonlinear speech modeling.** Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, Vol. 1, No. 18, p. 259-264, 1995.

BISHOP, C. M. **Neural networks for pattern recognition.** Oxford University Press, Oxford, England, 1997.

BITENCOURT JUNIOR, H. **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, MG, 2003.

BOX, G.; JENKINS, G. **Time series analysis: forecasting and control.** Holden Day: San Francisco, CA, USA, 1976.

BRAGA, P. C. A. S. **Previsão do IBOVESPA utilizando modelos híbridos.** Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2006.

BROCK, W.; DECHERT, W.; SCHEINKMAN, J. **A test for independence based upon the correlation dimension.** University of Wisconsin-Madison, Wisconsin, USA, 1987.

BROOMHEAD, D. S.; LOWE, D. **Multivariable functional interpolation and adaptive networks.** Complex Systems, Vol. 2, No. 1, p. 321--355, 1988.

BUHMANN, M. D.; EBRARY, I. **Radial basis functions theory and implementations.** Cambridge University Press, Cambridge, New York, USA, 2003.

CAMPBELL, J.; LO, A.; MACKINLAY, A. **The econometrics of financial markets**. 1. ed. Princeton University Press, Princeton, USA, 1997.

CASTRO, M. C. F. **Predição não-linear de séries temporais usando redes neurais RBF por decomposição em componentes principais**. Tese (Doutorado em Engenharia Elétrica) – Universidade Estadual de Campinas, Campinas, SP, 2001.

CASTRO, L. N. **Fundamentals of natural computing: an overview**. Physics of Life Reviews, Vol. 4, No. 1, p. 1-36, 2007.

CEPEA <http://www.cepea.esalq.usp.br/> [Acesso em 06 de Junho de 2008].

CHEN, S.; COWAN, C.; GRANT, P. **Orthogonal least squares learning algorithm for radial Basis function networks**, IEEE Transactions on Neural Networks, Vol. 2, No. 2, p. 302-309, 1991.

CHEN, A. S.; LEUNG, M. T.; DAOUK, H. **Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Sotck Index**. Computers & Operations Research, Vol. 30, No. 6, p. 901-923, 2003.

CIOCOIU, I. B. **RBF networks training using a dual extended Kalman filter**. Neurocomputing, Vol. 48, No. 1-4, p. 609-622, 2002.

COELHO, L. S. ; MARIANI, V. C. **Sistema híbrido neuro-evolutivo aplicado ao controle de um processo multivariável**. Sociedade Brasileira de Automática, Vol. 17, No. 17, p. 32-48, 2006.

CORTEZ, P. A. R. **Algoritmos genéticos e redes neuronais na previsão de séries temporais**. Dissertação (Mestrado em Informática) – Universidade do Minho, Braga, Portugal, 1997.

CORTEZ, P. A. R. **Modelos inspirados na natureza para a previsão de séries temporais**. Tese (Doutorado em Informática) – Universidade do Minho, Braga, Portugal, 2002.

DELURGIO, S. A. **Forecasting principles and applications**. 1st Edition. McGraw-Hill. Singapore, 1998.

DOGANIS, P.; ALEXANDRIDIS, A.; PATRINOS, P.; SARIMVEIS, H. **Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing**. Journal of Food Engineering, Vol. 75, No. 2, p. 196-204, 2005.

DU, J. X.; HUANG, D. S.; WANG, X. F.; GU, X. **Shape recognition based on neural networks trained by differential evolution algorithm**. Neurocomputing, Vol. 70, No. 35, p. 896-903, 2007.

EICKHOFF, R.; RÜCKERT, U. **Robustness of radial basis functions**. Neurocomputing, Vol. 70, No. 16-18, p. 2758-2767, 2007.

EHLERS, R. S. **Análise de Séries Temporais**. 4^a ed., Laboratório de Estatística e Geoinformação, Departamento de Estatística, UFPR, Curitiba, PR, 2007. Disponível em: <<http://leg.ufpr.br/~ehlers/notas/stemp.pdf>>. Acesso em: 20 de junho de 2007.

FALCO, G. P. **Técnicas univariadas aperfeiçoadas para previsão de curtíssimo prazo a partir de dados horários**. Dissertação (Mestrado em Engenharia Elétrica) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, 2005.

FALCONE, M. A. G. **Estudo comparativo entre algoritmos genéticos e evolução diferencial para otimização de um modelo de cadeia de suprimento simplificada**. Dissertação (Mestrado em Engenharia de Produção e Sistemas) - Pontifícia Universidade Católica do Paraná, Curitiba, PR, 2004.

FÉLIX, J. A.; RODRIGUEZ, F. F.; ARTILES, M. D. G.; RIVERO, S. S. **An empirical evaluation of non-linear trading rules**. Studies in Nonlinear Dynamics & Econometrics. Vol. 7, No. 3, p. 1160-1160, Spain, 2003.

FERRAZ, M. I. F.; SÁFADI, T.; LAGE, G. **Uso de modelos de series temporais na previsão de precipitação pluviiais mensais no município de Lavras – MG**. Revista Brasileira de Agrometeorologia, Vol. 7, No. 2, p. 259-267, 1999.

GOLDBERG, D. E. **Genetic algorithms in search, optimization & machine learning**. Addison-Wesley Publishing Company, 1989.

GOMES, D. T. **Redes neurais recorrentes para previsão de séries temporais de memórias curta e longa**. 2005. Dissertação (Mestrado em Estatística) – Universidade Estadual de Campinas, Campinas, SP, 2005.

GORDON, N. J.; SALMOND, D. J.; SMITH, A. F. M. **Novel approach to nonlinear/non-Gaussian Bayesian state estimation**. IEEE Proceedings, Vol. 140, No. 2, p. 107-113, 1993.

GREWAL, M. S.; ANDREWS, A. P. **Kalman filtering: theory and practice**. Prentice Hall, 1993.

GROTHMANN, R. **Multi-Agent Market Modeling Based on Neural Networks**. PhD Thesis, Faculty of Economics, University of Bremen, Germany, 2002.

GUERRA, F. A. **Análise de métodos de agrupamento para o treinamento de redes neurais de base radial aplicadas à identificação de sistemas**. Dissertação (Mestrado em Engenharia de Produção e Sistemas) - Pontifícia Universidade Católica do Paraná, Curitiba, PR, 2006.

HARDY, R. L. **Multiquadric equations of topography and other irregular surfaces**. Journal of Geophysics Research, Vol. 76, No. 8, p. 1905–1915, 1971.

HAYKIN, S. S. **Adaptive filter theory**. 3rd ed., Prentice Hall, Upper Saddle River, NJ, USA, 1996.

HAYKIN, S. S. **Redes neurais: princípios e prática**. 2^a ed. Porto Alegre, RS: Bookman, 2001.

HAYKIN, S. S. **Kalman filtering and neural network**. John Wiley & Sons, Inc, NY, USA, 2001.

HEITKOETTER J.; BEASLEY D. **The Hitch-Hiker's guide to evolutionary computation: A list of frequently asked questions.** USENET:comp.ai.genetic. Available via anonymous EMAIL from rtfm.mit.edu:/pub/usenet/news.answers/ai-faq/genetic/, 1994.

HRSTKA, O.; KUCEROVÁ, A.; LEPS, M.; ZEMAN, J. **A competitive comparison of different types of evolutionary algorithms.** Computer and Structures, Vol. 81, No. 18, p. 1979-1990, 2003.

HU, X.; PROKHOROV, D. V.; WUNSCH II, D. C. **Time Series Prediction with a Weighted Bidirectional Multi-Stream Extended Kalman Filter.** Neurocomputing, Vol. 70, No. 13-15, p. 2392-2399, 2007.

HUTCHINSON, J. M. **A radial basis function approach to financial time series analysis.** PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA, 1994.

ILKIVOVÁ, M. R.; ILKIV, B. R.; NEUSCHL, T. **Comparison of a linear and nonlinear approach to engine misfires detection.** Control Engineering Practice, Vol. 10, No. 10, p. 1141-1146, 2002.

INTRILIGATOR, M. **Economic and econometric models.** In: INTRILIGATOR, M., GRILICHES, Z. Handbook of Econometrics, 1st ed, Vol. 3, No. 1, North Holland, 1983.

JUNG, C. F. **Metodologia Para a Pesquisa & Desenvolvimento.** Rio de Janeiro: Axcel Books do Brasil Editora, 2004.

KAASHOEK, J. F.; DIJK, H. K. **Neural network pruning applied to real exchange rate analysis.** Journal of Forecasting, Vol. 21, No. 8, p. 559-577, 2002.

KALMAN, R. E. **A new approach to linear filtering and prediction problems.** Transactions of the ASME – Journal of Basic Engineering, Vol. 83, No. 1, p. 35-45, 1960.

KALMAN, R. E.; BUCY, R. S. **New results in linear filtering and prediction theory.** Transactions of the ASME - Journal of Basic Engineering, Vol. 84, No. 1, p. 95-107, 1961.

KARR, C. **Design of an adaptative fuzzy logic controller using a genetic algorithm.** Proceedings of Fourth Genetic Algorithms Conference, p. 450-457, 1991.

KUAN, C. M.; LIU, T. **Forecasting exchange rates using feedforward and recurrent neural networks.** Journal of Applied Econometrics, Vol. 10, No. 4, p. 347-364, 1995.

KUCHNER, H. J. **Approximations to optimal nonlinear filters.** IEEE Transactions on Automatic Control, Vol. 12, No. 5, p. 546-556, 1967.

LEE, T.; WHITE, H.; GRANGER, C. **Testing for neglected nonlinearity in time series models: a comparison of neural network methods and alternative tests.** Journal of Econometrics. Vol. 56, No. 3, p. 269-290, 1993.

LELAND, W. E.; TAQQU, M. S.; WILLINGER, W.; WILSON, D. V. **On the self-similar nature of Ethernet traffic.** IEEE/ACM Transactions on Networking, Vol. 2, No. 1, p. 1-15, 1994.

MADDALA, G. **Introduction to Econometrics.** Wiley, NY, USA, 2001.

MAKRIDAKIS, S.; WHEELWRIGHT, S. C.; HYNDMAN, R. J. **Forecasting: methods and applications.** Third Edition, John Wiley & Sons, 1998.

MARTIN, C. **Aplicação de redes neurais para prognóstico com base em séries temporais.** Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis, SC, 2000.

MARTIN, C. **Prognóstico de demanda de potência elétrica para planejamento e operação de sistemas elétricos.** Tese (Doutorado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis, SC, 2005.

MATHWORKS, INC. **MATLAB (MATrix LABoratory).** version 7. www.mathworks.com, 2008.

MCCULLOCH, W.; PITTS, W. **A logical calculus of the ideas immanent in nervous activity**. Bulletin of Mathematical Biophysics, Vol. 5, No. 7, p. 115-133, 1943.

MEDSKER, L. A. and BAILEY, D. L. **Models and guidelines for integrating expert systems and neural networks**. In Hybrid architectures for intelligent systems, ed. Kandel and Langholz, CRC Press, 1992.

MORETTIN, P. A.; TOLOI, C. M. C. **Previsão de séries temporais**. 2ª ed. Atual Editora, São Paulo, SP, 1987.

MULGREW, B. **Applying radial basis functions**. IEEE Signal Processing Magazine, Vol. 13, No. 2, p. 50-65, 1996.

NAGELKERKE, N. **A note on a general definition of the coefficient of determination**. Biometrika, Vol. 78, No. 3, p. 691-692, 1991.

NEELY, C.; WELLER, P. **Intraday technical trading in foreign exchange market**. Journal of International Money and Finance, Vol. 22, No. 2, p. 223-237, 2003.

NYGREN, K. **Stock prediction – a neural network approach**. Master Thesis, Royal Institute of Technology, Kungliga Tekniska Högskolan, Stockholm, Sweden, 2004.

OXMETRICS, INC. **OX Pack** version 3.31. www.oxmetrics.net, 2003.

PEREIRA, G. A. S.; AGUIRRE, L. A. **Filtro de Kalman: teoria e aplicações**. Relatório Técnico. Programa de Pós-Graduação em Engenharia Elétrica, UFMG, Belo Horizonte, MG, 2000.

PLUMMER, E. A. **Time series forecasting with feed-forward NN guidelines and limitations**. Master Thesis, Department of Computer Science, University of Wyoming, Laramie, Wyoming, USA, 2000.

PROTIL, R.; COELHO, L. S.; SILVA, W. V. **Previsão não-linear dos preços de troncos de eucalipto baseada em uma abordagem neuroevolutiva**. Gestão & Produção, Vol. 14, No. 1, São Carlos, SP, 2007.

PUSKORIUS, G. V.; FELDKAMP, L. A. **Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks**. IEEE Transactions on Neural Networks, Vol. 5, No. 2, p. 279-297, 1994.

ROCHA, F. L. **Identificação de sistemas não-lineares multivariáveis usando redes neurais perceptron multicamadas e função de base radial**. Dissertação (Mestrado em Engenharia de Produção e Sistemas) - Pontifícia Universidade Católica do Paraná, Curitiba, PR, 2006.

ROSENBLATT, F. **Principles of Neurodynamics**. Spartan Books, 1962.

SANTOS, A. A. P. **Previsão não-linear da taxa de câmbio real/dólar utilizando redes neurais e sistemas nebulosos**. Dissertação (Mestrado em Economia) – Universidade Federal de Santa Catarina, Florianópolis, SC, 2005.

SCARPEL, R. A. **Utilização de support vector machine em previsão de insolvência de empresas**. In: Anais do XXXVII SBPO – Simpósio Brasileiro de Pesquisa Operacional, Gramado, RS, 2005.

SCHAUER, T.; NEGARD, N. O.; PREVIDI, F.; HUNT, K. J.; FRASER, M. H.; FERCHLAND, E.; RAISCH, J. **Online identification and nonlinear control of electrically stimulated quadriceps muscle**. Control Engineering Practice, Vol. 13, No. 9, p. 1207-1219, 2005.

SILVA, W. V. ; SOUZA, A. M. ; KRÜGER, E. R. ; JACOBI, L. F. **Avaliação dos retornos de escala no setor petroquímico baiano nos anos de 2002 e 2003: um estudo econométrico usando dados cross section**. In: 16º Simpósio Nacional de Probabilidade e Estatística, 2004, Caxambu, MG. 16º SINAPE. São Paulo: Editora da ABE, 2004. Vol. 16. p. 1-5.

SIMON, D. **Training radial basis neural networks with the extended Kalman filter.** Neurocomputing, Vol. 48, No. 1-4, p. 255-475, 2002.

SKOTT, P. **Conflict and effective demand in economic growth.** Cambridge University, Cambridge, England, 1989.

SMITH, K. A.; GUPTA, J. N. D. **Neural networks in business: techniques and applications for the operations researcher.** Computer & Operations Research, Vol. 27, No. 1, p. 1023-1044, 2000.

SORENSEN, H. W. **Kalman filtering: theory and application.** IEEE, New York, NY, USA, 1985.

SOUZA, R. C. T.; COELHO, L. S. **Rede neural RBF com treinamento baseado em filtro de Kalman e evolução diferencial aplicada à previsão do preço da soja.** In: Anais do VIII Congresso Brasileiro de Redes Neurais, Florianópolis, SC, 2007.

STATSOFT, INC. **STATISTICA (data analysis software system),** version 7.1. www.statsoft.com, 2005.

STOCK, J.; WATSON, M. **A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series.** NBER working paper, Cambridge, USA, Vol. 66, No. 7, 1999.

STORN, R.; PRICE, K. **Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces.** Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA, USA, 1995.

STORN, R.; PRICE, K. **Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces.** Journal of Global Optimization, Vol. 11, No. 4, p. 341-359, 1997.

SWANSON, N.; WHITE, H. **A model-selection approach to assessing the information in the term structure using linear models and artificial neural networks.** Journal of Business & Economic Statistics, Vol. 13, No. 3, p. 265-275, 1995.

VAN GORP, J. **Nonlinear identification with neural networks and fuzzy logic.** Thesis, Vrije Universiteit Brussel, Faculteit Toegepaste Wetenschappen, Brussel, Belgium, 2000.

WEIGEND, A. S.; HUBERMAN, B. A.; RUMELHART, D. E. **Predicting the future: a connectionist approach.** International Journal of Neural Systems, Vol. 1, No. 3, p. 193-209, 1990.

WEIGEND, A. S.; GERSHENFELD, N. A. **Time series prediction: forecasting the future and understanding the past.** Addison-Wesley Publishing Company, Menlo Park, CA, USA, 1994.

WELCH, G.; BISHOP, G. **An introduction to the Kalman filter.** Technical Report TR-95-041, Department of Computer Science, University of North Carolina at Chapel Hill, NC, USA, 1995.

WERNER, L.; RIBEIRO, J. L. D. **Previsão de demanda: uma aplicação Box-Jenkins na área de assistência técnica de computadores pessoais.** Gestão & Produção, Vol. 10, No. 1, p. 47-67, 2003.

XIONG, S. S.; ZHOU, Z. Y. **Neural filtering of colored noise based on Kalman filter structure.** IEEE Transactions on Instrumentation and Measurement, Vol. 52, No. 3, p. 742-747, 2003.

YAO, X. **Evolving artificial neural networks.** Proceedings of IEEE, Vol. 87, No. 9, p. 1423-1439, 1999.

ZADEH, L. A. **Fuzzy sets.** Information and Control. Vol. 8, No. 3, p. 338-353, 1965.

ZHANG, L.; LUH, P. B. **Neural network-based market clearing price prediction and confidence interval estimation with an improved extended Kalman filter method.** IEEE Transactions on Power Systems, Vol. 20, No. 1, p. 59-66, 2005.