UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ELISEU CELESTINO SCHOPF

Método Neuro-estatístico para Predição de Séries Temporais Ruidosas

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Paulo Martins Engel
Orientador

Porto Alegre, julho de 2007

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Schopf, Eliseu Celestino

Método Neuro-estatístico para Predição de Séries Temporais Ruidosas / Eliseu Celestino Schopf. – Porto Alegre: PPGC da UFRGS, 2007.

105 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2007. Orientador: Paulo Martins Engel.

1. Inteligência artificial. 2. Redes neurais artificiais. 3. Métodos estatísticos. 4. Filtro de Kalman Estendido. 5. Predição de séries temporais. 6. Ruído. I. Engel, Paulo Martins. II. Título.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

AELB	Alisamento Exponencial Linear de Brown
AEQB	Alisamento Exponencial Quadrático de Brown
AES	Alisamento Exponencial Simples
AR	Auto-regressão
BP	*Back-propagation* - Algoritmo da retropropagação
DCBD	Descoberta de Conhecimento em Base de Dados
fdp	Função de densidade de probabilidade, para transição de estados
FK	Filtro de Kalman
FKD	Filtro de Kalman Discreto
FKE	Filtro de Kalman Estendido
FKED	FKE Disjunto
GRNN	Rede Neural de Regressão Geral
IMM	Interação com Múltiplos Modelos, técnica que utiliza múltiplos Filtros de Kalman
MD	Mineração de Dados
MEE	Modelos de Espaço de Estados
MLD	Modelos Lineares Dinâmicos
MLP	*Multi Layer Perceptron* - Perceptron de Múltiplas Camadas
MMS	Médias Móveis Simples
MSE	*Minimum Square Error* - Erro Mínimo Quadrado
NARX	*Nonlinear Auto-regressive with Exogenous Input* - Modelo Auto-regressivo Não-linear com Entradas Exógenas
NAR	*Nonlinear Auto-regressive* - Modelo Auto-regressivo Não-linear
NOE	*Nonlinear Output Error* - Modelo regressivo correspondente ao NARX
NDEKF	*Node-decoupled Extend Kalman Filter* - FKE Disjunto, com os pesos acoplados por nós
NE	Método Neuro-estatístico

NEKF *Neural Extended Kalman Filter* - Filtro de Kalman Estendido com rede neural acoplada

PLE Processos Lineares Estacionários

PLNEH Processos Lineares Não-estacionários Homogêneos

PML Processos de Memória Longa

PST Predição de Séries Temporais

RBF *Radial Basis Function* - Funções de Base Radial, um modelo de rede neural

RBNN *Regular Radial Basis Neural Networks* - rede neural de base radial regular

RN Rede Neural. No contexto deste trabalho significa RNA

RNA Rede Neural Artificial

SOM *Self-Organizing Maps* - Mapas Auto-organizáveis

TLFN *Focused Time Lagged Feedforward Network* - Redes alimentadas adiante focadas atrasadas no tempo

TDNN *Time-Delay Neural Networks* - Redes recorrentes atrasadas no tempo

UKF *Unscented Kalman Filter* - Filtro de Kalman *Unscented*, uma variação do Filtro de Kalman Estendido

VAD Variável Aleatória Discreta

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

O presente trabalho trata da criação de uma nova abordagem para predição de séries temporais ruidosas, com modelo desconhecido e que apresentam grandes não-linearidades. O novo método neuro-estatístico proposto combina uma rede neural de múltiplas camadas com o método estatístico Filtro de Kalman Estendido. A justificativa para a junção dessas abordagens é o fato de possuírem características complementares para o tratamento das peculiaridades das séries descritas. Quanto ao ruído, o FKE consegue minimizar a sua influência, trabalhando com a variância do ruído extraído dos dados reais. Quanto ao modelo gerador da série, as redes neurais aproximam a sua função, aprendendo a partir de amostras dos próprios dados. Grandes não-linearidades também são tratadas pelas RNs. O método neuro-estatístico segue a estrutura do FKE, utilizando a RN como processo preditivo. Com isso, elimina-se a necessidade de conhecimento prévio da função de transição de estados. O poder de tratamento de não-linearidades da RN é mantido, utilizando-se a previsão desta como estimativa de estado e os seus valores internos para cálculo das jacobianas do FKE. As matrizes de covariâncias dos erros de estimativa e dos ruídos são utilizadas para melhora do resultado obtido pela RN. A rede é treinada com um conjunto de dados retirado do histórico da série, de maneira *off-line*, possibilitando o uso de poderosas estruturas de redes de múltiplas camadas. Os resultados do método neuro-estatístico são comparados com a mesma configuração de RN utilizada em sua composição, sendo ambos aplicados na série caótica de Mackey-Glass e em uma série combinada de senos. Ambas séries possuem grandes não-linearidades e são acrescidas de ruído. O novo método alcança resultados satisfatórios, melhorando o resultado da RN em todos os experimentos. Também são dadas contribuições no ajuste dos parâmetros do FKE, utilizados no novo método. O método híbrido proporciona uma melhora mútua entre a RN e o FKE, explicando os bons resultados obtidos.

Palavras-chave: Inteligência artificial, redes neurais artificiais, métodos estatísticos, Filtro de Kalman Estendido, predição de séries temporais, ruído.

Neural Statistical Method to Noisy Time Series Prediction

ABSTRACT

This work presents a new forecast method over highly nonlinear noisy time series. The neural statistical method uses a multi-layer perceptron (NN) and the Extended Kalman Filter (EKF). The justification for the combination of these approaches is that they possess complementary characteristics for the treatment of the peculiarities of the series. The EKF minimizes the influence of noise, working with the variance of the noise obtained from the real data. The NN approximates the generating model's function. High nonlinearities are also treated by the neural network. The neural statistical method follows the structure of the EKF, using the NN as the predictive process. Thus, it isn't necessary previous knowledge of the state transition function. The power of treatment of nonlinearities of the NN is kept, using forecast of this as estimative of state and its internal values for calculation of the Jacobian matrix of the EKF. The error estimative covariance and the noise covariance matrixes are used to improve the NN outcome. The NN is trained off-line by past observations of the series, which enable the use of powerfuls neural networks. The results of the neural statistical method are compared with the same configuration of NN used in its composition, being applied in the chaotic series of Mackey-Glass and an sine mistures series. Both series are noisy and highly nonlinear. The new method obtained satisfactory result, improving the result of the regular NN in all experiments. The method also contributes in the adjustment of the parameters of the EKF. The hybrid method has a mutual improvement between the NN and the EKF, which explains the obtained good results.

Keywords: artificial intelligence, artificial neural networks, statistical methods, Extended Kalman Filter, time series prediction, noise.

1 INTRODUÇÃO

A descoberta de conhecimento em bases de dados (DCBD) e a mineração de dados (MD), sua principal componente, despertam interesse de várias áreas como aprendizado de máquina, reconhecimento de padrões, estatística e inteligência artificial (YEE; JIANG-HONG; WEN-XIU, 2001). A mineração de dados situa-se na zona de sobreposição entre estatística e ciência da computação, utilizando os avanços de ambas para melhorar a extração de informações de bases de dados. Isso indica que trabalhos que unam as duas áreas, como a criação de um método híbrido, podem ser muito proveitosos.

Tanto a mineração de dados como a estatística procuram aprender a partir dos dados, transformando dados em informação. Existe apenas uma diferença de ênfase, pois a mineração de dados envolve análise retrospectiva e está mais voltada para a compreensão do que para a precisão. Práticas atuais de mineração de dados estão mais focadas em padrões, deixando a modelagem em segundo plano. A tarefa de construir um modelo global e coerente fica para a estatística (GLYMOUR et al., 1996). A atividade da regressão (predição) consiste em aproximar saídas quantitativas. Na regressão, o grande objetivo é ter a melhor precisão possível nas predições. Torna-se mais difícil atingir esse objetivo quando se tem a tarefa da regressão em dados ruidosos.

Em aplicações com dados reais, as observações sempre estarão sujeitas a erros, fazendo com que as bases de dados sejam em sua maioria ruidosas. Ruídos são pequenas variações ou incertezas nos dados. Devido à presença de ruído, inferências em bases de dados atraem aplicações da teoria da probabilidade. Algumas técnicas estatísticas, como um filtro linear ótimo, conseguem minimizar a influência do ruído, trabalhando com a variância dos dados do modelo, como no Filtro de Kalman (FK) (KALMAN, 1960). Para isso é necessário possuir um modelo analítico ou criar uma abordagem totalmente explicitada, em que se possui a formulação matemática do modelo real. Como muitos modelos de sistemas reais não são conhecidos, torna-se inviável a predição de variáveis desses sistemas com um método estatístico. A presença de não-linearidades nas funções geradoras dos sistemas também é outro fator complicador. As expressões do modelo tornam-se equações matemáticas muito complexas, sendo equações de regressão (PAYLE, 1999).

Outra técnica concorrente, as redes neurais (RN) possuem grande poder computacional devido à sua estrutura maciçamente paralela e distribuída e de sua capacidade de aprender para generalizar (HAYKIN, 2001a). O tratamento de não-linearidades é uma característica muito importante, tornando as redes neurais mais poderosas. Essa característica se torna ainda mais útil (e necessária) quando se trata de dinâmicas não-lineares (ou dados inspirados em sinais não-lineares). A não-linearidade de uma rede neural é de um tipo especial, pois está presente em cada neurônio. Outra vantagem do uso de redes neurais é o emprego de aprendizado supervisionado. Esse paradigma permite o ajuste de parâmetros a partir de amostras ou exemplos rotulados. Esse ajuste é de maneira gradual,

semelhante à inferência estatística não-paramétrica, não sendo feitas suposições prévias sobre o modelo estatístico dos dados de entrada.

Uma rede neural também possui adaptabilidade em relação a modificações no ambiente. Em ambientes não-estacionários (características variam ao longo do tempo) as RNs também podem ser treinadas para adaptar-se (modificando seus pesos sinápticos) em tempo real. As redes neurais também são tolerantes a falha, uma vez que a falha de um neurônio apenas prejudica a qualidade da solução, mas não causa a falha total dessa solução. Além de as RNs fornecerem as informações sobre um padrão, também podem indicar a confiança na decisão tomada (HAYKIN, 2001a). Devido às vantagens das RNs, é viável construir um sistema que mantenha essas características, contendo também a modelagem do ruído, presente em métodos estatísticos de filtragem linear ótima, como o Filtro de Kalman.

Uma das grandes aplicações de métodos estatísticos e de redes neurais é a predição de séries temporais. As séries temporais são usadas para descrever variáveis de sistemas reais, tendo a previsão destas grande utilidade na economia, medicina, engenharias, meio ambiente e inúmeras outras áreas. Redes neurais são utilizadas com sucesso na predição de séries temporais desde a década de 70. O uso das redes foi muito estimulado e justificado pelos brilhantes resultados na primeira competição STI (WAN, 1994). Os métodos baseados em redes neurais foram os grandes vencedores, obtendo melhores resultados que métodos consagrados de regressão.

As próprias características das séries temporais propiciam o uso de redes neurais, (WAN, 1994) indica a existência de uma não-linearidade na definição das séries. A estrutura das RNs também é um fator decisivo, sendo que uma RN com múltiplas camadas alimentada adiante, com um número suficiente de neurônios, é considerada aproximador universal de funções (CYBENKO, 1989).

As RNs costumam apresentar bons resultados para predição de séries não-lineares e desconhecidas. Porém, a predição de séries ruidosas é algo pouco explorado com essas técnicas. O acréscimo de ruído em séries já bastante complexas dificulta muito a predição por parte das RNs, treinadas com esses dados ruidosos. Nesses casos, a RN apresenta dificuldades em identificar o que é a série original e o que é ruído.

As séries temporais não-lineares, ruidosas e com modelos desconhecidos são as mais abundantemente retiradas de sistemas reais. Porém, séries com todas essas características simultaneamente são pouco tratadas na literatura. Nas aplicações de redes neurais, no máximo utilizam-se séries com função geradora complexa (com grandes não-linearidades) desconhecida. Essas dificuldades já servem como desafio, beirando a capacidade das redes. Nas aplicações com métodos estatísticos, como o Filtro de Kalman, não são utilizados sistemas com modelo totalmente desconhecido e altamente não-linear, pois o método necessita possuir uma função representando o modelo gerador do sistema.

Para que todas as dificuldades apresentadas acima na predição de séries temporais possam ser tratadas concomitantemente, sugere-se a criação de uma abordagem híbrida (neural e estatística). Nessa abordagem visa-se manter toda a capacidade das RNs para implementar modelos complexos e desconhecidos. Adiciona-se a essas características, a capacidade de modelagem do ruído, por parte do Filtro de Kalman. Com isso, objetiva-se melhorar o resultado da RN a cada passo de predição. Mais especificamente, visa-se criar um método neuro-estatístico com as seguintes características:

- Capacidade de predição de séries temporais;
- Robustez a ruído, minimizando a influência deste;

- Obtenção de menores taxas de erro que uma RN com mesma estrutura atuando isoladamente;
- Maior aplicabilidade que o Filtro de Kalman, não necessitando da função do modelo gerador;
- Interação do FK com a RN, com cada método passando resultados melhorados para o outro, a cada passo de predição;
- Realismo: receber apenas dados ruidosos para treinamento e medidas; não necessitar conhecer previamente o modelo ideal, parâmetros ideais e outras informações que não são normalmente disponíveis na prática.

A dissertação está estruturada da seguinte forma: no capítulo 2 são mostradas as redes neurais, utilizadas na criação do novo método, sendo explicadas suas vantagens, o processo de aprendizagem e seus usos; no capítulo 3 é descrito o Filtro de Kalman, método no qual este trabalho também é baseado; no capítulo 4 é feita a revisão bibliográfica sobre a predição de séries temporais, comentando e comparando técnicas; o capítulo 5 trata dos métodos correlacionados; no capítulo 6 o novo método neuro-estatístico é apresentado, com a apresentação do modelo e sua explicação formal; o capítulo 7 apresenta os experimentos e resultados e o capítulo 8 mostra as considerações finais, com conclusões e sugestões de trabalhos futuros.

2 REDES NEURAIS

As Redes Neurais (RNs) formam um importante paradigma computacional, envolvem diversas áreas e utilizam conhecimento extraído a partir da experiência. Este capítulo trata do funcionamento das redes e dos conceitos, estruturas e aplicações envolvendo esse paradigma. O capítulo é composto por: conceitos e estruturas das RNs; o processo pelo qual as redes aprendem, mostrando os tipos de aprendizado; as tarefas básicas realizadas pelas redes neurais e as aplicações geradas com a execução das suas tarefas em diversas situações, enfocando os usos relacionados com este trabalho, como processamento temporal e identificação de sistemas.

2.1 Conceitos

Para uma melhor compreensão das bases e do funcionamento das RNs, esta seção abordará os conceitos necessários para a sua explicação. Serão apresentadas as redes neurais, com a sua importância; o funcionamento do neurônio artificial e a definição das funções de ativação, utilizadas nas RNs.

2.1.1 O que são Redes Neurais

O estudo de Redes Neurais tem sido motivado pelas diferenças entre o funcionamento do cérebro humano e o de um computador digital tradicional. O cérebro humano é um sistema de processamento altamente complexo, não-linear e paralelo por natureza. A organização do cérebro permite que sejam realizadas certas computações (como o reconhecimento de padrões, controle sensório-motor e percepção) de maneira mais rápida e precisa que os mais poderosos computadores. Por exemplo, uma pessoa consegue reconhecer um rosto familiar em uma cena não familiar demorando apenas uma fração de segundo, enquanto um computador convencional levaria horas ou dias para resolver uma versão simplificada desse problema. A razão dessa grande capacidade do cérebro é a habilidade de desenvolver suas próprias regras, moldando os neurônios e criando o que é chamado de "experiência". Os neurônios possuem grande plasticidade, o que permite que o cérebro em desenvolvimento adapte-se ao ambiente.

As redes neurais utilizam características de adaptação do cérebro humano e podem ser consideradas máquinas para modelar a maneira como o cérebro aprende uma tarefa ou função. As RNs são constituídas de neurônios artificiais e podem ser construídas com componentes eletrônicos, ou simuladas com computadores digitais. As RNs alcançam bom desempenho através da interligação maciça de neurônios artificiais, utilizando um processo de aprendizagem sobre eles. Uma RN pode ser definida como um processador maciçamente distribuído e paralelo, constituído de unidades simples de processamento,

com a propensão natural de transformar conhecimento experimental em conhecimento pronto para uso (HAYKIN, 2001a).

As RNs assemelham-se ao cérebro no sentido em que o conhecimento é adquirido do ambiente pela rede a partir do seu processo de aprendizagem e pela existência das forças de conexão entre os neurônios (pesos sinápticos), utilizadas para armazenar o conhecimento adquirido. O procedimento pelo qual as RNs aprendem é chamado de *algoritmo de aprendizagem* e serve para modificar os pesos sinápticos da rede, treinando a rede para que a mesma trabalhe (reconhecendo os padrões) da forma para a qual foi projetada. O aprendizado por modificação dos pesos é a forma tradicional pela qual as RNs são projetadas.

As RNs são comumente classificadas de várias formas: como subespecialidade da inteligência artificial; como uma classe de modelos matemáticos para classificação e reconhecimento de padrões; como parte da teoria conexionista de estados mentais ou como categoria de modelos em ciência da cognição (KOVÁCS, 2002). Embora as RNs sejam relacionadas com todas essas categorias, seria muito limitante classificá-las em apenas um desses setores. As RNs formam hoje uma teoria genuína para o estudo de fenômenos complexos. No que se refere à estrutura, as redes neurais possuem várias classificações:

- Quanto ao número de camadas, as redes podem ser *Redes de Camada Única* ou *Redes de Múltiplas Camadas* (com a existência de uma ou mais *camadas ocultas*, ou intermediárias);

- Quanto à conectividade, as redes podem ser *totalmente conectadas* ou *parcialmente conectadas*. Em uma rede totalmente conectada (como na figura 2.5), cada neurônio possuirá ligações com todos os neurônios da camada seguinte. Quando a rede for parcialmente conectada, algumas dessas conexões não existirão. Em grande parte das aplicações as redes apresentam conectividade alta (totalmente ou quase totalmente conectadas);

- Quanto à maneira como os sinais se propagam dentro da rede (se a rede possui retroalimentação ou não) as RNs podem ser classificadas em *alimentadas adiante* ou *recorrentes*. Nas redes alimentadas adiante, o fluxo de sinal é apenas em um sentido, como na figura 2.5. Nas redes recorrentes, existe pelo menos um ciclo de retroalimentação, em que o sinal retorna para uma camada anterior. As redes recorrentes são muito utilizadas em processamento temporal, em que a retroalimentação serve para armazenar entradas de tempos anteriores e colocá-las novamente na entrada nos instantes seguintes.

2.1.2 Importância das Redes Neurais

As RNs possuem a capacidade de generalizar informações, calculando saídas adequadas para entradas que não estavam presentes no arquivo de treinamento. As RNs ainda estão distantes de simularem um cérebro humano inteiro e trabalham apenas com subconjuntos de tarefas. Mesmo assim, atualmente as redes neurais já se apresentam com grande destaque nas atividades que eram há pouco tempo essencialmente do cérebro e geram expectativa de grandes avanços nas próximas décadas. As principais vantagens que o uso de RNs possibilita são (HAYKIN, 2001a):

Não-linearidade As RNs podem ter neurônios lineares ou não-lineares, a rede que possui ao menos um neurônio não-linear é considerada não-linear. A não-linearidade das RNs é de um tipo especial, distribuída por toda a rede;

Mapeamento de Entrada-Saída As RNs podem aprender através de exemplos, a partir de amostras rotuladas utilizadas no treinamento da rede. Assim a rede aprende com os exemplos a construir um mapeamento de entrada-saída para o problema considerado;

Adaptabilidade As redes possuem uma capacidade natural de adaptação dos pesos de seus neurônios de acordo com modificações no ambiente, podendo serem facilmente retreinadas. Também existem projetos de redes que conseguem adaptar os seus pesos em tempo real, para trabalharem em ambientes não-estacionários;

Informação Contextual A informação contextual é tratada naturalmente por uma RN pois o conhecimento é representado pela sua própria estrutura. Cada neurônio é influenciado pela atividade dos outros, formando automaticamente a noção de contexto;

Tolerância a Falhas Se um neurônio ou suas conexões falharem (em implementações de redes físicas) a rede apresenta apenas uma degradação suave, devido à natureza distribuída da informação na rede.

Uniformidade de Análise e Projeto Os neurônios são os processadores univesais de informação nas RNs. Com isso é possível o compartilhamento de algoritmos de aprendizagem em diferentes aplicações de RNs. Também podem ser construídas redes a partir de vários módulos;

Analogia Neurobiológica O estudo em RNs é motivado pela analogia com o cérebro humano. O cérebro é uma grande prova de que o processamento paralelo, tolerante a falhas e adaptativo é, além de possível, muito rápido e poderoso. As pesquisas em RNs visam desde auxiliar as ciências humanas e da saúde no entendimento dos fenômenos cerebrais, até desenvolver idéias para resolver problemas mais complexos que os resolvidos por técnicas tradicionais, auxiliando as ciências exatas e da tecnologia.

2.1.3 Neurônio Artificial

O neurônio artificial é a unidade de processamento básica das redes neurais, sendo uma simplificação do neurônio biológico. Na figura 2.1 é mostrado um modelo de neurônio artificial, podendo-se identificar os seus três elementos básicos:

1. Um conjunto de sinapses, correspondentes às entradas do neurônio k. Na figura, cada uma dessas entradas é a multiplicação de um dos sinais de entrada $(x_1, x_2, \cdots, x_N)$ pelo seu respectivo peso $(w_{k1}, w_{k2}, \cdots, w_{kN})$. No índice dos pesos, o primeiro dígito (k) corresponde ao neurônio de destino e o segundo dígito corresponde ao neurônio de origem do sinal. Os pesos das sinapses dos neurônios artificiais podem incluir também valores negativos. As entradas do neurônio artificial são correspondentes aos dentritos do neurônio biológico.

2. Um somador para computar os sinais ponderados de entrada, constituindo um combinador linear. Esse somador corresponde à membrana celular do neurônio biológico.

3. Uma função de ativação, que restringe a amplitude da saída do neurônio. A saída é restrita ao intervalo $[0,1]$ ou ao intervalo $[-1,1]$, dependendo da função de ativação escolhida. A função de ativação corresponde ao mecanismo de disparo dos potenciais de ação nos axônios do neurônio biológico.

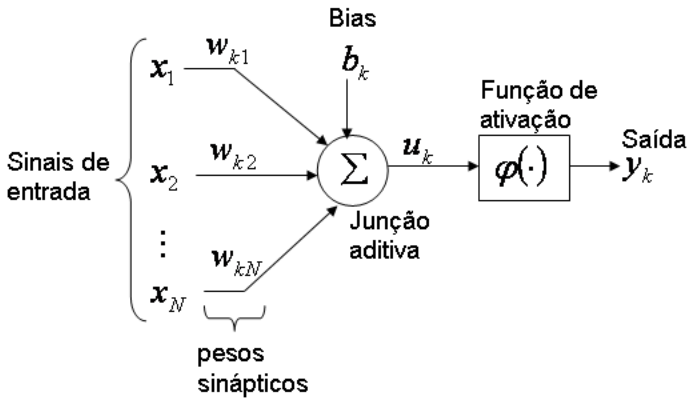

Figura 2.1: Modelo de neurônio artificial

O modelo do neurônio da figura 2.1 também apresenta um *bias* (b_k). O bias, ou viés, é aplicado externamente e tem a função de aumentar ou diminuir a entrada líquida na função de ativação, dependendo se for positivo ou negativo, respectivamente. A soma do neurônio k será:

$$u_k = \sum_{i=1}^{N} w_{ki} x_i + b_k \tag{2.1}$$

Onde cada entrada (x_i) é multiplicada por seu respectivo peso (w_{ki}), formando um somatório. O bias é acrescido diretamente nesse somatório, formando o potencial de ativação do neurônio k. A saída final do neurônio será a função de ativação aplicada sobre esse resultado:

$$y_k = \varphi(u_k) \tag{2.2}$$

2.1.4 Funções de Ativação

A função de ativação de um neurônio artificial ($\varphi(\cdot)$) calcula a saída (restringindo a amplitude) do neurônio, em função do valor do potencial de ativação u_k. As funções de ativação mais conhecidas são a função de limiar, a função linear por partes e as funções do tipo sigmóide: logística e tangente hiperbólica. A figura 2.2 mostra esses quatro tipos de funções de ativação. Atualmente as funções mais utilizadas em redes neurais são as sigmóides.

Função de Limiar Como mostrado na figura 2.2a, representa uma função de decisão abrupta, adaptada à característica binária do neurônio de McCulloch e Pitts (MCCULLOCH; PITTS, 1943), no qual era utilizada. Essa função é expressa por:

$$y_k = \begin{cases} 1 & \text{se} \quad u_k \geq 0 \\ 0 & \text{se} \quad u_k < 0 \end{cases} \tag{2.3}$$

Função Linear por Partes A função linear por partes possui uma ativação linear no intervalo de operação da função e comporta-se como sendo função limiar nos outros

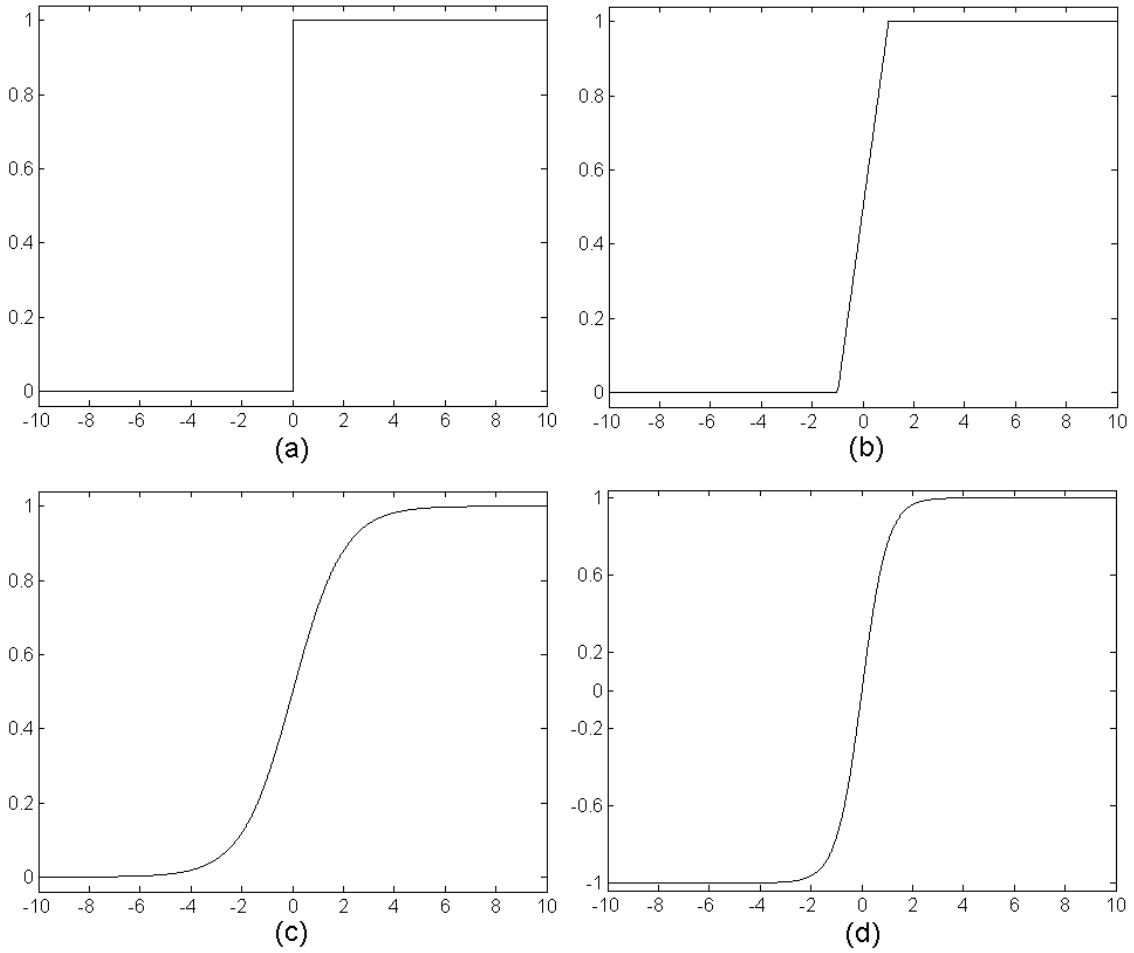

Figura 2.2: Funções de ativação para um neurônio artificial. (a) Função de limiar. (b) Função linear por partes. (c) Função logística. (d) Função tangente hiperbólica

trechos. A função mostrada na figura 2.2b é expressa por:

$$y_k = \begin{cases} 1 & \text{se} \quad u_k \geq 1 \\ \frac{u_k}{2} + 0.5 & \text{se} \quad -1 \leq u_k < 1 \\ 0 & \text{se} \quad u_k < -1 \end{cases} \tag{2.4}$$

Funções Sigmóides As funções sigmóides (em forma de s) são largamente as mais utilizadas e proporcionam um balanceamento entre o comportamento linear e não-linear. Outra vantagem é que as funções sigmóides são diferenciáveis. As funções sigmóides mais utilizadas são:

[Função Logística] Limita a entrada no intervalo $[0,1]$ e é descrita pela função abaixo, onde $\exp(\cdot)$ é a função exponencial.

$$\varphi(u) = \frac{1}{1+\exp(-u)} \tag{2.5}$$

[Função Tangente Hiperbólica] Limita a entrada no intervalo $[-1,1]$:

$$\varphi(u) = \tanh(u) \tag{2.6}$$

2.2 Processos de Aprendizagem em Redes Neurais

Um processo de aprendizagem em uma rede neural permite que a rede aprenda a partir de observações do ambiente, em um processo iterativo de ajustes aplicados aos seus pesos sinápticos, tornando-se apta a exercer sua ação no ambiente (tomada de decisão, previsão, classificação, etc.). O tipo de aprendizagem depende da maneira como os parâmetros livres da rede (pesos sinápticos) são alterados. Essa maneira é descrita por um conjunto bem definido de regras, chamado de algoritmo de aprendizagem. Existe uma grande variedade de algoritmos de aprendizagem, distribuídos pelas diferentes tarefas e aplicações desejadas para a rede. Esses algoritmos são classificados de acordo com o *paradigma de aprendizagem*, isto é, a maneira como a rede se relaciona com o ambiente. De acordo com o tipo de ambiente que a rede recebe, os métodos de aprendizagem podem ser classificados em dois grandes grupos: aprendizado supervisionado e aprendizado não-supervisionado (HAYKIN, 2001a). Esses dois paradigmas serão mostrados nesta seção.

2.2.1 Aprendizado Supervisionado

O aprendizado supervisionado realiza o treinamento da rede a partir de amostras de entrada e saída do sistema. Um conjunto de amostras rotuladas (entradas com sua respectiva saída desejada) representa o conhecimento que se possui inicialmente sobre o ambiente e é comumente chamado de professor. A figura 2.3 mostra o diagrama de blocos do modelo de aprendizado supervisionado. A diferença entre a resposta desejada (fornecida pelo professor) e a resposta do sistema (RN) alimenta novamente o sistema para aprendizado.

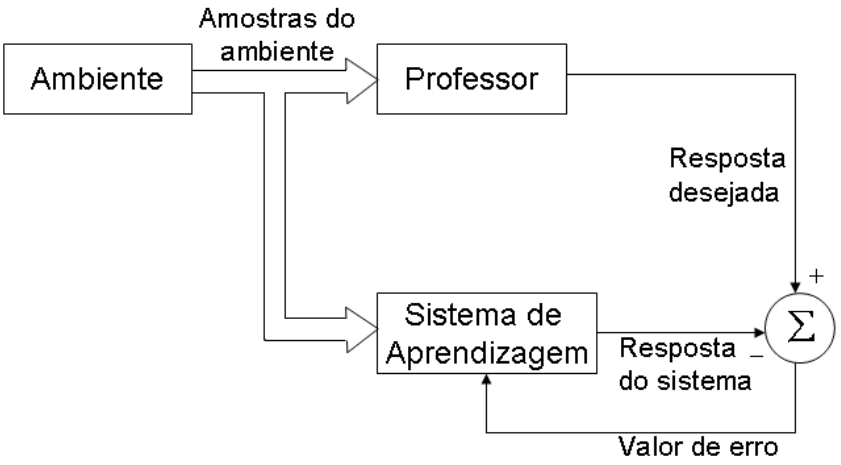

Figura 2.3: Modelo do aprendizado supervisionado

A entrada de cada amostra é passada para a rede para a obtenção de um resultado. Após estimar uma saída para essa entrada, a rede recebe (do professor) o rótulo da amostra (saída desejada). A diferença entre a saída desejada e a obtida pela RN é utilizada para corrigir os pesos da rede. O ajuste da rede é feito passo a passo, iterativamente, até que o conhecimento do professor seja transferido para a rede de maneira satisfatória. Quando a rede possuir uma boa representação do ambiente, pode-se dispensar o professor (dados de treinamento) e deixar a rede trabalhar com novos dados vindos do ambiente.

Uma das estratégias que podem ser usadas na correção dos pesos, no aprendizado supervisionado, é a utilização do *coeficiente de Momentum*. O Momentum deixa a variação dos pesos dependente também das variações passadas, suavizando as oscilações (JORIS,

2005). A utilização do coeficiente de Momentum é muito importante quando (HAYKIN, 2001a):

1. A variação do erro é muito pequena (superfície de descida do erro plana), nesses casos o Momentum acelera a convergência da descida do erro, aumentando o tamanho do passo em direção ao erro mínimo;

2. A variação do erro é muito grande (curvas acentuadas na superfície de descida do erro), nesses casos o Momentum controla a descida do erro, diminuindo a chance de queda em mínimos locais.

2.2.2 Aprendizado Não-supervisionado

No aprendizado não-supervisionado, a rede neural aprende diretamente das características intrínsecas dos dados, sem necessitar de um professor externo ou amostras rotuladas. A rede aprende diretamente do ambiente, como mostrado na figura 2.4, criando automaticamente novas classes. Diferentemente do aprendizado supervisionado, aqui as amostras não são rotuladas. A aprendizagem não-supervisionada é utilizada em tarefas de classificação e detecção de agrupamentos, onde é possível separar as amostras em grupos, levando em consideração apenas as proximidades entre seus atributos.

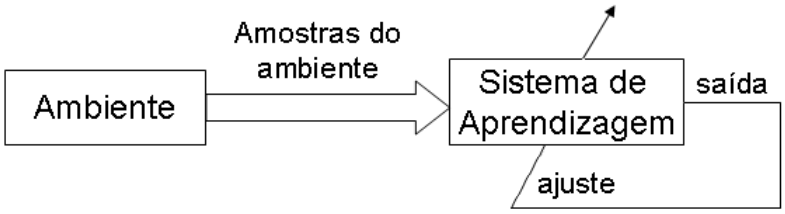

Figura 2.4: Modelo do aprendizado não-supervisionado

Uma das formas mais comuns de aprendizado não-supervisionado é a *regra de aprendizagem competitiva*, através da competição entre neurônios da rede. Por exemplo, pode-se utilizar uma RN de duas camadas: uma de entrada e a outra competitiva. A camada competitiva da rede é composta de neurônios que competem entre si, obedecendo uma regra de aprendizagem, tentando responder às características dos dados de entrada. Cada neurônio convergirá automaticamente para uma certa configuração, sendo que aquele que tiver a maior ativação "dá a vitória" para a sua configuração (classe).

As arquiteturas para aprendizado não-supervisionado normalmente são mais complexas que no caso do aprendizado supervisionado. Nessas arquiteturas, além das ligações para os neurônios da próxima camada, há ligações laterais entre neurônios (da mesma camada), para proporcionar a competição e também ligações para camadas anteriores (retroalimentação, em direção à camada de entrada). Essas características transformam as redes em sistemas dinâmicos com características de auto-organização (ENGEL, 2001). O principal exemplo de redes auto-organizáveis são os *Mapas Auto-Organizáveis* (SOM) (KOHONEN, 1990).

2.2.3 Aprendizado por Retropropagação em Redes de Múltiplas Camadas

O aprendizado por retropropagação do erro pertence ao aprendizado supervisionado, mas será tratado separadamente devido à importância desempenhada no aprendizado das redes neurais de múltiplas camadas (MLP), utilizadas neste trabalho. As redes MLP

(*Multi Layer Perceptron*) representam uma generalização do perceptron de camada única, possuindo várias camadas (camada de entrada, camadas ocultas e camada de saída). Uma rede MLP totalmente conectada com 2 camadas ocultas é mostrada na figura 2.5. As redes MLP tornaram possível a resolução de problemas complexos, não-lineares ou não separáveis por retas ou planos. Outras tarefas (além da classificação) como aproximação de funções e filtragem também são realizadas por esse modelo.

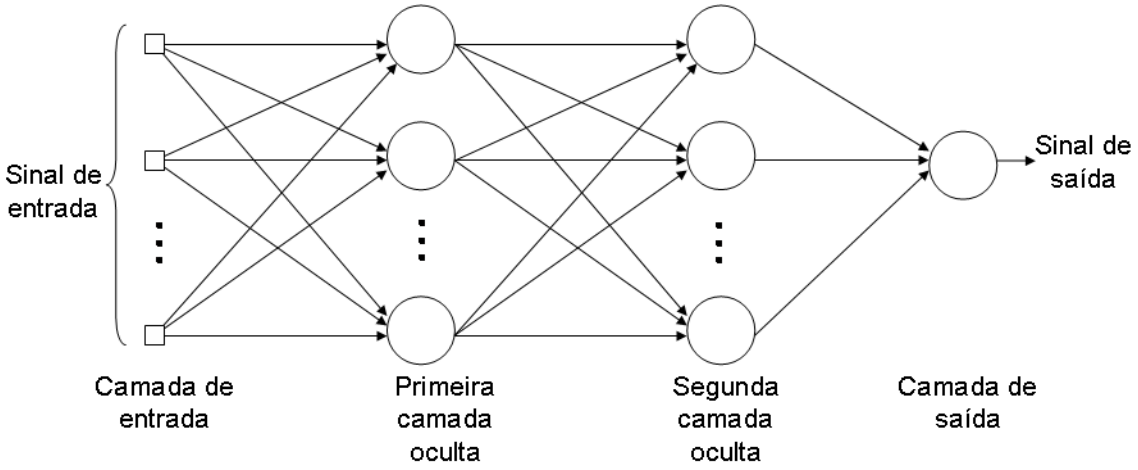

Figura 2.5: Modelo de uma rede MLP com duas camadas ocultas

A rede MLP possui a característica de alta conectividade entre os neurônios de uma camada para a próxima, como mostrado na figura 2.5. A presença dos neurônios das camadas ocultas capacita a rede a aprender tarefas complexas, extraindo progressivamente as características mais importantes dos padrões de entrada (HAYKIN, 2001a). Outra característica importante é que cada neurônio possui uma função de ativação não-linear (sigmóide). Essa não-linearidade é pequena, de primeira ordem apenas, podendo ser diferenciada sempre. Mesmo com uma não-linearidade suave na saída de cada neurônio, a existência de vários neurônios na(s) camada(s) oculta(s) propicia o tratamento de não-linearidades de graus muito maiores.

A eficiência e poder das redes MLP são obtidos devido ao seu uso combinado com o poderoso algoritmo da retropropagação do erro (*backpropagation*) (RUMELHART et al., 1986). O algoritmo da retropropagação é derivado da regra delta e funciona através de uma propagação para frente na rede e de uma propagação para trás. Na propagação para frente, a rede passa o sinal adiante de camada em camada. Esse sinal refere-se às saídas dos neurônios (depois da função de ativação). Durante a propagação, os pesos sinápticos são fixos. Quando o sinal passar pela camada de saída, é calculado o sinal de erro, subtraindo a saída da rede da saída desejada. Na fase da retropropagação o sinal de erro é passado de volta, da camada de saída até a camada de entrada, ajustando-se os pesos de acordo com uma parcela da sua contribuição no erro. A "contribuição de cada peso sináptico no erro" está relacionada com o quanto cada peso deveria ser ajustado na propagação atual do sinal de entrada atual e é calculada pela derivada parcial do erro em relação a cada peso. A "parcela" de ajuste é chamada de taxa de aprendizado (representada por η) e controla a velocidade com que o aprendizado convergirá.

A regra delta funciona ajustando-se o vetor de pesos de acordo com o *gradiente* do erro. O erro de um neurônio j, em um instante n é definido por:

$$e_j(n) = d_j(n) - y_j(n) \tag{2.7}$$

Onde $y_j(n)$ é o valor de saída do neurônio e $d_j(n)$ é o valor desejado para essa saída. O gradiente representa a derivada do erro pelos pesos no instante atual. Os pesos são atualizados no sentido oposto do gradiente (minimizando a derivada do erro pelos pesos). Essa correção é feita de acordo com um parâmetro η, que determina o "tamanho do passo" que será dado no sentido oposto ao gradiente. Então a equação de atualização dos pesos será:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \eta \nabla (E(\mathbf{w})) \tag{2.8}$$

Onde $\mathbf{w}(k)$ é o vetor de pesos e $\mathbf{w}(k+1)$ é vetor atualizado para o instante posterior. A função ∇ é o gradiente do erro em função dos pesos. $E(\mathbf{w})$ é uma função de custo, baseada no erro da rede. Para o ajuste dos pesos da rede, o algoritmo *backpropagation* deriva o MSE (*erro médio quadrado*) de saída pelo peso a ser ajustado. Visando dar uma explicação concisa e didática, será mostrada uma seqüência de passos em que o sinal de erro é derivado até chegar em um dos pesos da camada de saída:

1. Deriva-se o erro final (MSE) em função do erro de cada neurônio da camada de saída, representado por $e_j(n)$. O MSE ($\xi(n)$) é dado por:

$$\xi(n) = \frac{1}{2} \sum e_j^2(n) \tag{2.9}$$

   A derivada do MSE em função do erro de saída do neurônio da camada de saída é o próprio erro:

$$\frac{\partial \xi(n)}{\partial e_j(n)} = e_j(n) \tag{2.10}$$

2. Deriva-se o erro individual de cada neurônio em função da saída final do neurônio j (y_j). Como o erro é o valor desejado menos a saída da rede, diferenciando-se ambos os lados da Equação 2.7, a derivada será -1:

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \tag{2.11}$$

3. Deriva-se a saída do neurônio pelo valor do potencial de ativação u_j. Como y_j é a aplicação da função de ativação sobre u_j, a derivada será a derivada da função de ativação:

$$\frac{\partial y_j(n)}{\partial u_j(n)} = \varphi_j'\left(u_j(n)\right) \tag{2.12}$$

4. Finalmente, deriva-se o potencial de ativação do neurônio (u_j) por cada um de seus pesos (w_{ji}). O índice i é relativo ao neurônio de origem da conexão que tem o peso w_{ji}. Essa derivada será a saída do neurônio i:

$$\frac{\partial u_j(n)}{\partial w_{ji}(n)} = y_i(n) \tag{2.13}$$

Pela regra da cadeia das derivadas parciais, encadeando-se todas as derivadas da seqüência acima, a derivada do erro final pelo peso w_{ji} será expressa por:

$$\frac{\partial \xi(n)}{\partial w_{ji}(n)} = \frac{\partial \xi(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial u_j(n)} \frac{\partial u_j(n)}{\partial w_{ji}(n)} \tag{2.14}$$

O ajuste aplicado no peso w_{ji} é uma parcela (η) da derivada mostrada acima, definida como *regra delta*, que está no sentido contrário ao gradiente, de acordo com a Equação 2.8:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \xi(n)}{\partial w_{ji}(n)} \tag{2.15}$$

Mostrou-se o caso do ajuste de um peso de um dos neurônios da camada de saída. Para que esse ajuste seja feito em um peso de um dos neurônios das outras camadas, utiliza-se o conceito de gradiente local. O gradiente local é representado pela derivada parcial do erro quadrático em relação ao potencial de ativação de cada neurônio, com valor negativo:

$$\delta_j(n) = -\frac{\partial \xi(n)}{\partial u_j(n)} \tag{2.16}$$

A derivação em relação ao pesos pode ser expressa em função do gradiente local de cada neurônio:

$$\frac{\partial \xi(n)}{\partial w_{ji}(n)} = -\delta_j(n) \frac{\partial u_j(n)}{\partial w_{ji}(n)} \tag{2.17}$$

Os gradientes locais de uma camada (por exemplo última camada oculta) ($\delta_j(n)$) são calculados recursivamente, a partir dos gradientes locais da camada sucedente (camada de saída). Tem-se então a fórmula de retroprogação do gradiente local, a partir dos gradientes locais das camadas posteriores:

$$\delta_j(n) = \varphi'_j\left(u_j(n)\right) \sum_k \delta_k(n) w_{kj}(n) \tag{2.18}$$

Para uma explicação mais detalhada de todo o funcionamento do algoritmo da retropropagação, com demonstrações de cada valor de derivada que foi aqui apresentado, pode-se consultar (HAYKIN, 2001a).

2.3 Tarefas Realizadas por Redes Neurais

As principais tarefas em que são utilizadas redes neurais são tratadas nesta seção: reconhecimento de padrões, associação de padrões, aproximação de funções e filtragem. As RNs com saída discreta são utilizadas como classificadores universais e as redes com saída contínua podem ser usadas como regressores (aproximadores) universais (CYBENKO, 1989) (HAYKIN, 2001a) (ENGEL, 2001). Nas aplicações de RNs para predição de séries temporais são utilizadas as tarefas de aproximação de funções e filtragem.

2.3.1 Reconhecimento de Padrões

O reconhecimento de padrões é uma tarefa que aproxima muito as redes neurais dos seres humanos. Reconhecimento de padrões pode ser definido como o processo em que um conjunto de entradas ou características (padrão) é atribuído a uma classe entre um conjunto definido de classes. O reconhecimento de padrões também é chamado de classificação.

Como exemplos de reconhecimento de padrões habilmente realizados por seres humanos, pode-se citar o reconhecimento de rostos familiares em uma multidão; reconhecimento de uma pessoa mais envelhecida a partir de cenas dessa pessoa quando mais jovem ou de características de parentes; separação de grãos de feijão bons dos demais; identificação de voz no rádio ou telefone; classificação de modelos de carros ou tipos de

produtos por suas características. As RNs são muito aplicadas em reconhecimento de padrões, principalmente quando podem ser representados facilmente de forma numérica. Por exemplo, análise de crédito, identificando bons e maus pagadores; análise de qualidade de produtos a partir de índices de substâncias; classificação de riscos de doenças a partir de indicadores; identificação de graus e classificações de tumores; casamento de perfis de usuários da internet.

Para o reconhecimento de padrões, as RNs passam por um processo de treinamento, onde recebem repetidamente padrões de entrada com sua respectiva categoria. A partir de então a rede consegue classificar padrões não vistos de acordo com a variação estatística das características dos padrões e pela associação dessas características com as classes. As RNs podem funcionar de duas formas: com uma rede não-supervionada para a extração de características e com outra rede supervionada para classificação; ou com uma única rede MLP alimentada adiante, utilizando aprendizado supervisionado. Nesse último caso, as unidades da(s) camada(s) oculta(s) realizam a extração das características e as unidades da camada de saída realizam a classificação.

2.3.2 Associação de Padrões

A tarefa de associação de padrões é representada pelas memórias associativas (TAYLOR, 1956), que são memórias construídas com neurônios artificiais, inspiradas no cérebro e que aprendem por associação. A associação pode ser de dois tipos: auto-associação ou heteroassociação.

Na auto-associação, primeiramente passa-se um conjunto de padrões repetidamente para a rede armazenar. Posteriormente, apresenta-se uma representação parcial ou ruidosa de um padrão e a rede recuperará o padrão original. Por exemplo, pode-se armazenar uma imagem de um rosto e depois apresentar a região do olho desse rosto ou uma versão menos nítida da imagem para a rede recuperar a imagem original. Na heteroassociação, a associação ocorre entre um conjunto de padrões e outro conjunto diferente de padrões. Em associação de padrões o aprendizado é supervisionado. Em uma memória associativa linear, os neurônios da rede atuarão como combinadores lineares. Sendo $\mathbf{a}$ o vetor de entrada (índice) e $\mathbf{b}$ o vetor de saída (padrão recuperado), a relação de entrada e saída será dada por:

$$\mathbf{b} = \mathbf{M}\mathbf{a} \tag{2.19}$$

Onde $\mathbf{M}$ é a matriz de associação, representando a conectividade da rede. Em uma memória associativa não-linear, a relação de entrada e saída será dada por:

$$\mathbf{b} = \mathrm{f}[\mathbf{M}, \mathbf{a}] \tag{2.20}$$

Onde $\mathrm{f}[\mathbf{M}, \mathbf{a}]$ é uma função não-linear da relação de associação com a entrada. Uma memória associativa pode ser comparada com um classificador de padrões, onde as categorias de classificação são os vetores armazenados. Um padrão apresentado como entrada será classificado pela memória em uma dessas categorias, dependendo do critério de proximidade definido na memória (KOVÁCS, 2002).

2.3.3 Aproximação de Funções

Uma rede neural MLP, treinada com o algoritmo da retropropagação pode ser usada como um aproximador de funções de caráter geral. O objetivo da aproximação de funções é treinar uma RN para aproximar uma função com mapeamento de entrada-saída não-linear, representada por:

$$\mathbf{d} = \mathbf{f}(\mathbf{x}) \tag{2.21}$$

Onde $\mathbf{x}$ é o vetor de entrada, $\mathbf{d}$ é o vetor de saída e $\mathbf{f}(\cdot)$ é uma função desconhecida de valor vetorial. A rede neural não conhece a função $\mathbf{f}(\cdot)$, mas possui um conjunto de amostras rotuladas da função. O objetivo é projetar uma RN que tenha um mapeamento de entrada-saída $\mathbf{F}(\cdot)$ suficientemente próximo de $\mathbf{f}(\cdot)$. A proximidade citada é no sentido euclidiano dos vetores de saída (erro médio quadrado). A distância, ou erro, deverá estar abaixo de um limiar máximo de aceitação.

O teorema da aproximação universal (CYBENKO, 1989) considera as RNs alimentadas adiante de apenas uma camada oculta, com número suficiente de neurônios, como uma classe viável de soluções aproximativas para funções. Uma rede MLP, treinada com algoritmo de retropropagação, funciona em um esquema em que as funções de ativação atuam sucessivamente (em cascata). Para o caso de uma função de uma única saída, teria-se a representação para o aproximador universal:

$$F(\mathbf{x},\mathbf{w}) = \varphi\left(\sum_k w_{ok}\varphi\left(\sum_j w_{kj}\varphi\left(\cdots\varphi\left(\sum_i w_{li}x_i\right)\right)\right)\right) \tag{2.22}$$

Onde $\varphi(\cdot)$ é uma função de ativação sigmóide. w_{ok} é o peso do neurônio k, na última camada oculta, para o neurônio de saída o. Os pesos das outras camadas seguem a mesma notação. x_i representa cada uma das entradas. A sucessão de funções de ativações suaves possibilita que uma MLP possa aproximar as derivadas de uma mapeamento de entrada-saída desconhecido, como as funções diferenciáveis por partes (HAYKIN, 2001a).

2.3.4 Filtragem

A função dos filtros é separar sinais que pertencem a certas classes dos demais. Normalmente extrai-se um tipo de sinal (dominante) em dados onde todos os outros tipos de sinais são considerados ruído. O ruído pode ser desde erro em sensores até sinais adversos ao sinal de interesse no ambiente. Existem três tarefas básicas para um filtro (HAYKIN, 2001a) (RUSSELL; NORVIG, 2004):

Filtragem Refere-se à estimativa do sinal (informação) no tempo n, utilizando-se os dados (ruidosos) obtidos até n (inclusive);

Previsão Utilizam-se os dados medidos até o tempo n (inclusive) para estimar informação no tempo futuro $n+k$, onde $k > 0$;

Suavização Utilizam-se não apenas dados medidos até o instante n, mas também após (dados já filtrados ou estimados). A estimativa é feita em um instante atrasado, melhorando a medida ruidosa obtida anteriormente ou recuperando um dado faltante. Pode-se também retroceder em todo o conjunto de dados para melhorá-lo (suavizando todos esses dados). Do ponto de vista estatístico, a suavização é mais precisa que a filtragem, uma vez que já utiliza dados filtrados como entrada.

Na figura 2.6 é mostrada uma rede neural funcionando como um filtro previsor. As entradas $(x(n-1), x(n-2), \cdots, x(n-T))$ são medidas de instantes anteriores, em um intervalo finito de T atrasos eqüidistantes. O funcionamento da rede é como no aprendizado supervisionado pois o sinal no instante n atua como resposta desejada. O sinal de erro $e(n)$ (diferença entre a saída da rede e o desejado) é utilizado para ajustar os pesos da rede.

A previsão realizada pela RN pode ser considerada como uma construção do modelo. Quanto menor for o erro da rede, melhor será o desempenho desta rede como modelo do

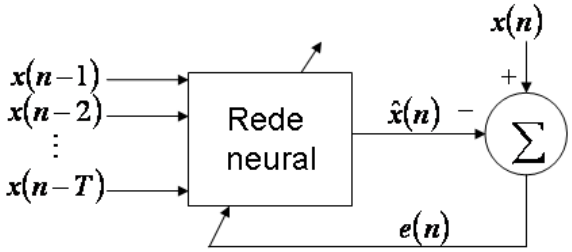

Figura 2.6: Modelo de um filtro com rede neural

processo físico gerador dos dados. Quando o processo gerador for não-linear, a rede será uma poderosa alternativa de previsão, devido às suas unidades não-lineares.

2.4 Aplicações de Redes Neurais

As tarefas desempenhadas pelas redes neurais, tratadas na seção anterior, capacitam as redes para uma infinidade de aplicações. Nesta seção, dois grandes campos de aplicações, relacionados com este trabalho, são descritos: identificação de sistemas dinâmicos e processamento temporal. Essas áreas estão interligadas, uma vez que a identificação de sistemas também utiliza processamento temporal. Descreve-se o uso das RNs nessas aplicações, apresentando-se o modelo das redes utilizadas.

2.4.1 Processamento Temporal com Redes Neurais

O processamento temporal é muito importante em grande parte das atividades do corpo humano como fala, interpretação de sinais visuais, controle motor e em uma infinidade de aplicações do cotidiano, como mostrado no capítulo das séries temporais. Sendo o tempo muito importante para essas atividades, também o será no uso de redes neurais para representá-las. O tempo pode ser contínuo ou discreto, mas para uso em sistemas computacionais sempre será considerado o tempo discreto.

Para que uma RN possa trabalhar com informações em seqüência temporal, é preciso haver uma representação explícita ou implícita no tratamento dos dados pela rede:

Representação Explícita do Tempo Essa representação é utilizada nas redes TLFN (redes alimentadas adiante atrasadas no tempo). Os dados são modificados para dar a idéia de tempo, o tempo está presente na própria estrutura dos dados. As T entradas anteriores são repetidas e entram novamente nos instantes seguintes. A cada entrada a rede receberá simultaneamente as entradas do tempo $n-T$ até n, onde n é instante atual e T é o número de atrasos temporais;

Representação Implícita do Tempo Os dados entram na rede de maneira normal, em seqüência temporal, sem repetição. A cada instante n, entram na rede apenas os dados relativos a esse instante. Neste caso, toda a estrutura de atrasos temporais é feita internamente pela rede através de sua arquitetura. Existem vários tipos de redes construídas para realizar a representação temporal internamente, como as redes recorrentes atrasadas no tempo (TDNN) (CLOUSE et al., 1997) e as Redes de Elman (ELMAN, 1990).

Existem numerosas aplicações com processamento temporal como previsão de séries temporais, filtragem de ruído, controle adaptativo e identificação de sistemas. Neste trabalho é empregada a representação explícita do tempo.

2.4.2 Redes Neurais em Identificação de Sistemas Dinâmicos

Um sistema é dito dinâmico quando seu estado varia com o tempo (HAYKIN, 2001a). A identificação de sistemas consiste em criar uma abordagem experimental para modelar um processo de dinâmica desconhecida. Os passos para a identificação são a seleção de um modelo, a configuração desse modelo e a sua validação. A identificação de sistemas dinâmicos lineares e sem ruído é um problema relativamente simples, podendo ser resolvido por um método algébrico de determinação de parâmetros. Quando as medidas são imprecisas, envolvendo ruídos ou incertezas, o problema passa a ser de **estimação** de parâmetros, resolvido por métodos estatísticos. Porém, se o sistema dinâmico for não-linear, os métodos estatísticos não dispõem de ferramentas muito precisas (KOVÁCS, 2002).

Uma planta dinâmica não-linear pode ter sua identificação baseada em um modelo de espaço de estados (MEE) ou em um modelo de entrada-saída. Um MEE é utilizado na identificação com RNs na forma da figura 2.7. A rede neural da figura 2.7a serve para estimar o estado. O estado calculado realimenta a entrada do sistema. A função $\mathrm{f}(\cdot,\cdot)$ representa a função real de cálculo do estado. A rede da figura 2.7b serve para estimar a medida. A função $\mathrm{h}(\cdot)$ representa a função real de medida. As duas redes mostradas na figura 2.7 operam em modo síncrono na identificação do sistema.

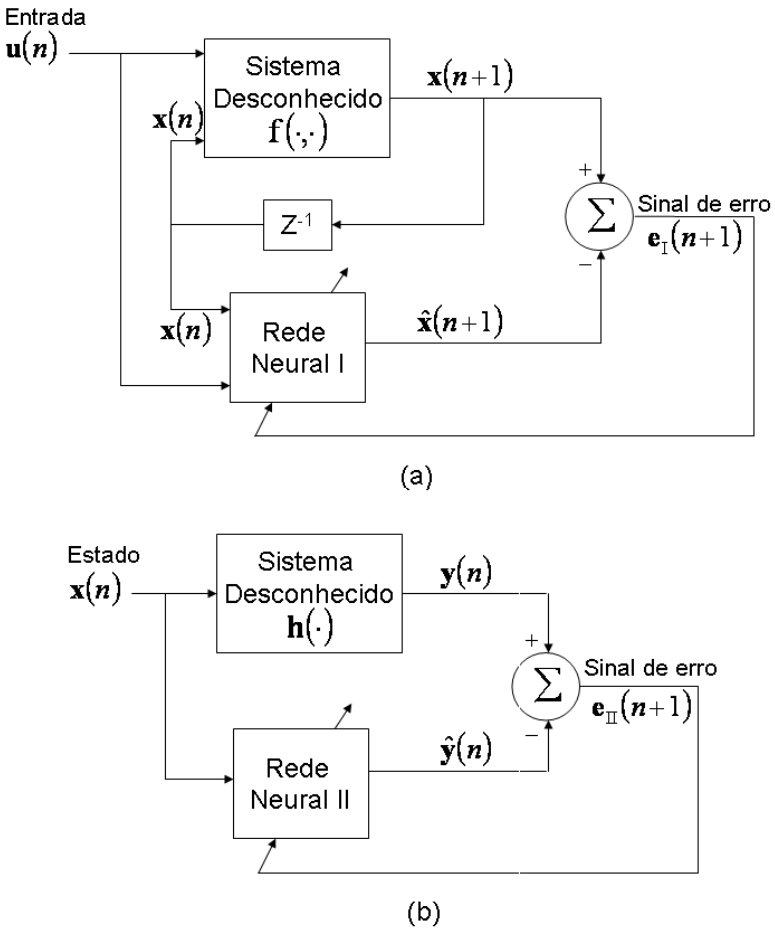

Figura 2.7: Identificação de sistemas com redes neurais, baseada no modelo de espaço de estados

Quando os sistemas são pouco conhecidos, aplica-se o modelo de entrada-saída. Esse modelo supõe que o sistema seja acessível somente por meio de suas saídas, não existindo

o conceito de estados. A arquitetura da rede neural é uma rede de múltiplas camadas alimentada adiante (MLP). Ocorre a realimentação da saída da MLP para a entrada, através de uma linha de atraso de T unidades. A rede também recebe uma linha de atraso de T entradas *exógenas*. Normalmente utiliza-se o Modelo Auto-regressivo Não-linear com Entradas Exógenas (NARX). Para a simplificação de um sistema de uma única entrada e única saída, o modelo NARX estabelece uma relação entre as saídas passadas e a saída prevista na seguinte forma:

$$y(n+1) = F(y(n), \cdots, y(n-T+1), u(n), \cdots, u(n-T+1)) \tag{2.23}$$

Onde $y(n), \cdots, y(n-T+1)$ representam os valores anteriores da saída; $y(n+1)$ é a regressão da saída do modelo e $u(n), \cdots, u(n-T+1)$ são as entradas exógenas. Um modelo NARX de segunda ordem ($T = 2$) é mostrado na figura 2.8, onde Z^{-1} representa um atraso temporal. A estimativa é subtraída da saída real para produzir o sinal de erro. A saída $y(n+1)$ desempenha o papel de resposta desejada e o erro $e(n+1)$ é utilizado para ajustar os pesos da rede:

$$e(n+1) = y(n+1) - \hat{y}(n+1) \tag{2.24}$$

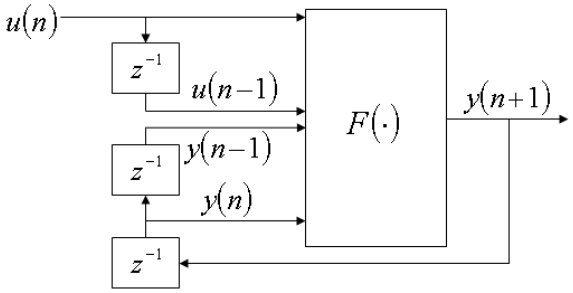

Figura 2.8: Modelo recorrente de entrada-saída NARX

Na literatura de regressão para identificação de sistemas, o modelo NARX não é auto-regressivo, pois recebe as saídas atrasadas medidas diretamente do sistema real. A figura 2.9a mostra um modelo de identificação de sistema NARX de segunda ordem. Esse modelo tem a seguinte relação de entrada-saída:

$$\hat{y}(n+1) = F(y(n), \cdots, y(n-T+1), u(n), \cdots, u(n-T+1)) \tag{2.25}$$

Onde $\hat{y}(n+1)$ representa a estimativa de saída do modelo, calculada a partir das saídas atrasadas do sistema e das entradas exógenas. O modelo realmente auto-regressivo, correspondente à arquitetura NARX de redes recorrentes é chamado na literatura de regressão de NOE (*Nonlinear Output Model*). A figura 2.9b mostra um modelo NOE de segunda ordem. O modelo NOE utiliza como entrada as saídas de predições passadas em vez de utilizar as medidas do sistema real e possui a relação de entrada-saída:

$$\hat{y}(n+1) = F(\hat{y}(n), \cdots, \hat{y}(n-T+1), u(n), \cdots, u(n-T+1)) \tag{2.26}$$

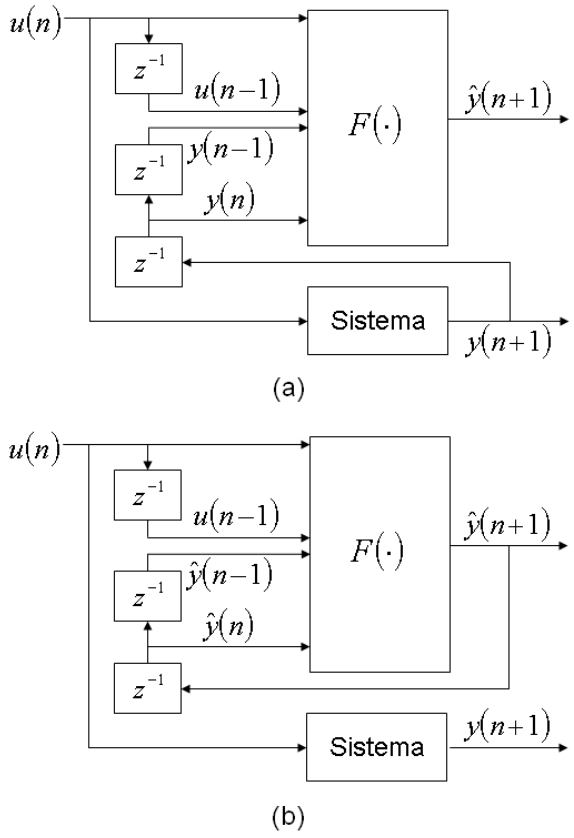

Figura 2.9: Modelo de entrada-saída na regressão para identificação de sistemas. (a) Modelo NARX. (b) Modelo NOE.

3 FILTRO DE KALMAN

Considerando os exemplos abaixo, podem ser percebidas algumas similaridades (adaptados de (RUSSELL; NORVIG, 2004)):

- Uma pessoa observando o aparecimento de um vaga-lume "pisca-pisca" e tentando adivinhar onde será a posição em que ele acenderá a luz novamente;
- Um operador de radar da segunda guerra mundial tentando descobrir a posição do inimigo a partir de um sinal fraco e impreciso que surge a cada 5 segundos na tela;
- Um astrônomo tentando descobrir a trajetória dos planetas a partir de um conjunto de observações inexatas de ângulos em intervalos irregulares de tempo, medidos de forma imprecisa.

A semelhança de todos esses casos é que se tenta avaliar o estado (posição, velocidade, etc.) de um sistema físico através de observações ruidosas ao longo do tempo. Todos os problemas citados podem ser formulados como inferência em um modelo de probabilidade temporal. O método do Filtro de Kalman foi criado para resolver esse tipo de problema. A física do movimento será o modelo de transição de estado e o sistema de observação (visão, sensores, etc.) será o modelo de medida. Neste capítulo são mostradas as definições iniciais para entender o Filtro de Kalman, o seu algoritmo e funcionamento e as diferenças entre o Filtro de Kalman Estendido e o Filtro de Kalman Discreto.

3.1 Definições Iniciais

O Filtro de Kalman (FK) utiliza muitos conceitos que embasam o seu funcionamento e suas aplicações. Baseado nos trabalhos (KALMAN, 1960) (WELCH; BISHOP, 2001) (HAYKIN, 2001b) (MACHADO, 2003) (HAYKIN, 2001a) (ENGEL, 2005), pode-se definir que o FK é um filtro linear ótimo, que modela processos estocásticos baseando-se na transição de estados e executando predições em dados ruidosos. Para que as definições anteriores sejam mais facilmente entendidas, necessitam-se dos conceitos de estimativa ótima, processo estocástico, sistema de transição de estados e ruído. Nesta seção serão abordados todos esses conceitos para uma melhor compreensão do FK.

3.1.1 Ruído

O ruído representa variações ou incertezas nos dados. Ruído pode ser definido como o conjunto das influências não-sistemáticas sobre o comportamento de um sistema, não estando compreendido no modelo determinístico (previsível) desse sistema (MACHADO, 2003). A presença de ruído pode causar grandes dificuldades para métodos de predição,

que interpretam o ruído como parte integrante dos dados. Muitos métodos preditivos, como é o caso das redes neurais, apresentam dificuldades em diferenciar o sinal (informação pura) do ruído. Os experimentos apresentados nesta dissertação comprovam essa dificuldade.

Mesmo que o ruído não seja previsível, é possível modelá-lo. Modelagem de ruído atrai estudos da teoria da probabilidade (GLYMOUR et al., 1996) e normalmente utiliza estatísticas da ocorrência do ruído, como a variância (WELCH; BISHOP, 2001). O modelo mais comum de ruído é o ruído branco com distribuição gaussiana. Essa distribuição de ruído é utilizada como padrão para o Filtro de Kalman e é definida pela seguinte função:

$$f(x, x_m, q) = \frac{e^{-\frac{1}{2}\frac{(x-x_m)^2}{q^2}}}{\sqrt{2\pi q^2}} \tag{3.1}$$

Onde x_m é a média dos valores possíveis de x e q^2 é a variância desses valores. A distribuição gaussiana é obtida a partir de diversas fontes de influência independentes, com média finita (MACHADO, 2003).

3.1.2 Processo Estocástico

A idéia de modelo matemático é muito utilizada para descrever sistemas físicos, principalmente nas ciências exatas e engenharias. Um modelo matemático é classificado em dois tipos: determinístico e estocástico. Chama-se um modelo de determinístico quando não existe aleatoriedade sobre o comportamento do sistema em um dado instante de tempo e chama-se de modelo estocástico quando existe aleatoriedade. No modelo determinístico, supõe-se que, de posse das mesmas entradas e circunstâncias, a saída (comportamento) do sistema será previsível. Porém, em sistemas reais, normalmente existem muitas incertezas ou um conjunto de variáveis desconhecidas que atribuem características aleatórias ao sistema, dificultando a sua passagem para um modelo matemático determinístico. Esses sistemas podem ser descritos em termos probabilísticos, em função da probabilidade de a saída estar entre dois limites definidos, e são chamados de processos estocásticos.

Um processo estocástico possui duas propriedades fundamentais (HAYKIN, 2001c):

- O processo é em função do tempo;
- É aleatório, no sentido de que antes de uma transição temporal não é possível prever com exatidão os valores futuros.

Um processo estocástico será então uma coleção de Variáveis Aleatórias Discretas (VAD), organizadas em função do tempo, formando um espaço de estados, ou espaço amostral. A descrição do processo não poderá ser realizada de forma determinística, mas poderá ser dada pelos momentos do processo estocástico: primeiro momento (média); segundo momento (variância e funções de covariância) (MANTOVANI, 2004). Se o tempo for discreto (exemplo: $T = 0, 1, 2, 3, \cdots$), tem-se um processo estocástico discreto, porém, se o tempo for contínuo (exemplo: $T = 0 < t < +\infty$), tem-se um processo estocástico contínuo.

3.1.3 Modelo de Espaço de Estados

Modelo de Espaço de Estados (MEE) é uma ampla classe de modelos, também chamados de Modelos Lineares Dinâmicos (MLD), introduzidos por Rudolph Kalman (KAL-

MAN, 1960). Esses modelos têm sido muito usados para modelar dados da economia, da área médica, de meteorologia, de ciências do solo, dentre outros. O estado de um sistema dinâmico é formalmente definido como o "conjunto de quantidades que assumem toda a informação sobre o comportamento passado e que é necessária para descrever o seu comportamento futuro, exceto pelos efeitos puramente externos que surgem devido à entrada (excitação) aplicada".

Segundo (MORETTIN; TOLOI, 2004), todo modelo de séries temporais de q dimensões possui representação em espaço de estados, relacionando um vetor de observações $\mathbf{Z}_t$ e um vetor de ruído $\mathbf{v}_t$, através de um processo $\mathbf{X}_t$, com p dimensões, chamado de vetor de estados. Essa representatividade das séries temporais em espaços de estados motiva o uso da estrutura de estados do Filtro de Kalman para a PST, neste trabalho.

Um MLD possui duas equações. A Equação 3.2 é chamada de *equação de processo*, pois calcula o vetor de estados do processo $\mathbf{X}_t$ e a Eq. 3.3 é chamada de *equação de medida*, pois calcula a medida das variáveis observáveis do processo:

$$\mathbf{X}_t = \mathbf{G}_t\mathbf{X}_{t-1} + \mathbf{w}_t, \quad t = 1, \cdots, N \tag{3.2}$$

$$\mathbf{Z}_t = \mathbf{A}_t\mathbf{X}_t + \mathbf{v}_t \tag{3.3}$$

Onde:

- $\mathbf{G}_t$ é a matriz de transição de estado, de ordem $(p \times p)$;
- $\mathbf{w}_t$ é um vetor de ruído, representando o ruído (perturbação) do sistema, de ordem $(p \times 1)$, com média zero e matriz de covariância $\mathbf{Q}$;
- $\mathbf{A}_t$ é a matriz de observação do sistema, de ordem $(q \times p)$;
- $\mathbf{v}_t$ é o vetor ruído da observação, de ordem $(q \times 1)$, com média zero e matriz de covariância $\mathbf{R}$;
- Os vetores de ruído $\mathbf{v}_t$ e $\mathbf{w}_t$ são não-correlacionados entre si e não-correlacionados com o estado inicial.

Quando os vetores de ruído forem normalmente distribuídos, diz-se que o espaço de estados é gaussiano. As matrizes $\mathbf{A}$ e $\mathbf{G}$ são determinísticas, então se houver variação no tempo, esta variação será definida a priori. Quando as matrizes de transição não variam no tempo o sistema é chamado *invariante no tempo* ou *homogêneo no tempo*. Um caso especial de *sistemas invariantes no tempo* são os modelos estacionários. Nesse caso, além de possuírem o mesmo sistema de transição, esses modelos seguem uma mesma variância em torno de uma média (MORETTIN; TOLOI, 2004).

3.1.4 Estimativa Ótima

O conceito de estimação ótima é muito importante para a compreensão do princípio de funcionamento do FK, que é um filtro linear ótimo. Esta seção é apresentada no contexto das variáveis aleatórias discretas (VAD), generalizando-se para um vetor de VAD (vetor de estados) propagado adiante no tempo. Segundo (HAYKIN, 2001b), tem-se a variável observável y_k:

$$y_k = x_k + v_k \tag{3.4}$$

Onde x_k é um sinal desconhecido e v_k uma componente de ruído aditivo. Realiza-se uma estimativa $\hat{x}_k$ do sinal x_k, que normalmente difere desse sinal. Para derivar a

estimativa de uma maneira ótima, necessita-se de uma função de custo (função de perda) para estimativas incorretas. Essa função deve satisfazer duas condições:

- Ser não-negativa
- Ser uma função não-decrescente do *erro de estimativa* δx, definido por:

$$\delta x_k = x_k - \hat{x}_k \tag{3.5}$$

As duas condições são satisfeitas por um erro mínimo quadrado (MSE), definido por:

$$J_k = E\left[(x_k - \hat{x}_k)^2\right] = E\left[(\delta x_k)^2\right] \tag{3.6}$$

Onde E representa a função expectativa. A dependência da função de custo J_k no tempo k salienta a natureza não-estacionária do processo recursivo de estimação.

3.1.5 Introdução ao Filtro de Kalman

O Filtro de Kalman (FK) foi proposto por Rudolf Emil Kalman, em 1960, em seu famoso artigo descrevendo uma solução recursiva para o problema da filtragem linear de dados discretos (KALMAN, 1960). Essa primeira versão era usada apenas para problemas lineares. Na época, a resolução de problemas não-lineares era inviável devido ao baixo poder de processamento dos computadores.

O Filtro de Kalman resolve eficientemente o problema da variância mínima do erro, utilizando a abordagem da filtragem ótima e prevê estados passados, presentes e até estados futuros. Esses estados pertencem a sistemas dinâmicos lineares, ou seja, processos governados por uma equação linear estocástica . No Filtro de Kalman Estendido, os estados pertencem a sistemas dinâmicos não-lineares, governados por equação estocástica não-linear (WELCH; BISHOP, 2001).

O FK pode ser usado para estimativa analítica de problemas, onde estima-se o estado de um sistema com processo linear e modelos de medidas com incertezas gaussianas. A *função de densidade de probabilidade* (fdp) sobre o vetor de estados é uma distribuição gaussiana inteiramente determinada por seu vetor de média e matriz de covariância. A fdp define como será a transição de estados do processo. Essas médias e covariâncias são atualizadas com o algoritmo do FK.

O Filtro de Kalman (assim como todas suas variantes) é aplicado em sistemas que possuem variáveis de estados contínuos, cujas medições (e normalmente também o processo) apresentam ruído. Como exemplos desses sistemas pode-se citar: trajetórias de aeronaves e mísseis; acompanhamento de pessoas e automóveis; reconstrução da trajetória de partículas; determinação de correntes oceânicas e acompanhamento acústico de submarinos. Outras exemplos de aplicações podem se dar em indústrias químicas, reatores nucleares, ecossistemas vegetais e variáveis da economia (RUSSELL; NORVIG, 2004). Em (RUTGEERTS et al., 2005) é mostrado um sistema de treinamento para robôs por demonstração humana. (NYGREN; JANSSON, 2004) mostra a utilização do FK para navegação de submarinos. Outra utilização do FK, para rastreamento de pessoas em tempo real, é mostrada por (GIRONDEL; CAPLIER; BONNAUD, 2004).

3.2 O Algoritmo do Filtro de Kalman

Esta seção apresenta o funcionamento do algoritmo do Filtro de Kalman em cada uma de suas fases, com suas fórmulas e explicações. As inspirações e bases para esta seção podem ser encontradas em (KALMAN, 1960), (WELCH; BISHOP, 2001), (ENGEL,

2005), (HAYKIN, 2001b) e (MACHADO, 2003). As fórmulas aqui descritas equivalem ao Filtro de Kalman Estendido (FKE), uma vez que o FKE é uma generalização do Filtro de Kalman que pode ser utilizado também em sistemas não-lineares. O FK sem a linearização (com matrizes) pode ser considerado abrangido pelo FKE e será tratado na seção seguinte.

O Filtro de Kalman assume o problema de estimar um estado $\mathbf{x}$, de um processo controlado de tempo discreto, regido por uma equação estocástica:

$$\mathbf{x}(n) = \mathbf{f}[\mathbf{x}(n-1), \mathbf{u}(n-1), \mathbf{w}(n-1)] \tag{3.7}$$

Onde $\mathbf{x}(n)$ é o vetor de estados do sistema no instante n, $\mathbf{u}(n)$ é o vetor de entrada (ação tomada no instante n) e $\mathbf{w}(n)$ é o ruído que a dinâmica do processo possui. A função $\mathbf{f}$ representa a dinâmica determinística do processo, isto é, a parte conhecida do processo de transição de estado. O ruído $\mathbf{w}(n)$ é assumido como gaussiano de média zero (ruído branco) e torna o processo estocástico, necessitando ser estimado. Isso significa que a dinâmica do processo não é determinística, havendo incerteza sobre o estado real do sistema, após uma ação ser tomada. A covariância do ruído de processo é representada pela matriz $\mathbf{Q}$, dada pelo produto externo do vetor de ruído de processo:

$$\mathbf{Q}(n) = \left\langle \mathbf{w}(n)\,\mathbf{w}(n)^T \right\rangle \tag{3.8}$$

O estado do sistema não pode ser medido diretamente, com isso é necessário fazer estimativas sobre o estado real do sistema. O FK utiliza dois modelos lineares, um para o processo e outro para a medida. O FK funciona em duas fases. A primeira fase estima o próximo estado (projeta o estado adiante), com base na função de transição sobre o estado anterior (antes da medida), e a segunda fase atualiza a estimativa de estado, com base na medida no instante atual. A medida do sistema ($\mathbf{z}(n)$), representa um vetor de variáveis observáveis e é uma função do estado real $\mathbf{x}(n)$. O vetor $\mathbf{z}(n)$ depende também de um ruído $\mathbf{v}(n)$, chamado de ruído de medida, originado da imprecisão do mecanismo de medida do estado:

$$\mathbf{z}(n) = \mathbf{h}[\mathbf{x}(n), \mathbf{v}(n)] \tag{3.9}$$

A função de medida é determinística. Na prática (na execução do filtro), $\mathbf{h}$ representa a forma como uma estimativa de medida é inferida a partir de uma estimativa de estado. O ruído de medida $\mathbf{v}(n)$ é também considerado gaussiano "branco". A covariância do ruído de medida será dada pela matriz $\mathbf{R}$:

$$\mathbf{R}(n) = \left\langle \mathbf{v}(n)\,\mathbf{v}(n)^T \right\rangle \tag{3.10}$$

O FK armazena a matriz de covariância do erro de predição do estado ($\mathbf{P}(n)$) para utilizar na atualização da estimativa do estado. O funcionamento do filtro, com suas duas fases é mostrado na figura 3.2.

O Ganho de Kalman, representado por $\mathbf{K}(n)$, é ajustado de modo a minimizar a covariância do erro de estimação. A forma de representação $\mathbf{x}(n|n-1)$ é lida como "valor do vetor $\mathbf{x}$ para o instante n, calculada no instante $n-1$". Essa forma de representação é necessária pelo fato de muitas variáveis serem previstas *a priori* (antes da medida) e revisadas (atualizadas) *a posteriori* (depois da medida).

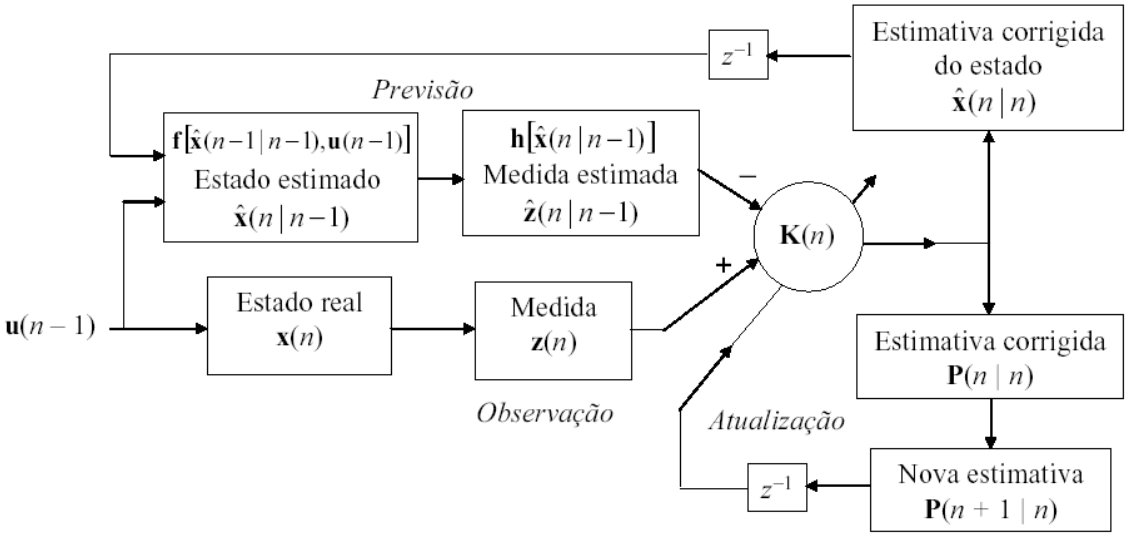

Figura 3.1: Modelo de funcionamento do Filtro de Kalman

3.2.1 Não-linearidades e Jacobianas

Um sistema é não-linear se o modelo de transição não pode ser descrito como uma multiplicação de matrizes do vetor de estados (RUSSELL; NORVIG, 2004). O Filtro de Kalman Estendido (FKE) serve para prever não-linearidades no modelo e utiliza as derivadas parciais do processo e das funções de medida (WELCH; BISHOP, 2001). O tipo de não-linearidade que o FKE trata é apenas de primeira ordem (HAYKIN, 2001b). Isso se deve ao fato que o FKE mantém toda a estrutura linear do FKD, com um processo de linearização com equações diferenciais (derivadas de primeira ordem).

O processo de linearização no FKE se dá pela utilização das matrizes jacobianas $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$, $\frac{\partial \mathbf{f}}{\partial \mathbf{w}}$, $\frac{\partial \mathbf{h}}{\partial \mathbf{x}}$ e $\frac{\partial \mathbf{h}}{\partial \mathbf{v}}$. Onde:

- $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ é a matriz jacobiana das derivadas parciais da função de transição de estado **f** em relação ao vetor de estados **x**.
- $\frac{\partial \mathbf{f}}{\partial \mathbf{w}}$ é a matriz jacobiana das derivadas parciais da função de transição de estado **f** em relação ao vetor de ruído de processo **w**.
- $\frac{\partial \mathbf{h}}{\partial \mathbf{x}}$ é a matriz jacobiana das derivadas parciais da função de medida **h** em relação ao vetor de estados **x**.
- $\frac{\partial \mathbf{h}}{\partial \mathbf{v}}$ é a matriz jacobiana das derivadas parciais da função de medida **h** em relação ao vetor de ruído de medida **v**.

Cada jacobiana representa uma matriz de derivadas parciais de cada uma das saídas da função por cada uma das posições do seu vetor de entrada. Por exemplo, a matriz $\frac{\partial \mathrm{h}}{\partial \mathbf{x}}$ será constituída das derivadas parciais de cada uma das posições do vetor de saída gerado pela função **h** em relação a cada uma das posições do vetor de estado **x**. As saídas da função representam as linhas da matriz e as posições do vetor de estados representam as

colunas. Como as saídas da função são as posições do vetor de medida $\mathbf{z}$, tem-se:

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_1}{\partial x_2} & \cdots & \frac{\partial z_1}{\partial x_T} \\ \frac{\partial z_2}{\partial x_1} & \frac{\partial z_2}{\partial x_2} & \cdots & \frac{\partial z_2}{\partial x_T} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial z_N}{\partial x_1} & \frac{\partial z_N}{\partial x_2} & \cdots & \frac{\partial z_N}{\partial x_T} \end{bmatrix} \tag{3.11}$$

Onde T é o tamanho do vetor de estados $\mathbf{x}$ e N, o tamanho do vetor de medidas $\mathbf{z}$.

Uma importante característica do FKE é que a jacobiana $\frac{\partial \mathbf{h}}{\partial \mathbf{x}}$ (na equação de Ganho de Kalman) serve para propagar corretamente ou apenas aumentar os componentes relevantes da informação de medida. Se não há um mapeamento (correlação) entre duas variáveis, o Ganho de Kalman não altera o estado dessas variáveis. A seguir serão mostradas as duas fases do FKE: Previsão (*a priori*) e Atualização (*a posteriori*).

3.2.2 Fase de Previsão

A estimativa do vetor de estados atual (Equação 3.7) *a priori* trata o modelo do processo como sendo determinístico (sem ruído):

$$\hat{\mathbf{x}}(n|n-1) = \mathbf{f}[\hat{\mathbf{x}}(n-1|n-1), \mathbf{u}(n-1), 0] \tag{3.12}$$

Onde $\hat{\mathbf{x}}(n-1|n-1)$ é a medida *a posteriori* anterior, ou seja, calculada na fase de atualização do instante anterior.

Teoricamente, o erro da estimativa $\hat{\mathbf{x}}(n|n-1)$ é dado por:

$$\delta\mathbf{x}(n|n-1) = \mathbf{x}(n) - \hat{\mathbf{x}}(n|n-1) \tag{3.13}$$

Foi dito "teoricamente", porque na prática, o valor exato de $\mathbf{x}(n)$ não é conhecido. Supondo que o erro de estimativa de processo e o ruído sejam suficientemente pequenos, pode-se expandir a equação de processo em uma Série de Taylor de primeira ordem:

$$\mathbf{x}(n) = \mathbf{f}[\hat{\mathbf{x}}(n-1|n-1), \mathbf{u}(n-1), 0] + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\delta\mathbf{x}(n-1|n-1) + \frac{\partial \mathbf{f}}{\partial \mathbf{w}}\mathbf{w}(n-1) \tag{3.14}$$

Substituindo as Equações 3.14 e 3.12 na Equação 3.13 , pode-se calcular o erro teórico da estimativa como:

$$\delta\mathbf{x}(n|n-1) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\delta\mathbf{x}(n-1|n-1) + \frac{\partial \mathbf{f}}{\partial \mathbf{w}}\mathbf{w}(n-1) \tag{3.15}$$

A covariância *a priori* do erro do erro de estimação é dada por:

$$\mathbf{P}(n|n-1) = \left\langle \delta\mathbf{x}(n|n-1)\, \delta\mathbf{x}(n|n-1)^T \right\rangle \tag{3.16}$$

Substituindo a Equação 3.15 na Eq. 3.16, tem-se:

$$\mathbf{P}(n|n-1) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{P}(n-1|n-1)\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T + \frac{\partial \mathbf{f}}{\partial \mathbf{w}}\mathbf{Q}(n-1)\left(\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right)^T \tag{3.17}$$

A estimativa de medida, que é calculada a partir da estimativa de estado é representada por:

$$\hat{\mathbf{z}}(n|n-1) = \mathbf{h}[\hat{\mathbf{x}}(n|n-1), 0] \tag{3.18}$$

Pode-se perceber que a função **h** recebe o valor 0 como ruído, isto é, trabalha apenas com a parte determinística. O erro da estimativa de medida será dado por:

$$\delta\mathbf{z}(n|n-1) = \mathbf{z}(n) - \hat{\mathbf{z}}(n|n-1) \tag{3.19}$$

Supondo erros de estimativa e ruído suficientemente pequenos, a equação de medida também pode ser expandida por uma Série de Taylor de primeira ordem:

$$\mathbf{z}(n) = \mathbf{h}[\hat{\mathbf{x}}(n|n-1), 0] + \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\delta\mathbf{x}(n|n-1) + \frac{\partial \mathbf{h}}{\partial \mathbf{v}}\mathbf{v}(n) \tag{3.20}$$

Substituindo as Equações 3.18 e 3.20 na Equação 3.19, pode-se calcular o erro teórico da estimativa de medida:

$$\delta\mathbf{z}(n|n-1) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\delta\mathbf{x}(n|n-1) + \frac{\partial \mathbf{h}}{\partial \mathbf{v}}\mathbf{v}(n) \tag{3.21}$$

3.2.3 Fase de Atualização

Para calcular a atualização do estado, antes é necessário computar o ganho de Kalman (**K**). **K** é uma matriz que representa a parte da inovação (diferença entre o que foi estimado e o que foi medido) que será incorporada ao estado em cada iteração. Essa matriz é escolhida de forma a minimizar a variância do erro final de estimação de cada uma das componentes do vetor de estados do sistema. O ganho de Kalman é gerado a partir das covariâncias do erro de predição, com as matrizes $\mathbf{S}_{zz}$ e $\mathbf{S}_{xz}$. $\mathbf{S}_{xz}$ representa a covariância entre o erro da estimativa de estado e o erro da estimativa de medida:

$$\mathbf{S}_{xz}(n|n-1) = \left\langle \delta\mathbf{x}(n|n-1)\,\delta\mathbf{z}(n|n-1)^T \right\rangle \tag{3.22}$$

Substituindo as fórmulas dos erros (Eq. 3.15 e Eq. 3.21), a Eq. 3.22 resulta em:

$$\mathbf{S}_{xz}(n|n-1) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{P}(n-1|n-1)\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T\left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T + \frac{\partial \mathbf{f}}{\partial \mathbf{w}}\mathbf{Q}(n-1)\left(\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right)^T\left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T \tag{3.23}$$

Que também é equivalente a:

$$\mathbf{S}_{xz}(n|n-1) = \mathbf{P}(n|n-1)\left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T \tag{3.24}$$

A matriz $\mathbf{S}_{zz}$ é obtida do erro de estimativa de medida:

$$\mathbf{S}_{zz}(n|n-1) = \left\langle \delta\mathbf{z}(n|n-1)\,\delta\mathbf{z}(n|n-1)^T \right\rangle \tag{3.25}$$

Substituindo os erros de estimativa de medida (Eq. 3.19), a Eq. 3.25 resulta em:

$$\begin{aligned}\mathbf{S}_{zz}(n|n-1) = {} & \tfrac{\partial \mathbf{h}}{\partial \mathbf{x}}\tfrac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{P}(n-1|n-1)\left(\tfrac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T\left(\tfrac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T \\ & + \tfrac{\partial \mathbf{h}}{\partial \mathbf{x}}\tfrac{\partial \mathbf{f}}{\partial \mathbf{w}}\mathbf{Q}(n-1)\left(\tfrac{\partial \mathbf{f}}{\partial \mathbf{w}}\right)^T\left(\tfrac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T + \tfrac{\partial \mathbf{h}}{\partial \mathbf{v}}\mathbf{R}(n)\left(\tfrac{\partial \mathbf{h}}{\partial \mathbf{v}}\right)^T\end{aligned} \tag{3.26}$$

A fórmula acima também é equivalente a:

$$\mathbf{S}_{zz}(n|n-1) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\mathbf{P}(n|n-1)\left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T + \frac{\partial \mathbf{h}}{\partial \mathbf{v}}\mathbf{R}(n)\left(\frac{\partial \mathbf{h}}{\partial \mathbf{v}}\right)^T \tag{3.27}$$

Combinando as matrizes $\mathbf{S}_{xz}$ e $\mathbf{S}_{zz}$, tem-se o Ganho de Kalman:

$$\mathbf{K}(n) = \mathbf{S}_{xz}(n|n-1)\,\mathbf{S}_{zz}^{-1}(n|n-1) \tag{3.28}$$

Com o Ganho de Kalman, calcula-se a estimativa de estado atualizada (*a posteriori*), que é uma combinação linear da estimativa anterior e da nova medida (HAYKIN, 2001b):

$$\hat{\mathbf{x}}(n|n) = \hat{\mathbf{x}}(n|n-1) + \mathbf{K}(n)\left(\mathbf{z}(n) - \mathbf{h}\left[\hat{\mathbf{x}}(n|n-1), 0\right]\right) \tag{3.29}$$

A diferença entre a medida real e a estimativa de medida *a priori* é chamada de *inovação* e representa a nova informação contida na medida. Essa nova informação não é incorporada toda de uma vez na estimativa de estado, apenas uma parte (combinação linear) dela é utilizada, de maneira ótima pelo ganho de Kalman.

Por fim, a covariância do erro da estimativa *a posteriori* é dada por:

$$\mathbf{P}(n|n) = \left\langle \delta\mathbf{x}(n|n)\,\delta\mathbf{x}(n|n)^T \right\rangle \tag{3.30}$$

ou

$$\mathbf{P}(n|n) = \mathbf{P}(n|n-1) - \mathbf{K}(n)\,\mathbf{S}_{zz}(n|n-1)\,\mathbf{K}(n)^T \tag{3.31}$$

3.3 Filtro de Kalman com Matrizes

O Filtro de Kalman foi inicialmente projetado apenas para problemas lineares, servindo para estimar estados de processo controlado, discreto no tempo. A sua equação de controle (processo) é uma equação linear estocástica.

3.3.1 Fórmulas Utilizando Matrizes

A transição de estados é feita de forma linear por meio de multiplicação por matrizes. A predição do vetor de estados $\mathbf{x}$ é feita da forma:

$$\mathbf{x}(n) = \mathbf{A}\mathbf{x}(n-1) + \mathbf{B}\mathbf{u}(n-1) + \mathbf{w}(n-1) \tag{3.32}$$

Onde $\mathbf{A}$ e $\mathbf{B}$ são matrizes que caracterizam a dinâmica determinística e linear do processo. $\mathbf{A}$ e $\mathbf{B}$ representam a função de transição $\mathbf{f}$.

A função de medida $\mathbf{h}$ também é substituída pela multiplicação por uma matriz. A matriz $\mathbf{H}$ representa a função linear de medida:

$$\mathbf{z}(n) = \mathbf{H}\mathbf{x}(n) + \mathbf{v}(n) \tag{3.33}$$

Como aqui os modelos de processo e de medida são lineares, não é necessário aplicar a linearização pelo cálculo das derivadas parciais a cada instante de tempo. Logo as matrizes de derivadas de $\mathbf{f}$ e de $\mathbf{h}$, em função do estado, serão as próprias matrizes de transição dessas funções, $\mathbf{A}$ e $\mathbf{H}$ respectivamente. As matrizes derivadas em função dos ruídos serão matrizes identidade, pois as funções $\mathbf{f}$ e $\mathbf{h}$ são diretamente relacionadas ao ruído (que considera-se somado diretamente), então tem-se:

$$\frac{\partial\mathbf{f}}{\partial\mathbf{x}} = \mathbf{A}, \quad \frac{\partial\mathbf{h}}{\partial\mathbf{x}} = \mathbf{H}, \quad \frac{\partial\mathbf{f}}{\partial\mathbf{w}} = \mathbf{I}, \quad \frac{\partial\mathbf{h}}{\partial\mathbf{v}} = \mathbf{I} \tag{3.34}$$

Onde $\mathbf{I}$ é a matriz identidade. A função de estimativa do vetor de estados do FKD é então escrita por:

$$\hat{\mathbf{x}}(n|n-1) = \mathbf{A}\hat{\mathbf{x}}(n-1|n-1) + \mathbf{B}\mathbf{u}(n-1) \tag{3.35}$$

E a estimativa de medida:

$$\hat{\mathbf{z}}(n|n-1) = \mathbf{H}\hat{\mathbf{x}}(n|n-1) \tag{3.36}$$

A matriz de covariância do erro é escrita por:

$$\mathbf{P}(n|n-1) = \mathbf{A}\mathbf{P}(n-1|n-1)\mathbf{A}^T + \mathbf{Q}(n-1) \tag{3.37}$$

As covariâncias dos erros, que formam o filtro de Kalman também são escritas em função das matrizes:

$$\mathbf{S}_{xz}(n|n-1) = \mathbf{P}(n|n-1)\mathbf{H}^T \tag{3.38}$$

$$\mathbf{S}_{zz}(n|n-1) = \mathbf{H}\mathbf{P}(n|n-1)\mathbf{H}^T + \mathbf{R}(n) \tag{3.39}$$

Substituindo as matrizes $\mathbf{S}_{xz}$ e $\mathbf{S}_{zz}$ no cálculo do Ganho de Kalman, tem-se:

$$\mathbf{K}(n) = \left(\mathbf{P}(n|n-1)\mathbf{H}^T\right)\left(\mathbf{H}\mathbf{P}(n|n-1)\mathbf{H}^T + \mathbf{R}(n)\right)^{-1} \tag{3.40}$$

A atualização da estimativa de estado permanece da mesma forma, mas pode ser escrita também com a substituição da matriz de medida na inovação:

$$\hat{\mathbf{x}}(n|n) = \hat{\mathbf{x}}(n|n-1) + \mathbf{K}(n)\left(\mathbf{z}(n) - \mathbf{H}\hat{\mathbf{x}}(n|n-1)\right) \tag{3.41}$$

E, por fim, a covariância *a posteriori* também pode ser escrita em função das matrizes:

$$\mathbf{P}(n|n) = (\mathbf{I} - \mathbf{K}(n)\mathbf{H})\mathbf{P}(n|n-1) \tag{3.42}$$

3.3.2 Limitações do FK Linear

As limitações de tratamento de não-linearidade e de modelo de ruído pelo Filtro de Kalman justificam o uso de redes neurais. Essa vantagem é mostrada na prática por (DE-CRUYENAERE; HAFEZ, 1992). O artigo mostra uma comparação do Filtro de Kalman Discreto (FKD) com um modelo de rede neural recorrente. A rede mostrada possui 2 camadas ocultas, recorrência da camada de saída para a camada de entrada e é treinada com algoritmo gradiente conjugado.

A maioria dos experimentos apresentados inclui não-linearidades e distribuições não-gaussianas, o que viola as hipóteses do FKD. Apenas no caso em que as hipóteses não são violadas, o FKD consegue ter um desempenho levemente superior a essa RN. Em todos os outros casos, a RN possui desempenho bastante superior. Como simulação, testaram-se 24 sistemas de equações, em que apenas o sistema de número I atende todas as hipóteses do FKD. Os outros 23 possuem combinações de não-linearidades com distribuições não-gaussianas. Desses 23, são mostrados 3, os sistemas de número II, III e IV. Abaixo são mostrados os 4 sistemas:

- Sistema I: Satisfaz todas as hipóteses de Kalman. Possui a seguinte equação:

$$x(n) = 0.9x(n-1) + w(n) \tag{3.43}$$

- Sistema II: Inclui uma não-linearidade sigmóide. Possui a seguinte equação:

$$\begin{aligned} x(n) &= \mathrm{G}(x(n-1)) + \mathbf{w}(n) \\ \mathrm{G}(x) &= \tfrac{1}{1+e^{-8x}} - \tfrac{1}{2} \end{aligned} \tag{3.44}$$

- Sistema III: Possui ruído com distribuição não-gaussiana. Possui a seguinte equação:

$$x(n) = 0.9x(n-1) + (w(n))^3 \tag{3.45}$$

- Sistema IV: Inclui tanto não-linearidade quanto ruído não-gaussiano. Possui a seguinte equação:

$$x(n) = [x(n-1) + w(n)]^{\frac{1}{3}} \tag{3.46}$$

Todos os sistemas possuem $v(n)$ e $w(n)$ (ruídos de medida e ruídos de processo) como ruídos gaussiano branco com média zero e desvio padrão de 0.5. A única exceção é o sistema I, que possui desvio padrão de 0.2 e 1.0 para $w(n)$ e $v(n)$, respectivamente. Os outros 20 sistemas apresentam distribuições alternativas para os ruídos, não-linearidades nas funções de medidas, correlações entre os ruídos e várias combinações dessas características. A comparação da RN com FKD em cada um dos 4 sistemas é feita utilizando a média absoluta do erro de estimação, conforme mostrado na tabela 3.1. A melhora é calculada através da diferença percentual do erro do FKD e do erro da RN.

Tabela 3.1: Comparação da RN com o FKD, nos quatro sistemas

Sistema	**Erro do FKD**	**Erro da RN**	**Melhora da RN**
I	0,267824	0,276415	-3,2%
II	0,337429	0,286842	15%
III	0,280056	0,249324	11%
IV	0,266397	0,165234	38%

O sistema I foi o único que o FKD teve um razoável melhor desempenho. Em todos os outros a RN saiu-se substancialmente melhor. A maior melhora da RN deu-se em um sistema com $w(n)$ e $v(n)$ idênticos (100% de correlação). Com esses experimentos, mostra-se que a RN apresentada teve desempenho pouco inferior ao FK quando as hipóteses de Kalman são atendidas. Quando as hipóteses não são atendidas, a RN mostra um desempenho bastante superior em todos os casos. Quando as violações aumentam, aumenta o grau de melhora no desempenho. O grau de melhora depende do tipo exato e do grau da violação da hipótese.

3.4 Conclusões sobre o FK

As funções **f** e **h** (funções da dinâmica de estado e de medida, respectivamente) podem na prática variar com o tempo, de acordo com as características da maioria dos sistemas reais. Porém, na maioria das aplicações do FK, essas funções são constantes. Essa simplificação deve-se principalmente à grande dificuldade de se modelar a estatística de transição de estado. Então, descobrir vários desses modelos ao longo do tempo torna-se inviável. Essa dificuldade motiva o uso de redes neurais como processo do FK neste trabalho, pois as RN adaptam-se automaticamente a mudanças na função de transição. A própria necessidade de possuir uma função **f** definida é uma limitação do FK. Essa necessidade limita o campo de aplicações do FK, não podendo ser utilizado onde o modelo não é conhecido, como na predição de séries temporais. Outra limitação é a suposição que o ruído obedece distribuição gaussiana. Existem também limitações no tratamento de não-linearidades, sendo que o FKD não as trata e o FKE trata apenas as não-linearidades de primeira ordem.

Quando a covariância do erro de predição (**P**) aproxima-se de 0, o ganho de Kalman utilizará uma parcela menor da inovação. Neste caso, o erro já estará muito pequeno e as medidas devem ser "levadas menos em consideração". Se a covariância do ruído de medida **R** tender a 0, significa que as medidas são muito precisas e o ganho "considerará" mais a inovação. Por outro lado, se o ruído de medida for muito grande, a observação é pouco confiável e o filtro considera mais a predição antiga. Então o ganho de Kalman será diretamente proporcional à covariância do erro de estimativa do vetor de estados e inversamente proporcional à covariância do ruído de medida (WELCH; BISHOP, 2001). A expressão "considerar mais a inovação", neste caso, significa que a multiplicação com a matriz **K** (ganho de Kalman) dará um peso maior para a inovação (diferença da estimativa com a nova medida real) e um peso menor para a estimativa anterior.

As matrizes de covariância de ruído aparecem com argumento variável ($\mathbf{Q}(n)$ e $\mathbf{R}(n)$), ou seja, poderiam ter valores diferentes a cada instante de tempo. Em boa parte dos trabalhos, essas matrizes permanecem constantes, como em (WELCH; BISHOP, 2001). Porém, como as covariâncias do ruído influenciam diretamente no cálculo do Ganho de Kalman, covariâncias desatualizadas podem levar a uma atualização de estado não otimizada. Sobre o uso dos parâmetros, (WELCH; BISHOP, 2001) comenta que a matriz **R** é normalmente definida antes da execução do filtro, sendo obtida a partir de medidas. Com essas medidas, podem-se utilizar estatísticas para a calibragem do parâmetro. A estimação do parâmetro **Q** pode ser feita através de estatísticas de diferenças entre a saída do processo e a saída esperada. Experimentos práticos de configuração inicial das covariâncias de ruído serão mostrados no capítulo dos resultados desta dissertação.

4 PREDIÇÃO DE SÉRIES TEMPORAIS

Esse capítulo trata da Predição de Séries Temporais (PST), importantíssima tarefa para redes neurais e métodos estatísticos. O capítulo apresenta os conceitos, aplicações, métodos lineares de predição e a utilização de redes neurais.

4.1 Conceitos Iniciais

Nesta seção são apresentados os conceitos necessários para a apresentação da PST. Apresentam-se as definições, textual e matemática, de uma série temporal. Também são mostrados os objetivos da análise das séries, exemplos, uma introdução aos procedimentos de predição e o conceito de estacionariedade.

4.1.1 Definição de Série Temporal

Uma série temporal é qualquer conjunto de observações ordenadas no tempo, em instantes determinados (MORETTIN; TOLOI, 2004). Entre os elementos de uma série temporal, só varia o instante em que a observação é realizada. Os outros elementos, como fato e local das observações, permanecem constantes. Uma série temporal pode ser discreta ou contínua. A série será discreta, se o conjunto de observações for discreto no tempo e será contínua, se o conjunto de observações for contínuo. Grande parte das séries discretas é obtida da amostragem de séries contínuas e toda série contínua pode ser discretizada (OLIVEIRA, 2002). A grande maioria dos métodos de predição utiliza séries discretas ou discretizadas. A conversão de uma série contínua para discreta pode ser realizada pela medição de N pontos em um dado intervalo, com diferença de tempo igual entre os pontos. Outra forma de discretização, de um intervalo da série contínua, é a acumulação (ou agregação) de valores em subintervalos iguais.

Matematicamente, pode-se definir uma série temporal como uma seqüência de valores Y_1, Y_2,..., Y_T de uma variável Y nos instantes t_1, t_2,..., t_T. Y será, então, uma função de t, descrita por $Y = \mathrm{F}(t)$ (CROCE FILHO, 2000).

4.1.2 Aplicações

Como exemplos de séries temporais, pode-se citar:

- Valor diário de fechamento de uma certa ação da Bolsa de Valores (série discreta),
- Valores médios mensais de temperatura em uma certa cidade (série discreta, obtida pela média de amostragens de uma série contínua),
- Registro do nível de água em uma determinada represa (série contínua),

- Índice Nacional de Preços ao Consumidor (série discreta),
- Medida do nível de vibração em determinada posição de um equipamento (série contínua),
- Índice de precipitação atmosférica anual em determinada cidade (série discreta, obtida pelo somatório de um intervalo de uma série contínua),
- Faturamento anual de uma empresa (série discreta),
- Número médio anual de manchas solares (série discretizada),
- Número e intensidade média de furacões em uma região, em determinada época do ano (séries discretizadas).

Como pode-se perceber, pela variedade dos exemplos, existem séries temporais nas mais diversas áreas. Essas séries são encontradas abundantemente na natureza (meteorologia, astrofísica), nas ciências sociais (demografia, indicadores de qualidade de vida), na economia (mercado acionário, taxas de câmbio), na área médica (variação de níveis de substâncias no corpo, seqüências de produção de anticorpos, etc.), na área tecnológica (comportamento de sinais, sistemas dinâmicos, etc.) e em muitas outras áreas (CASTRO, 2001). Essa imensa quantidade de aplicações motiva muito a predição de séries temporais. Muitas dessas séries influenciam diretamente no futuro da humanidade e a predição de seus comportamentos pode significar grandes lucros de acionistas, um melhor atendimento a pessoas, organização estratégica de empresas ou até prevenção de catástrofes.

4.1.3 Objetivos da Análise de Séries Temporais

Dada uma série temporal $Y_1, ..., Y_T$, observada nos instantes $t_1, ..., t_T$, os objetivos da análise são (MORETTIN; TOLOI, 2004):

- Investigar o mecanismo gerador da série;
- Fazer previsões de valores futuros da série;
- Descrever o comportamento da série, observando tendências, ciclos e variações e construindo gráficos;
- Procurar periodicidades relevantes nos dados.

4.1.4 Procedimentos de Predição

Os procedimentos de predição de séries temporais, utilizados na prática, podem ser desde simples e intuitivos até complexos e quantitativos. No primeiro grupo, usa-se pouca ou nenhuma análise dos dados, enquanto no segundo pode-se analisar profundamente os dados, desenvolvendo-se teorias e modelos de comportamentos.

Em economia, há dois tipos de procedimentos para prever uma série: econométrico e de séries temporais. O procedimento econométrico é fortemente baseado na teoria econômica, utilizando muitas variáveis. O segundo é a análise pura da série, deixando os dados "falarem por si", sem se preocupar com o contexto e variáveis econômicas. Nesse caso, os modelos não precisam ter nenhuma relação com a teoria econômica, desde que apresentem bons resultados (MORETTIN; TOLOI, 2004).

4.1.5 Estacionariedade

Uma das suposições mais importantes para caracterizar uma série temporal é se ela é estacionária, isto é, se a série permanece ao redor de uma média constante, refletindo um equilíbrio estável (MORETTIN; TOLOI, 2004) (NUNES, 2003) (MANTOVANI, 2004). Tratando-se de séries reais, a maior parte delas apresenta alguma forma de não-estacionariedade. Por exemplo, as séries econômicas apresentam tendências, que podem ser positivas ou negativas. O caso mais simples de tendência é quando a série flutua em torno de uma reta, nesse caso tem-se uma tendência linear. Pode-se ter também não-estacionaridades explosivas, como o exemplo do crescimento de uma colônia de bactérias. Na figura 4.1 é mostrado um exemplo de uma série não-estacionária, com uma tendência linear crescente ao longo de toda a série, acrescida de múltiplas sub-tendências lineares temporárias.

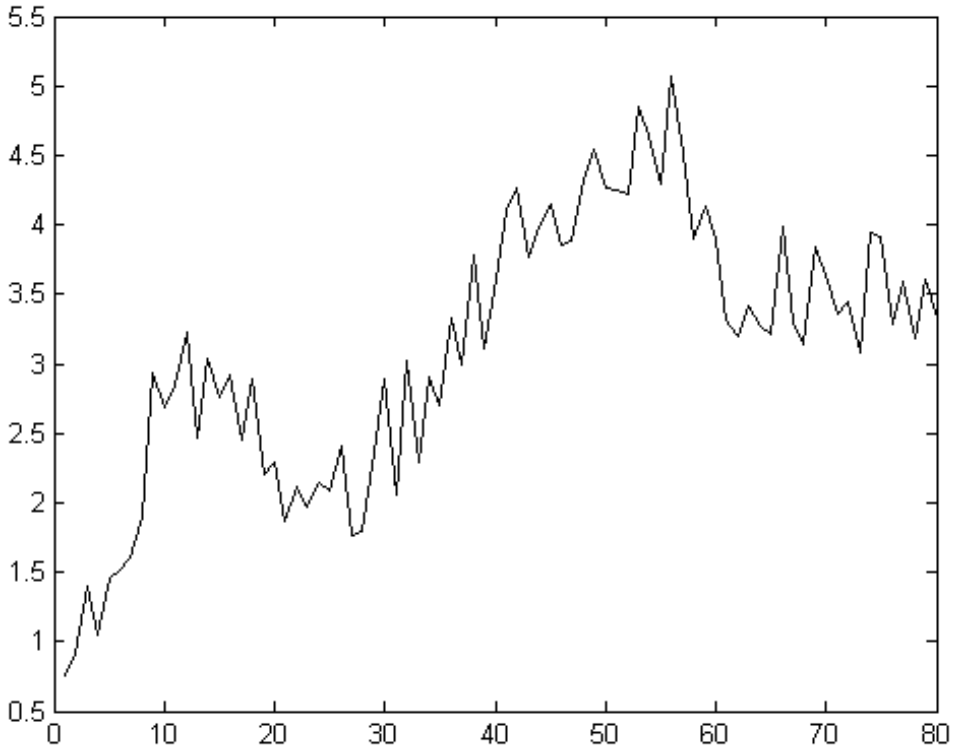

Figura 4.1: Série temporal não-estacionária

A maioria dos métodos de PST trata as séries como estacionárias, por isso normalmente usa-se o método das diferenças sucessivas, até obter-se uma série estacionária. A primeira diferença é o vetor de diferenças de cada valor da série original pelo seu valor anterior e é dada por:

$$\Delta Y_t = Y_t - Y_{t-1} \tag{4.1}$$

Onde:

- Y_t é o valor da série na posição t
- t varia de 2 até o tamanho da série.

A segunda diferença é calculada a partir da primeira diferença, utilizando-se o mesmo procedimento. Na figura 4.2 é mostrada a primeira diferença da série não-estacionária,

da figura 4.1. Pode-se perceber na figura que agora há uma série estacionária. Na grande maioria das séries, uma ou duas diferenças são suficientes para obter-se uma série estacionária(MORETTIN; TOLOI, 2004).

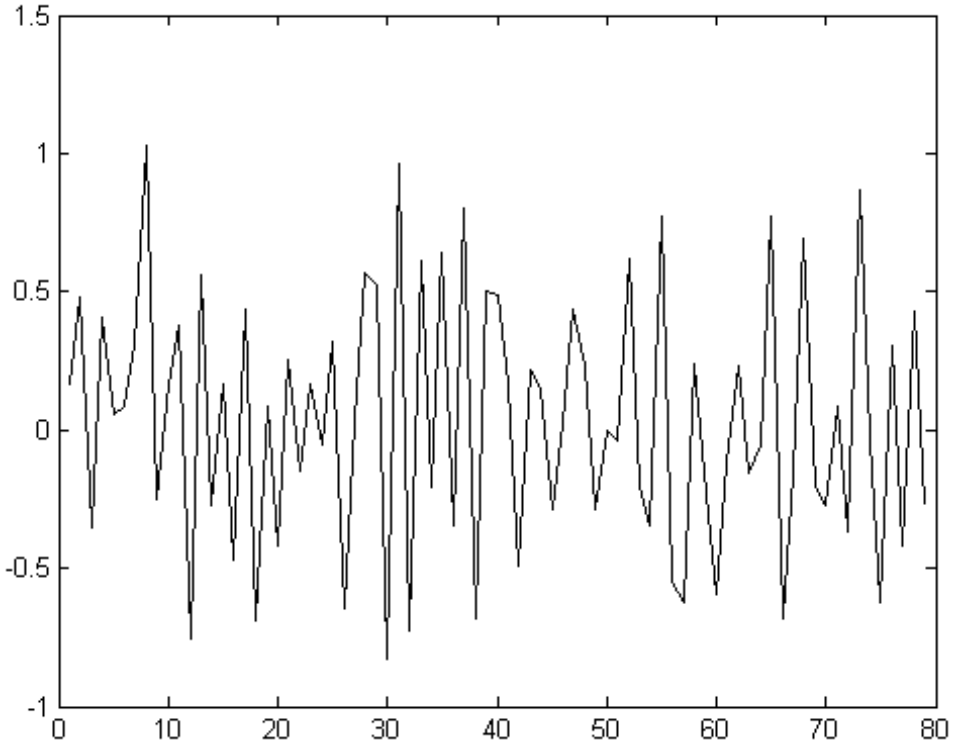

Figura 4.2: Primeira diferença da série temporal

4.2 Métodos Lineares de Predição de Séries Temporais

Aqui são apresentados os principais métodos de predição linear. A maior parte dos métodos apresentados é relativamente simples e consiste de filtros com combinações de médias simples e ponderadas. Maiores detalhes e comparações dos métodos dessa seção podem ser encontrados em (MORETTIN; TOLOI, 2004) (MORETTIN; TOLOI, 1981) e (OLIVEIRA, 2002).

4.2.1 Médias Móveis Simples

A técnica das Médias Móveis Simples (MMS) consiste em calcular a média aritmética das r observações mais recentes da série, na forma:

$$M_t = \frac{Y_t + Y_{t-1} + \cdots + Y_{t-r+1}}{r} \tag{4.2}$$

A estimativa M_t não leva em conta as observações mais antigas, fazendo a tendência variar suavemente com o tempo e esquecendo o passado mais distante. O termo *médias móveis* se deve à substituição da observação mais antiga pela mais recente, a cada passo de tempo. A previsão de valores futuros é dada pela última média móvel calculada:

$$\hat{Z}_t(h) = M_t \tag{4.3}$$

Onde h é o horizonte de previsão (quantidade de instantes a frente), então $\hat{Z}_t(h)$ representa a previsão de Z para o instante $t+h$. Uma boa escolha da quantidade de termos utilizada na média (r) é imprescindível para o bom desempenho da técnica de MMS. Se o valor de r for muito grande, a previsão acompanhará lentamente as mudanças de parâmetros. Se r for muito pequeno, a reação à mudança de parâmetro será muito rápida. Existem dois extremos:

- Se $r = 1$, então o valor mais recente da série é utilizado como previsão de todos os valores futuros. Esse tipo de predição é chamado de "método ingênuo".
- Se r é igual ao número de valores anteriores, a previsão será a média aritmética de todos os valores observados. Nesse caso, tem-se uma suavização muito grande, só utilizada quando a série é altamente aleatória.

Conclui-se que o valor de r é proporcional ao tamanho da aleatoriedade da série. Um procedimento adequado é selecionar um valor de r que dá a melhor previsão de um passo das observações já obtidas. Isso equivale a encontrar um valor que minimize:

$$S = \sum_{t=\ell+1}^{n} \left(Z_t - \hat{Z}_{t-1}(1)\right)^2 \tag{4.4}$$

Onde ℓ é escolhido de forma que o valor inicial não influencie a previsão.
As principais vantagens das MMS são:

- Aplicação simples da técnica;
- Pode-se aplicar mesmo quando se tem um número pequeno de observações;
- Permite grande flexibilidade devido à variação do parâmetro r de acordo com as características da série.

E as desvantagens são:

- Utilização somente na predição de séries estacionárias;
- Necessidade de armazenamento de pelo menos $r-1$ observações;
- Dificuldade em determinar o valor de r.

4.2.2 Alisamento Exponencial Simples

A técnica de Alisamento Exponencial Simples (AES) representa uma média ponderada, que dá pesos maiores às observações mais recentes da série. A AES é descrita por:

$$\begin{aligned} &\tilde{Z}_t = \alpha Z_t + (1-\alpha)\tilde{Z}_{t-1} \\ &\tilde{Z}_0 = Z_1 \\ &t = 1, \cdots, N \end{aligned} \tag{4.5}$$

Onde $\tilde{Z}_t$ é chamado de valor exponencialmente alisado e α é a constante de alisamento, com $0 \leq \alpha \leq 1$. Expandindo a equação de $\tilde{Z}_t$, tem-se:

$$\tilde{Z}_t = \alpha Z_t + \alpha(1-\alpha)Z_{t-1} + \alpha(1-\alpha)^2 Z_{t-2} + \cdots \tag{4.6}$$

Percebe-se na equação, que as observações mais recentes recebem pesos maiores, eliminando uma das desvantagens do método de MMS. A previsão de todos os valores futuros é dada pelo último valor exponencialmente alisado:

$$\hat{Z}_t(h) = \tilde{Z}_t \tag{4.7}$$

Quanto maior o valor de α, maior será a importância dada às observações recentes. Se o valor de α for muito pequeno, pesos maiores serão atribuídos às observações passadas e com isso as flutuações aleatórias do presente exercerão um peso menor no cálculo da predição. O valor de α é ajustado de acordo com a aleatoriedade da série; quanto mais aleatoriedade tiver na série, menor deverá ser o valor de α. A variação de α é análoga (e inversa) à variação do parâmetro r no MMS. Uma maneira simples de calcular o valor de α é análogo ao descrito no MMS, utilizando a melhor predição a um passo das observações já obtidas.

As principais vantagens do AES são:

- Fácil entendimento;
- Aplicação não dispendiosa;
- Grande flexibilidade pela possibilidade de variação da constante de suavização α;
- Necessidade de armazenar apenas Z_t, $\tilde{Z}_t$ e α;

Como desvantagem do AES tem-se a dificuldade em estimar o parâmetro α, podendo ser solucionada pela suavização exponencial adaptativa (OLIVEIRA, 2002).

4.2.3 Alisamento Exponencial Linear de Brown

As Médias Móveis Simples e o Alisamento Exponencial Simples são as mais simples técnicas de suavização e são adequadas para estimar o valor de um único coeficiente em processos localmente constantes (BROWN, 1963). As técnicas de alisamento de Brown aplicam-se também à modelos que não são localmente constantes. A técnica de Alisamento Exponencial Linear de Brown (AELB) (BROWN, 1963) consiste em calcular um segundo valor exponencialmente alisado. A formulação matemática tem a forma:

$$\begin{aligned}
&\tilde{\tilde{Z}}_t = \alpha\tilde{Z}_t + (1-\alpha)\tilde{\tilde{Z}}_{t-1} \\
&\tilde{\tilde{Z}}_1 = Z_1 \\
&\tilde{Z}_t = \alpha Z_t + (1-\alpha)\tilde{Z}_{t-1} \\
&\tilde{Z}_0 = Z_1
\end{aligned} \tag{4.8}$$

Supondo que a tendência seja linear, a previsão será feita da seguinte forma (OLIVEIRA, 2002):

$$\begin{aligned}
&\hat{Z}_t(h) = a_{1,t} + a_{2,t}h \\
&a_{1,t} = 2\tilde{Z}_t - \tilde{\tilde{Z}}_t \\
&a_{2,t} = \frac{\alpha}{1-\alpha}\left(\tilde{Z}_t - \tilde{\tilde{Z}}_t\right)
\end{aligned} \tag{4.9}$$

Onde $a_{1,t}$ é estimativa do intercepto (ponto que cruza o eixo das ordenadas) e $a_{2,t}$ é a estimativa da tendência (inclinação da reta). Da mesma forma que as técnicas anteriores, o α também pode ser calculado a partir da melhor predição de um passo.

4.2.4 Alisamento Exponencial Quadrático de Brown

O Alisamento Exponencial Quadrático de Brown (AEQB) (BROWN, 1963) é semelhante ao AELB, com a diferença que a tendência se apresenta de forma quadrática. Então tem-se um terceiro alisamento:

$$\begin{aligned} &\tilde{Z}_t = \alpha Z_t + (1-\alpha)\,\tilde{Z}_{t-1} \\ &\tilde{\tilde{Z}}_t = \alpha \tilde{Z}_t + (1-\alpha)\,\tilde{\tilde{Z}}_{t-1} \\ &\tilde{\tilde{\tilde{Z}}}_t = \alpha \tilde{\tilde{Z}}_t + (1-\alpha)\,\tilde{\tilde{\tilde{Z}}}_{t-1} \\ &t = 2, \cdots, N \end{aligned} \tag{4.10}$$

A predição é feita da seguinte forma:

$$\begin{aligned} &\hat{Z}_t(h) = a_{1,t} + a_{2,t}h + a_{3,t}h^2 \\ &a_{1,t} = 3\tilde{Z}_t - 3\tilde{\tilde{Z}}_t + \tilde{\tilde{\tilde{Z}}}_t \\ &a_{2,t} = \frac{\alpha}{2(1-\alpha)^2}\left[(6-5\alpha)\,\tilde{Z}_t - 2\,(5-4\alpha)\,\tilde{\tilde{Z}}_t + (4-3\alpha)\,\tilde{\tilde{\tilde{Z}}}_t\right] \\ &a_{3,t} = \left(\frac{\alpha}{1-\alpha}\right)^2\left(\tilde{Z}_t - 2\tilde{\tilde{Z}}_t + \tilde{\tilde{\tilde{Z}}}_t\right) \end{aligned} \tag{4.11}$$

A determinação de α é da mesma forma dos métodos anteriores. O AEQB também pode ser generalizado para ordens de tendências maiores.

4.2.5 Modelos de Auto-regressão

Os modelos de Auto-Regressão (AR) supõem que os valores da série sejam linearmente relacionados com seus próprios valores defasados. Um modelo auto-regressivo de ordem k será chamado de AR(k) e pode ser descrito por:

$$\begin{aligned} &Z_t = w_1 Z_{t-1} + w_2 Z_{t-2} + \cdots + w_k Z_{t-k} + e_t \\ &t = 1, 2, \cdots, N \end{aligned} \tag{4.12}$$

Onde:

- w são os pesos atribuídos a cada uma das observações passadas;
- e_t é o ruído no tempo t;
- $Z_{t-1}, \cdots, Z_{t-k}$ são os valores anteriores da série utilizados na regressão.

Para que o modelo possa ser aplicado são necessárias as seguintes suposições (MORETTIN; TOLOI, 2004):

- e_t tem média zero e variância σ_e^2;
- $Z_t, \cdots, Z_{t-k}$ são vistos como seqüências de constantes;
- As raízes do polinômio abaixo são em módulo menores que um, garantindo a estabilidade do modelo:

$$x^k + \sum_{j=1}^{k} w_j x^{k-j} \tag{4.13}$$

As estimativas dos pesos w são feitas de acordo com os mínimos quadrados dos erros, então tem-se:

$$\sum_{t=k+1}^{n} e_t^2 = \sum_{t=k+1}^{n} \left(Z_t - \left(w_1 Z_{t-1} + w_2 Z_{t-2} + \cdots + w_k Z_{t-k}\right)\right)^2 \tag{4.14}$$

Para calcular os pesos, (MORETTIN; TOLOI, 2004) apresenta o operador auto-regressivo estacionário de ordem k:

$$w(B) = 1 - w_1 B - w_2 B^2 - \cdots - w_k B^k \tag{4.15}$$

Onde B é um operador de translação para o passado, definido por:

$$BZ_t = Z_{t-1}, \quad B^m \; Z_t = Z_{t-m} \tag{4.16}$$

Após estimar-se os coeficientes w adequados, a previsão pode ser feita por:

$$\hat{Z}_t(h) = \hat{w}_1 Z_{t+h-1} + \hat{w}_2 Z_{t+h-2} + \cdots + \hat{w}_k Z_{t+h-k} \tag{4.17}$$

4.2.6 Modelos ARIMA

O principal exemplo de métodos lineares são os modelos ARIMA (*Autoregressive Integrated Moving Averages* ou modelos Auto-regressivos Integrados de Médias Móveis) (BOX; JENKINS; REINSEL, 1994) (MORETTIN; TOLOI, 2004) (MANTOVANI, 2004). Os modelos ARIMA são uma combinação de três componentes, interpretados como filtros: o componente auto-regressivo, o componente de médias móveis e o filtro de integração. Nem sempre serão necessárias essas três características, podendo haver uma combinação dessas componentes.

O preditor ARIMA pode ser configurado para realizar previsões de acordo com três casos de modelos de séries temporais: processos lineares estacionários (PLE), processos lineares não-estacionários homogêneos (PLNEH) e processos de memória longa (PML). PLE é a classe geral, os outros dois casos são ajustados para essa classe. Os PLNEH são uma especialização dos PLE, supondo que o mecanismo gerador da série produz erros auto-correlacionados e que a não-estacionariedade seja apenas em nível ou em inclinação (desse caso excluem-se as não-estacionariedades explosivas). Essas séries podem gerar séries lineares com o método das diferenças (geralmente primeira ou no máximo segunda diferença), como mostrado na seção sobre estacionariedade. Os PML são processos estacionários, mas que possuem uma função de autocorrelação com decaimento muito lento, necessitando de uma diferença fracionária para tornar-se "puramente estacionária". Essa diferença varia entre 0 e $0,5$, necessitando do uso do modelo ARIMA com todas as componentes, com ordens p, d e q: ARIMA(p,d,q). Onde:

- p é a ordem dos modelos auto-regressivos (AR(p));
- q é a ordem dos processos de médias móveis (MA(q));
- d é a ordem de não-estacionariedade do modelo.

O modelo geral ARIMA para descrever séries temporais é dado por:

$$\varphi(B) Z_t = w(B) (1-B)^d Z_t = \theta_0 + \theta(B) a_t \tag{4.18}$$

O termo $w(B)$, chamado operador auto-regressivo, é assumido como estacionário e representado por:

$$w(B) = 1 - w_1 B - w_2 B^2 - \cdots - w_p B^p \tag{4.19}$$

O termo $w(B)(1-B)^d$ torna-se um operador não-estacionário e é chamado de operador auto-regressivo generalizado. O operador de médias móveis $\theta(B)$, que é somado ao termo constante θ_0, é represesentado por:

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q \tag{4.20}$$

Como descrito inicialmente, pode-se ter submodelos do ARIMA. Além de AR(p) e MA(q) tem-se os processos autoregressivos e de médias móveis de ordem p e q (ARMA(p,q)). O modelo ARIMA passa por um ciclo iterativo para sua construção, no qual a escolha da estrutura do modelo é baseada nos próprios dados, descobrindo-se quais partes do ARIMA são necessárias e quais parâmetros serão usados. Os estágios desse ciclo são:

1. Especificação de uma classe geral de modelos para análise;
2. Identificação de um modelo, com base na análise de autocorrelações, autocorrelações parciais e outros critérios;
3. Estimação dos parâmetros do modelo identificado;
4. Validação do modelo ajustado, através de uma análise de resíduos, para saber se ele é adequado para os objetivos (predição).

No passo 4 do ciclo, caso descubra-se que o modelo não é adequado, o ciclo é repetido, voltando para a fase de identificação. Pode-se identificar vários modelos, e depois escolher o melhor. Se o objetivo é a predição, será escolhido o modelo que oferecer o menor erro médio quadrado (MSE) de previsão.

Geralmente os modelos ARIMA contêm um número pequeno de parâmetros e as predições são bastante precisas. Uma dificuldade da aplicação do método é que ele requer experiência e algum conhecimento além do uso automático do algoritmo (MORETTIN; TOLOI, 2004). Em (MORETTIN; TOLOI, 1981) é mostrada uma comparação dos modelos ARIMA com outros métodos de predição, como AES e AR, em várias séries temporais reais. Os modelos ARIMA apresentaram os melhores resultados, seguidos da técnica de auto-regressão. O AES mostrou-se menos preciso, uma vez que este é mais adequado a séries localmente constantes e as séries testadas apresentavam tendências ou sazonalidade. A AR mostrou-se adequada apenas às séries com grande quantidade de amostras. Em séries estacionárias o AES pode ter resultados melhores que o ARIMA, como mostrado em um exemplo de (OLIVEIRA, 2002).

4.3 Predição de Séries Temporais com Redes Neurais

Os modelos convencionais (estatísticos lineares) necessitam que um conjunto bem definido de parâmetros seja conhecido a priori. Porém, em boa parte das situações reais, essas características não são conhecidas inicialmente. As redes neurais (RN) possuem bastante vantagens nesse quesito, pela sua grande adaptabilidade, conseguindo extrapolar padrões a partir dos dados existentes. As RNs também adaptam o seu comportamento a medida que novos dados são introduzidos, sem a necessidade de alterar a sua estrutura (CORTEZ, 1997).

As técnicas convencionais também consistem em procurar dentro de um conjunto limitado de modelos, aqueles que melhor representam os processos geradores das séries. Cada análise representa assumir uma estrutura para os dados, modelo e parâmetros, testando a validade dessa estrutura repetidas vezes, uma tarefa muito custosa e às vezes inviável. As RNs apresentam grandes vantagens pois aprendem os padrões subjacentes nos dados, apresentando resultados muito melhores que os métodos estatísticos tradicionais quando o processo regente dos dados é desconhecido, não-linear ou não-estacionário (CASTRO, 2001).

A própria estrutura das séries temporais beneficia o uso de redes neurais. Nas formas mais clássicas de representação, o cálculo do próximo instante de uma série temporal é descrita por:

$$y(k) = \sum_{n=1}^{T} a(n) y(k-n) + e(k) = \hat{y}(k) + e(k) \tag{4.21}$$

Onde:

- $y(k)$ é o valor atual a ser calculado;
- T é o número de termos anteriores que são considerados no cálculo do valor atual;
- $y(k-n)$ representa cada um dos T valores anteriores da série;
- $a(n)$ é o peso dado a cada observação passada;
- $e(k)$ é o erro do cálculo.

O erro é assumido ser ruído branco, pelos construtores das técnicas de regressão linear. O artigo de (WAN, 1994) indica a existência de uma não-linearidade na definição acima, com um mapeamento diferencial em relação a cada um dos termos anteriores. A auto-regressão não-linear fica na forma $y(k) = g[y(k-1), y(k-2), ..., y(k-T)]$ e modela a série exatamente, assumindo que a mesma não tenha ruído. A indicação de características não-lineares nas séries temporais motiva o uso de redes neurais. A rede aproxima a função ideal $g(\cdot)$. Uma rede MLP (Multi camadas) alimentada adiante, com um número suficiente de neurônios, é capaz de aproximar uma função uniformemente contínua (CYBENKO, 1989) (HAYKIN, 2001a).

A maior parte das RNs utilizadas na área de séries temporais é do tipo alimentada adiante ou *feedforward*, com algoritmos derivados do *backpropagation*. Existem muitas aplicações de redes desse tipo em mercados financeiros, mostrando bons resultados, inclusive melhores que o modelo ARIMA. (CORTEZ, 1997) comenta sobre experimentos em que as RNs obtêm melhores resultados que os métodos lineares, em especial em predições de mais longo prazo. O trabalho comenta que as redes alimentadas adiante, com conexões de atalho conseguem funcionar como um super conjunto de modelos ARIMA, pois combinam componentes lineares (gerados pelas conexões de atalho) e não-lineares (proporcionados pelas camadas intermediárias). O bom desempenho das RNs depende da estrutura da rede, dos parâmetros utilizados e da natureza da série temporal.

4.3.1 Histórico de PST com RN

A utilização de redes neurais na PST tornou-se mais intensa no início dos anos oitenta, tendo como principal objetivo completar a lacuna deixada pelos métodos estatísticos convencionais quanto a séries não-lineares. As primeiras aplicações foram no mercado financeiro, onde comprovadamente os métodos de alisamento eram incapazes de prever

rápidas e pequenas flutuações nos valores dos índices. As primeiras aplicações de RNs para prever valores de ações frustaram as grandiosas expectativas existentes, mas aos poucos foram sendo descobertas circunstâncias e metodologias que fizeram as redes surgirem como boas alternativas também para esse tipo de aplicação (CORTEZ, 1997).

O interesse de pesquisadores de redes neurais para predição de séries temporais é ainda mais antigo. Em 1964, Ho aplicou uma rede linear adaptativa em estudos de previsão climática. Mais tarde, em 1987, Lapedes e Farber aplicaram uma rede neural não-linear para descobrir a relação entre pontos sucessivos de séries temporais geradas computacionalmente (CASTRO, 2001).

4.3.2 Concursos de PST

Surgiu no Santa Fé Institute (localizado em New Mexico, USA), em 1990, a idéia de realização de uma competição para comparação de desempenhos de métodos para PST. A motivação para a competição foi a dificuldade em encontrar literatura consistente, das diversas áreas do conhecimento envolvidas na PST (CASTRO, 2001). Mesmo sendo uma iniciativa aparentemente pouco científica, a idéia foi bem aceita pela comunidade científica e realizou-se o concurso, patrocinado pelo Santa Fé Institute. A competição contou com um grupo de consultores das diversas áreas de conhecimento envolvidas na PST, como economia, física, biologia, astrofísica, estatística e sistemas dinâmicos. O objetivo era organizar a discussão de tópicos importantes de PST, difundir novas técnicas e criar padrões de comparação para técnicas no futuro. Em 1992 aconteceu um encontro para apresentar os resultados do concurso, chamado *NATO Advanced Research Workshop*.

O maior interesse dos métodos apresentados foi na PST, baseada principalmente em modelos não-lineares. O grande destaque do concurso foram os métodos conexionistas (baseados em redes neurais), com maior número de participantes e as melhores predições. (WAN, 1994) também comenta o sucesso das RN no concurso, onde estas foram imparcialmente contrastadas com uma variedade de outros métodos, e justifica o uso de RNs para prever séries temporais.

4.4 Conclusões do Capítulo

Os modelos de Box e Jenkins (ARIMA) tiveram frutíferas aplicações nas áreas sociais, econômicas, engenharias, comércio internacional, etc. A grande vantagem desse método está em previsões para curtos espaços de tempo. Os modelos ARIMA são muito tradicionais e existem muitos estudos mostrando suas vantagens (principalmente comparando com outros métodos estatísticos mais simples). Existem casos que as técnicas simples como MMS e AES são indicadas, como séries estacionárias. A utilização de um modelo em detrimento de outro depende muito da aplicação em questão e também da área de origem dos participantes do projeto.

As redes neurais foram inicialmente pouco valorizadas, situação que foi sendo amenizada devido ao seu grande sucesso em competições para avaliação de desempenho de métodos de PST. Mesmo com resultados favoráveis das RN, grande parte da literatura ainda sub-valoriza esses métodos nas comparações, devido ao fato de as RN serem pouco "explicáveis" e de que boa parte dos pesquisadores preferem métodos com extensa teoria sobre seu funcionamento, em detrimento da obtenção de melhores resultados. Redes neurais possuem grandes vantagens em dados de situações reais, onde o comportamento do processo é desconhecido. As vantagens das RNs são a sua adaptabilidade a modelos desconhecidos, a séries não-estacionárias e com grandes não-linearidades. Uma RN

modela as não-linearidades da série, comparando-se a múltiplos modelos ARIMA juntos, devido à presença de componentes lineares e não-lineares e adaptação aos processos, sem necessidade de construção e validação de inúmeros modelos.

5 TRABALHOS CORRELACIONADOS

Este capítulo de trabalhos correlacionados trata principalmente das abordagens híbridas, em que uma rede neural é utilizada conjuntamente com um Filtro de Kalman (FKE ou outra variante). O treinamento de RNs com FK também é abordado, bem como o uso de redes para ajustar parâmetros do filtro. Ao final, compara-se o presente trabalho com os demais trabalhos correlacionados.

5.1 Extensão do Filtro de Kalman com uma Rede Neural

Esta seção trata dos trabalhos em que realmente ocorre uma hibridização entre a rede neural e o Filtro de Kalman (normalmente o FKE). A RN é utilizada como uma extensão do filtro, tentando prever o erro deste para melhorar os resultados.

5.1.1 Primeiros Trabalhos com RN Prevendo o Erro do FKE

O artigo de (VEPA, 1993) lança as idéias de uma abordagem híbrida, com uma rede neural estimando o erro de um Filtro de Kalman Estendido. A aplicação é em estimação da posição de veículos, que é um problema com muitas particularidades. A solução popular é baseada na estimação do *quatérnio*, elemento de um conjunto que representa um corpo exceto pela propriedade da multiplicação, representado pela soma $a+bi+cj+dk$, onde a, b, c e d são números reais. Essa estimativa é feita com um FKE. Porém, a predição com FKE somente é adequada se a incerteza do sensor de posição puder ser modelada de maneira muito próxima a um ruído branco ou ruído colorido. Em muitos casos não é possível modelar dessa maneira.

O trabalho de (VEPA, 1993) utiliza uma abordagem híbrida com um modelo particular para estimativa de posição. Além do aprendizado dos pesos, essa abordagem cooperativa também adapta as macro estruturas da RN. A RN acaba sendo moldada em função do FKE. A justificativa para utilização da RN na forma híbrida é que o FKE isoladamente necessita que o modelo não-linear seja diferenciado com a estatística totalmente conhecida a priori. A arquitetura da RN também é híbrida. A primeira camada é uma rede retropropagada e a segunda camada é propagada adiante. A primeira camada visa representar o estado interno do observador e a segunda, o estado das relações de saída.

A modelagem híbrida apresenta o seguinte formato, seguindo a estrutura do FKE. A etapa de previsão é idêntica ao FKE. A etapa de atualização é dividida em dois estágios: o primeiro estágio é idêntico à atualização do FKE; o segundo baseia-se numa melhora da estimativa com uma RN dinâmica. Resumindo, esse modelo é um FKE com uma RN para melhorar seus resultados, tentando prever os erros do filtro. A estimativa de próximo

estado é dada por (seguindo a notação original do artigo):

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)\,\mathbf{r}(k+1) + \mathbf{F}\left[\mathbf{r}(k+1), \hat{\mathbf{x}}(k+1|k), \mathbf{w}(k+1)\right] \tag{5.1}$$

Onde $\hat{\mathbf{x}}(k+1|k)$ é a estimativa das quatro posições do vetor quatérnio; $\hat{\mathbf{x}}(k+1|k+1)$ é a estimativa corrigida para o vetor; $\mathbf{r}(k+1)$ é a medida de erro no instante k (inovação); $\mathbf{K}(k+1)$ é Ganho de Kalman, baseado no FKE; $\mathbf{F}[\cdot]$ é a estimativa corrigida, obtida pela rede neural e $\mathbf{w}(k+1)$ é o vetor de pesos usado na RN.

O artigo de Vepa não comenta sobre o treinamento da RN e demais detalhes da ligação da RN com o FKE. O autor comenta o sucesso da aplicação da técnica apenas quando o FKE apresenta moderados erros de estimação. Mesmo de forma suscinta, esse artigo lança idéias que serão melhoradas posteriormente no *Neural Extended Kalman Filter*, descrito a seguir.

5.1.2 Neural Extended Kalman Filter

O Neural Extended Kalman Filter (NEKF) é um modelo que utiliza uma RN para prever o erro de um FKE, de maneira *on-line*, com a RN sendo treinada por outro FKE. O artigo inicial de (STUBBERUD; LOBBIA; OWEN, 1995) mostra um neuro-observador, que é um FKE que tem a atualização de estados melhorada por uma RN.

Sabe-se que é necessário para o FKE o conhecimento a priori de toda a estrutura do modelo estatístico, para fazer a estimativa dos estados e cálculo das jacobianas. Em grande parte dos casos reais o modelo é parcialmente ou totalmente desconhecido. O NEKF trata de sistemas parcialmente conhecidos (com função original f), com uma função $\hat{f}$ que aproxima o sistema real. A diferença entre o sistema real e a aproximação terá um erro representado por:

$$\varepsilon_k = f_k(x_k, u_k) - \hat{f}_k(x_k, u_k) \tag{5.2}$$

O NEKF utiliza uma RN para estimar o erro (ε_k), que é a diferença entre o verdadeiro modelo e aquele encontrado pela implementação padrão do FKE. A RN pode ser multi-camadas alimentada adiante, representada por $g_k(x_k, u_k, w_k)$, onde w_k são os pesos passados para a rede. Então a equação de estado resultante será:

$$x_{k+1} = \hat{f}_k(x_k, u_k) + g_k(x_k, u_k, w_k) \tag{5.3}$$

Para calcular a covariância do erro, também acrescenta-se a estimativa do erro, feita pela RN. A matriz de covariância segue o formato padrão do FKE, sendo acrescida da jacobiana da saída da RN em função do estado. A arquitetura da RN utilizada no artigo (STUBBERUD; LOBBIA; OWEN, 1995) foi uma rede padrão multi-camadas retropropagada. No experimento do artigo utilizou-se 3 camadas. O experimento mostrado consiste na comparação do FKE com o neuro-observador em um sistema altamente não-linear. As equações do sistema original são:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}(k+1) = \begin{matrix} x_2(k) \\ 2\left(0.5 - e^{-(x_1(k)+x_2(k)+u(k))^2}\right) \end{matrix} \tag{5.4}$$

$$z(k) = x_2(k) \tag{5.5}$$

A parte modelada do neuro-observador (composta pelo FKE) é baseada no modelo de equações de referência abaixo. Essas equações representam a parte conhecida do sistema.

$$\mathbf{x}(k+1) = \begin{bmatrix} 0 & 1 \\ 3/32 & 1/4 \end{bmatrix}\mathbf{x}(k) + \begin{bmatrix} 0 \\ 1/4 \end{bmatrix} r(k) \tag{5.6}$$

$$z(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{x}(k) \tag{5.7}$$

Comparou-se o neuro-observador com o FKE, aplicados no sistema de equações acima. Na figura 5.1 é mostrado o comportamento do previsor apenas com a utilização do FKE. Na figura 5.2 é mostrado o comportamento do neuro-observador, formado pelo FKE e pela RN. Pode-se observar um grande acréscimo no desempenho, com a utilização da RN. O custo computacional do modelo inicial do neuro-observador é discutido em (STUBBE-RUD; OWEN, 1998). Uma nova versão do neuro-observador é proposta nesse artigo para diminuir o custo e viabilizar o treinamento *on-line*, sendo utilizadas RN mais simples, com menos neurônios.

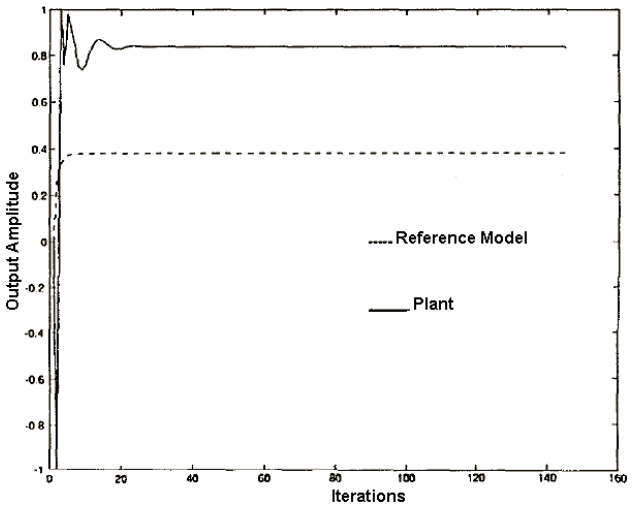

Figura 5.1: Previsão do sistema não-linear sem o Neuro-observador

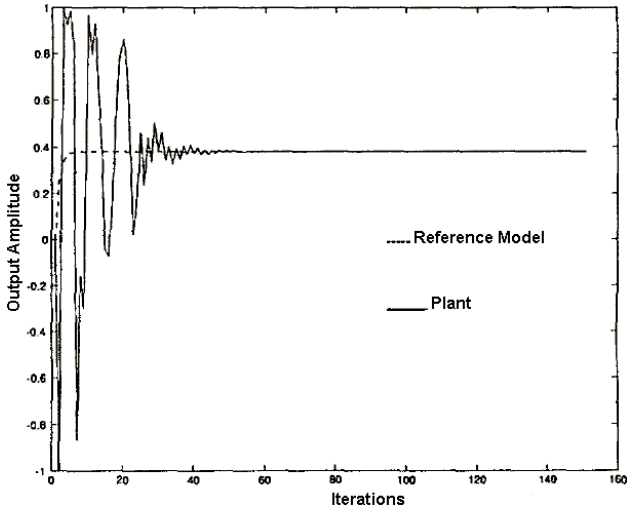

Figura 5.2: Previsão do sistema não-linear com o Neuro-observador

5.1.3 Usos do NEKF

As principais aplicações do NEKF são o acompanhamento de trajetórias de alvos, cálculo de distâncias de mísseis, rastreamento de projéteis, etc. O NEKF é empregado nesses tipos de problemas devido aos seus objetivos iniciais e à sua estrutura, pois todo o projeto foi financiado por organizações militares dos Estados Unidos.

5.1.3.1 Perseguição de Alvos

A perseguição de alvos e a interceptação (mostrada na subseção seguinte) são semelhantes, sendo que para a interceptação é necessário perseguir (rastrear) o alvo. Os dois tipos de usos são mostrados separadamente pelo enfoque mais abrangente dado pela perseguição de alvos e pela utilização conjunta com a técnica de Interação com Múltiplos Modelos (IMM). O uso de NEKF com IMM na perseguição de alvos é descrito por (OWEN; STUBBERUD, 1999) e (OWEN; STUBBERUD, 2003).

A técnica de IMM proporciona uma estrutura flexível e adaptativa para estimação de estados. A estrutura é formada por *N* modelos (podendo cada modelo ser um FKE ou um NEKF, por exemplo) rodando em paralelo. Cada modelo pode conter um diferente sistema de equações de transição de estados, modelo de observação (medidas), dimensão do vetor de estados e tipo de ruído de processo. Combinando o NEKF com IMM, os autores projetam um estimador muito robusto. O NEKF IMM, descrito em (OWEN; STUBBERUD, 2003), utiliza 3 modelos: dois deles utilizam velocidade constante, com baixo e alto ruído, respectivamente; o terceiro modelo é o NEKF. O NEKF IMM combina a robustez e intercâmbio entre modelos (do IMM) com a capacidade de aprendizado on-line de manobras do NEKF.

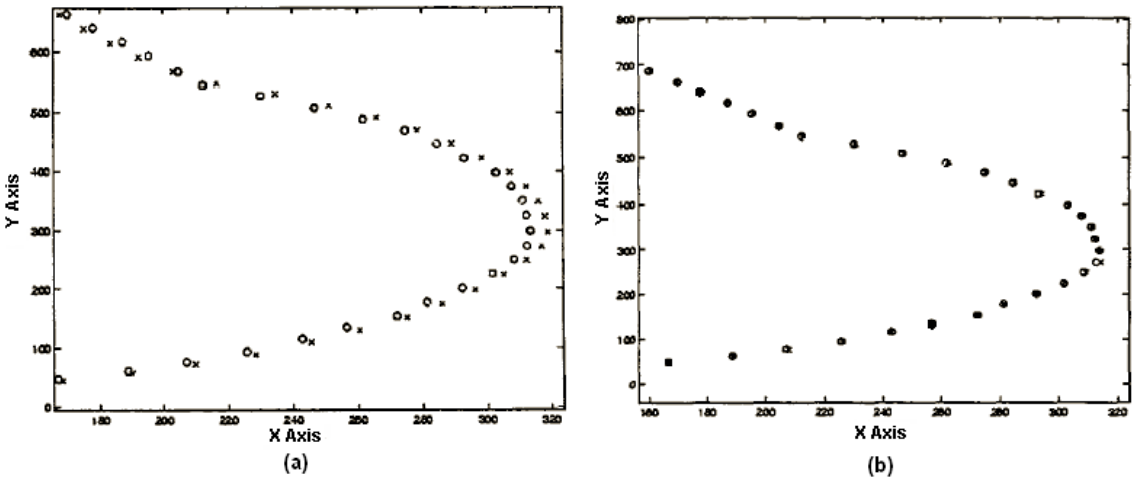

Figura 5.3: Acompanhamento da trajetória do alvo: (a) com o método da "linha reta" (b) com o método NEKF IMM

Um dos experimentos para validação do NEKF é mostrado em (OWEN; STUBBERUD, 1999), que é a perseguição do alvo em uma manobra num espaço bidimensional. A figura 5.3(a) mostra a predição com o "modelo da linha reta", que calcula a próxima posição com base na direção anterior. Na figura os círculos representam as medidas (com ruído) e os x's representam as estimativas. Pode-se perceber que o acompanhamento da manobra é retardado. Na figura 5.3(b) são mostrados os resultados de predição do NEKF IMM para a mesma manobra. Como pode-se ver na figura, ocorre uma significativa melhora na perseguição.

O artigo (OWEN; STUBBERUD, 2003) mostra os resultados do NEKF IMM em uma série de *benchmarks* de perseguição de alvos que se deslocam em três dimensões (aéreos). Os resultados mostram bons resultados para problemas difíceis, atestando a eficiência dessas técnicas (NEKF e NEKF IMM) para problemas de rastreamento *on-line*.

5.1.3.2 Interceptação de Alvos

A aplicação do NEKF para interceptação de alvos é descrita por (STUBBERUD; KRAMER, 2005). A interceptação de alvos é muito utilizada na robótica, sistemas espaciais e para mísseis de defesa. Alguns sinais do alvo são utilizados, como posição, ângulo e velocidade. Esses sinais são fornecidos por um modelo de trajetória. Geralmente o verdadeiro modelo é desconhecido e o sistema de interceptação dispõe apenas de rastros do alvo, fornecido por sensores. Como esses rastros (informações do alvo) são muito ruidosos, o sistema deve dispor de um modelo de movimento do alvo. Se uma manobra não for corretamente identificada, o desempenho do rastreamento será muito prejudicado. O NEKF visa fornecer esse modelo.

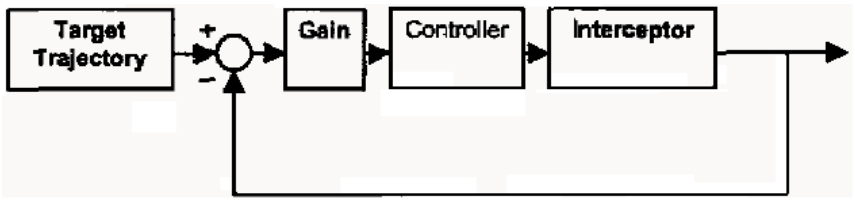

Figura 5.4: Sistema de controle para a interceptação de alvos com o NEKF

O sistema interceptador é mostrado na figura 5.4. O sistema recebe a estimativa de predição do alvo, calcula o ajuste (*Gain*), passa para o controlador (NEKF), que determina a posição e velocidade do alvo. O NEKF calcula a posição e velocidade com a maior precisão possível, pois são informações fundamentais para que o alvo possa ser interceptado.

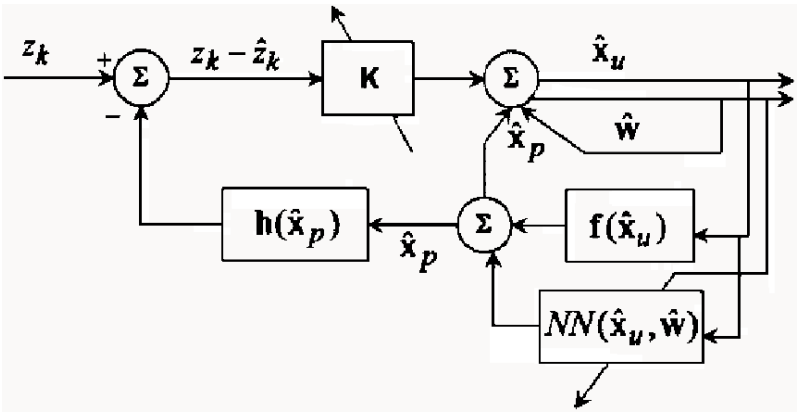

Figura 5.5: Modelo do Neural Extended Kalman Filter

O NEKF é tanto o preditor do estado como o treinador da RN, sendo que ambas tarefas utilizam a mesma informação de medida. O NEKF faz essa estimativa com a RN atuando como controle. O modelo do NEKF, utilizado em (STUBBERUD; KRAMER, 2005) é mostrado na figura 5.5. Pode-se observar na figura, que o mesmo erro utilizado para melhorar o estado também é usado para treinar a RN. Os resultados do artigo mostram que o NEKF melhora as estimativas de estado na presença de erros de modelagem, obtendo resultados muito melhores que o FKE padrão na localização do alvo.

5.1.3.3 Balística

A aplicação do NEKF em balística (KRAMER; STUBBERUD, 2005) visa a predição da trajetória e posição de um projétil ao longo do tempo, com isso calculando o instante e a posição de sua queda. A figura 5.6 mostra uma trajetória balística, a linha contínua representa a trajetória balística normal e a linha tracejada representa a trajetória modificada por *drags* (interferências). Existe uma série de fatores que incorporam-se ao modelo, fazendo com que a função $\hat{f}$ conhecida não seja a mesma do sistema real. Esses fatores podem ser a pressão do ar, ventos ou o choque com algum objeto.

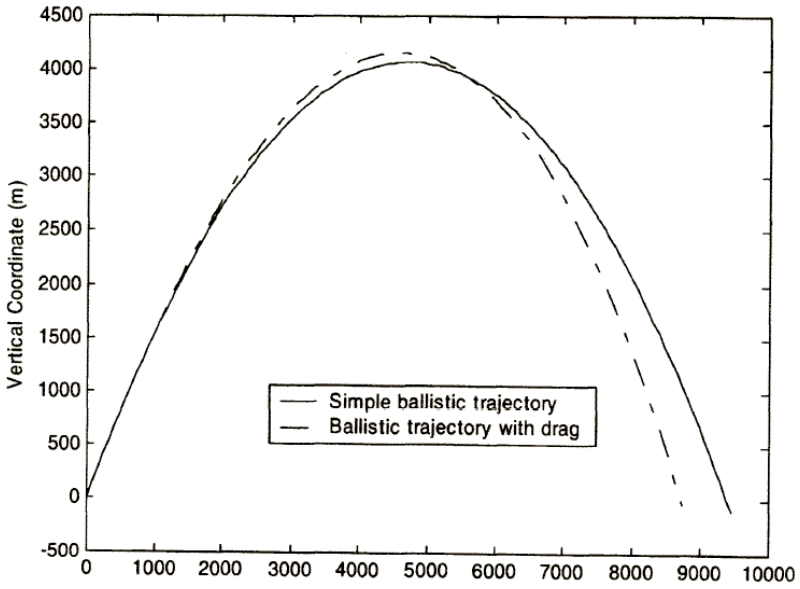

Figura 5.6: Trajetória balística, com e sem desvios

O modelo que forma a trajetória real do projétil é um composto do modelo da trajetória balística (função f do Filtro de Kalman) e de outro modelo adicional, desconhecido a priori (rede neural). Esse tipo de modelo possui uma função conhecida a priori que sofrerá variações, em que o NEKF é aplicado. O NEKF possui uma aplicabilidade muito maior que o FKE (por causa da rede neural), podendo ser aplicado também a modelos parcialmente conhecidos e não somente a modelos totalmente conhecidos.

Na predição da trajetória balística, inicialmente o NEKF possui apenas a função a priori (trajetória balística normal). Posteriormente, o NEKF vai utilizando a função melhorada pela RN, que vai aprendendo *on-line*. O NEKF utiliza a técnica de linearização pelas jacobianas, referida como linearização de "sinal pequeno" por (KRAMER; STUBBERUD, 2005). Os próprios autores comentam que, em dinâmicas altamente não-lineares, o erro pode crescer significativamente pelo desvio do estado do "ponto de linearização".

A figura 5.7 mostra os resultados da predição do local de queda do projétil. Como poderia-se imaginar, as predições realizadas no início da trajetória foram menos precisas que as realizadas mais na parte final. Porém, mesmo as predições iniciais tiveram erro pequeno, em torno de 15% do valor final, considerado pelos autores um bom grau de confiança.

5.1.4 Versão do NEKF com *Unscented Kalman Filter*

O artigo (ZHAN; WAN, 2006) apresenta a proposta de um método baseado no NEKF, mas com a substituição do FKE pelo *Uscented Kalman Filter* (UKF). O UKF é utilizado

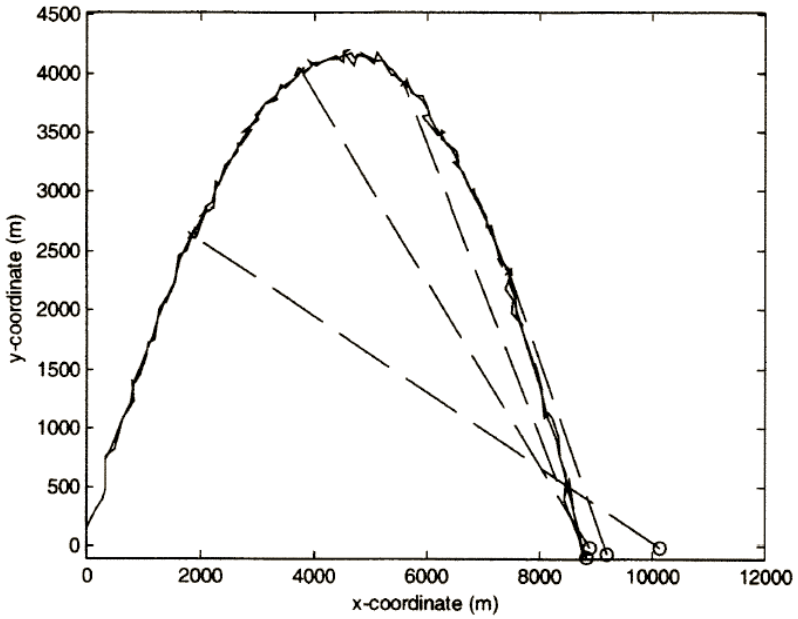

Figura 5.7: Estimativas de posição de queda do projétil, ao longo da trajetória

simultaneamente para predição dos estados e para treinamento da RN. As razões para a tentativa com o UKF é que o FKE, mesmo sendo simples e diretamente propagado, possui algumas desvantagens (no controle *on-line* de sistemas não-lineares): instabilidade na linearização; custo de cálculo das matrizes jacobianas; natureza parcial das estimativas. A principal vantagem do UKF é que é não é necessário nenhuma linearização para calcular a predição de estados e covariâncias. Por isso, sua covariância e Ganho de Kalman tendem a ser mais precisos, levando a melhores estimativas de estados.

A justificativa básica para a utilização do *Unscented Kalman Filter* é que é mais fácil aproximar uma distribuição gaussiana que aproximar uma função não-linear arbitrária. Em vez de fazer linearização utilizando matrizes jacobianas, o UKF usa uma abordagem amostral determinística para capturar as estimativas de média e variância com um conjunto mínimo de pontos de amostra (LAVIOLA, 2003). A transformada *unscented* é um método para calcular a estatística de variáveis aleatórias e utiliza uma transformação não-linear. Essa transformada usa um conjunto de pontos *sigma* que une propriedades fixas da distribuição anterior e permite a propagação direta da média e covariância através do sistema de equações não-lineares (GUANG-FU; XUE-YUAN, 2005), sem a necessidade de calcular a matriz jacobiana.

Para comparação entre o NEKF e a sua variação que Utiliza o UKF (chamado no artigo de NN-UKF), uma das funções utilizadas para aproximação foi:

$$y = \frac{2}{1+\exp\left(\frac{1}{1+\exp(0.1-0.5x)}+\frac{1}{1+\exp(0.5+0.4x)}-1\right)} + 0.5\sin(0.5x) + 0.5\frac{x}{1+x^2} \qquad (5.8)$$

A tabela 5.1 mostra a comparação de erro e variância de erro entre os métodos. O "NN-UKF" mostra-se superior ao NEKF nas comparações. Também é comentado no artigo (ZHAN; WAN, 2006) que a participação da RN é maior na predição do sistema do que o seu respectivo FKE ou UKF, particularmente na presença de incertezas.

Tabela 5.1: Comparação do NEKF com o NN-UKF

Algoritmos	MSE médio	Variância do MSE
FKE	0,3584	0,01295
UKF	0,2661	0,00925
NEKF	0,1380	0,00696
NN-UKF	0,0769	0,00176

5.1.5 Estimação Não-linear com *Unscented Kalman Filter* e Redes Neurais

Um modelo híbrido de rede neural com Filtro de Kalman para predição de séries temporais ruidosas é proposto por (WAN; MERVE, 2000). A RN serve como função de estimação de estados do *Unscented Kalman Filter*. A série temporal sem ruído é definida no artigo por:

$$x_k = f\left(x_{k-1}, \cdots, x_{k-M}, \mathbf{w}\right) + v_k \tag{5.9}$$

Onde o modelo f, parametrizado por $\mathbf{w}$, é aproximado pelo treinamento de uma rede neural com os dados limpos (sem ruído). O erro da RN (v_k) é considerado o ruído de processo. Adiciona-se ruído gaussiano branco na série original para gerar a série ruidosa $y_k = x_k + n_k$. O correspondente modelo de espaço de estados é dado por:

$$\begin{bmatrix} \mathbf{x}_k \\ x_k \\ x_{k-1} \\ \vdots \\ x_{k-M+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}\left(\mathbf{x}_{k-1}, \mathbf{w}\right) \\ F\left(x_{k-1}, \cdots, x_{k-M}, \mathbf{w}\right) \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_{k-1} \\ \vdots \\ x_{k-M+1} \end{bmatrix} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \cdot v_{k-1} \\ \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot v_{k-1} \end{bmatrix} \tag{5.10}$$

$$y_k = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} . \mathbf{x}_k + n_k \tag{5.11}$$

O trabalho apresenta a predição da série caótica de Mackey-Glass ruidosa com parâmetro de ciclo 30, mostrada na figura 5.8. Compara-se os algoritmos FKE e UKF, ambos com a rede neural como função de transição de estados. Essa compraração é mostrada na figura 5.8. O UKF apresenta resultados bem melhores que o FKE para esse experimento.

O trabalho apresenta a colocação da RN como função "pura" do FKE e do UKF, sendo treinada com dados não-ruidosos (série ideal). Para aplicações reais, a série não ruidosa não está disponível, pois é o que se pretende prever. O principal enfoque do trabalho (WAN; MERVE, 2000) e do trabalho anterior (WAN; MERVE; NELSON, 2000) é a comparação de treinamento de redes neurais com UKF e FKE.

5.2 Ajuste de Parâmetros do Filtro de Kalman com Redes Neurais

O desempenho ótimo do Filtro de Kalman só se dá com o ajuste ótimo de parâmetros. O ajuste é considerado o processo de obtenção de melhores valores dos parâmetros, como as matrizes de covariância de ruído **Q** e **R**, dando melhor desempenho ao filtro no sentido de diminuir o erro. Tradicionalmente o ajuste de parâmetros é feito por intuição técnica ou tentativa e erro, o que não garante o melhor desempenho para o filtro, devido ao grande número de parâmetros a ser estimado (KORNIYENKO; SHARAWI; ALOI, 2005).

O primeiro dos exemplos de ajuste de parâmetros de um FK por uma RN é descrito em (FISHER; RAUCH, 1994). O artigo mostra um FKE com uma RN para estimar os seus parâmetros e condições iniciais. Tem-se várias situações de modelos de sistemas

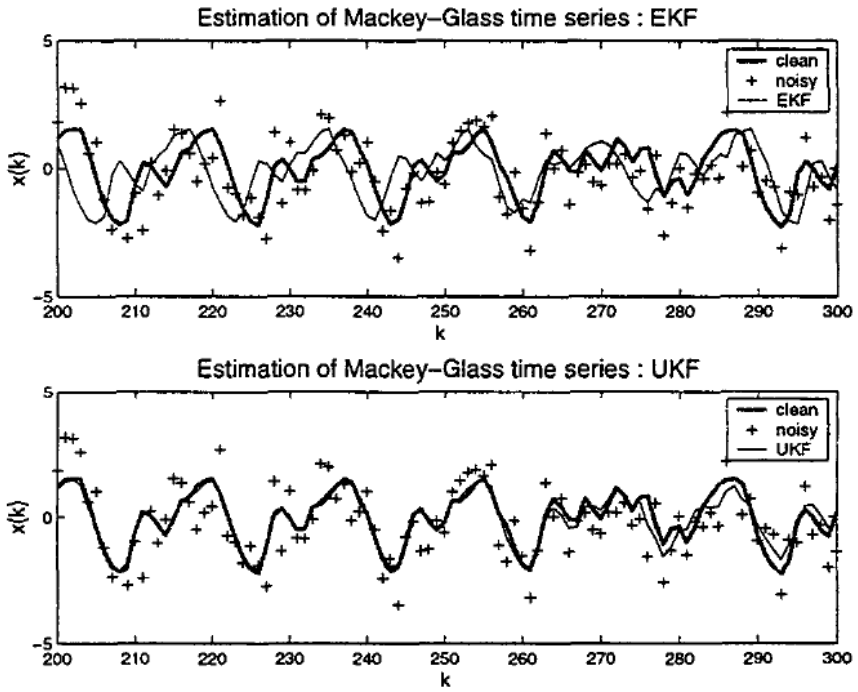

Figura 5.8: Estimação da série de Mackey-Glass com a RN como função do FKE e do UKF

em que várias abordagens são utilizadas. Quando o modelo é conhecido, pode-se utilizar um FK linear. Quando existem pequenas não-linearidades o FKE funciona satisfatoriamente. Porém, quando existem grandes linearidades com parâmetros e condições iniciais desconhecidas, é necessária a utilização de múltiplos FKE. Cada FKE possui diferentes condições iniciais e parâmetros. Uma RN pode cumprir esse papel de múltiplos FKE.

A RN utilizada no artigo foi a Rede Neural de Regressão Geral (GRNN). As saídas do FKE são passadas para um estado posterior e então comparadas com os dados pré-calculados pela GRNN. Essa base de dados pré-calculada consiste dos dados de várias execuções *off-line* do FKE com grande variedade de condições iniciais e parâmetros. A rede GRNN encontra os melhores mapeamentos da saída do FKE para os devidos valores das condições iniciais. Os valores corrigidos são utilizados para atualizar as condições e parâmetros do filtro. O exemplo apresentado no artigo relata a detecção de trajetória por um interceptador de mísseis. Mesmo que o alvo saia fora de alcance (não fique visível) continua-se o rastreamento, com as informações existentes enquanto o alvo estava visível.

O artigo de (KORNIYENKO; SHARAWI; ALOI, 2005) trata do uso de RNs para estimar a melhor configuração de parâmetros para um Filtro de Kalman, onde as RNs utilizam bases de dados de execuções do filtro. O artigo mostra critérios de otimização global para a escolha de quais dados são passados para a RN. Forma-se uma base de dados com as combinações de parâmetros e seus respectivos valores de otimização (ou minimização, no caso do MSE). Escolhe-se valores de parâmetros dentro das faixas aceitáveis de cada parâmetro.

Nesse artigo, comparou-se uma rede GRNN com uma rede RBNN (*Regular Radial Basis Neural Network* ou RN de base radial). A rede RBNN apresentou melhores resultados que a GRNN. Como o experimento visava a estimação de 2 parâmetros (**Q** e **R**), escolheu-se 9 experimentos rotulados para a RN interpolar. A RN constrói a superfície de decisão dos melhores valores de parâmetros, como mostrado na figura 5.9. No exemplo,

a rede RBNN estima a otimização dos parâmetros (quadradinho, na figura) de maneira muito próxima aos verdadeiros parâmetros ótimos (triângulo).

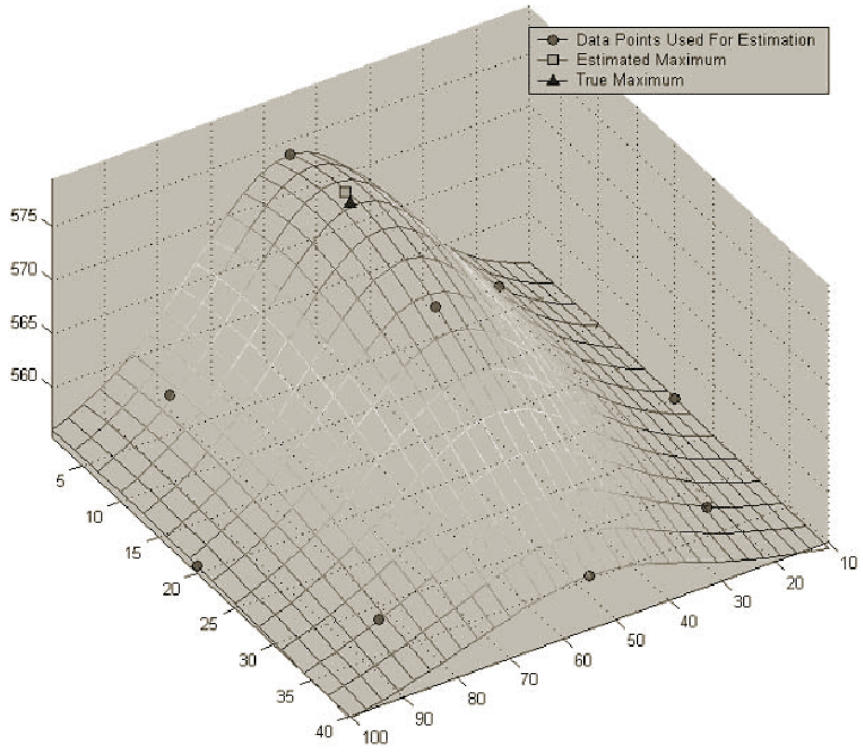

Figura 5.9: Superfície de decisão da otimização dos parâmetros com rede RBNN

5.3 Treinamento de Redes Neurais com Filtro de Kalman Estendido e suas Variantes

O funcionamento do FK é voltado a estimar o estado de um sistema que pode ser modelado como um sistema linear com ruído gaussiano branco e onde as medidas disponíveis são combinações lineares dos estados do sistema corrompidas pelo ruído. Para o treinamento neural, os pesos da RN são os estados do FK a serem estimados e a saídas desejadas da rede são as medidas utilizadas pelo FK, conforme as equações abaixo (SHUHUI, 2001):

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{q}(n) \tag{5.12}$$

$$\mathbf{d}(n) = \mathbf{h}(\mathbf{w}(n), \mathbf{x}(n)) + \mathbf{r}(n) \tag{5.13}$$

Onde, em um instante n:

- $\mathbf{w}(n)$ é um vetor com todos os pesos da rede;
- $\mathbf{x}(n)$ é o vetor de entrada da rede, do conjunto de treinamento;
- $\mathbf{d}(n)$ é o correspondente vetor de saída desejado para x(n);
- $\mathbf{h}(n)$ define um relacionamento não-linear entre as entradas, saídas e pesos da RN;
- $\mathbf{q}(n)$ é o ruído de processo no modelo do sistema;

- $\mathbf{r}(n)$ é o ruído de medida.

O artigo de (TAKENGA et al., 2004) trata da comparação do Gradiente Descendente (GD), do FKE e do FKE Desacoplado (FKED) para treinamento de Redes Neurais. A RN utilizada é a Rede de Função de Base Radial (RBF) e a aplicação descrita é a detecção de posição, baseada em sinais digitais. Sabe-se que os algoritmos de treinamento têm papel decisivo no desempenho das RNs. Os algoritmos mais utilizados são aqueles baseados em gradiente, porém descobriu-se que métodos baseados em Filtro de Kalman também podem ser utilizados.

Para sistemas lineares dinâmicos com ruído branco, um FK é considerado um estimador ótimo. Para sistemas não-lineares com ruído colorido, o FK pode ser estendido por linearização do sistema em torno das estimativas dos parâmetros atuais. Embora com custo computacional bem menor que o gradiente descendente, o FKE também é custoso, tornando ainda proibitivo o seu uso em grandes redes neurais. Com isso, encontraram-se variantes para diminuir o seu custo. O FKE Disjunto é uma forma derivada do FKE em que se assume que os pesos entre muitas estimativas podem ser ignorados, necessitando de uma menor quantidade de operações por iteração.

No experimento mostrado, a posição do sistema móvel é automaticamente encontrada, conhecendo-se o tamanho do sinal de um ponto. Os sinais são gerados a partir da segmentação de uma distância de 450 metros em 15 segmentos com 10 pontos cada. O treinamento é feito com a emissão de sinais em todos esses pontos. Os experimentos mostram que os treinamentos baseados em FK (FKE e FKED) apresentaram menor taxa média de erro, de 40 metros. O GD apresentou erro médio de 65 metros. Essa diferença de precisão diminui se o número de neurônios na camada oculta for aumentado.

Os métodos FKE e FKED possuem a mesma precisão, pois os dois são baseados no Filtro de Kalman. O FKED é preferível porque consome menos tempo de treinamento. No exemplo do artigo, o FKE consome 12 minutos e o FKED, 8 minutos. Essa diferença se deve às iterações mais rápidas do FKED. O número de iterações é praticamente o mesmo, como podemos ver na figura 5.10. Na figura, EKF significa FKE e DKF, FKED. Também podemos perceber na figura 5.10 que os métodos baseados na filtragem de Kalman convergem em menos iterações, se comparados ao GD.

O artigo de (SHUHUI, 2001) compara o Backpropagation (BP) com o FKE, testando os algoritmos em sua forma tradicional e na forma de lote. Para uma rede multi-camadas não-linear, o usual FK pode ser usado apenas se o sistema é anteriormente linearizado, assim como faz o FKE. Normalmente no treinamento utilizando FKE, a atualização é feita instância por instância. Outra forma é fazer o treinamento em lote. Nesse caso, os dados são apresentados um a um, mas com uma única atualização dos pesos no final do lote. O exemplo apresentado é o uso de RN para estimar o poder de giro de turbinas, comparando-se as abordagens de BP e FKE, com e sem processamento em lote. A estimação do poder das turbinas serve para maximizar o uso da eletricidade. Esse poder é influenciado por muitos fatores como velocidade do giro, direção do giro, terreno, densidade do ar, estrutura da turbina, clima e estação de um ano. Um conjunto de dados de 2048 padrões foi usado para treinar a RN, com as técnicas de FKE e BP, até um critério de parada.

Na figura 5.11, DEFK refere-se ao FKE Disjunto. A técnica chamada *multi-stream* é uma forma em lote (*batch*) desse algoritmo. A figura 5.11 mostra os erros da rede do BP e FKE em forma de lote. Cada passo na figura significa a atualização de um lote de 32 padrões, mas o erro é calculado para todos os exemplos de treinamento. Podemos perceber, na figura 5.11, que o algoritmo de Kalman pode encontrar menor erro que o BP. O erro do FKE decresce rapidamente com a apresentação dos dados de treinamento,

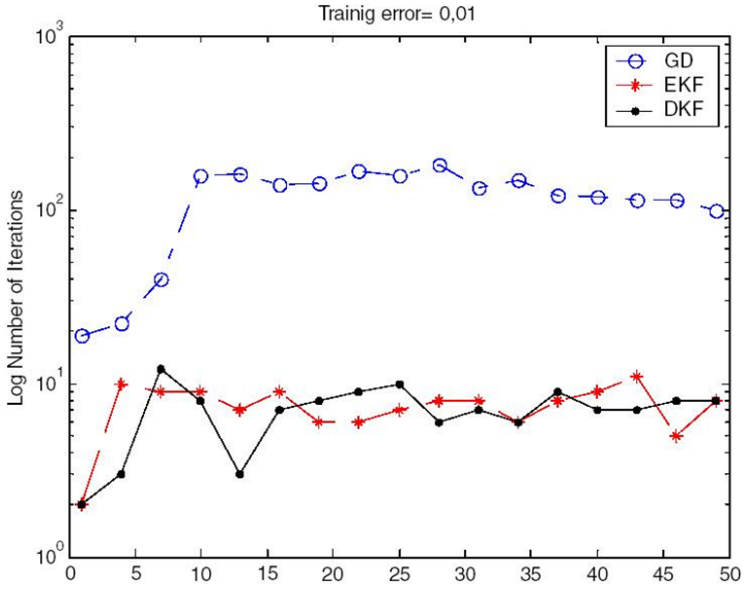

Figura 5.10: Número de iterações necessárias para convergência em cada um dos métodos de treinamento

refletindo seu poder de aprendizado e convergência.

Realizaram-se também simulações para comparar a forma tradicional do FKE com a forma em lote. Descobriu-se que as formas em lote exibem melhores propriedades de convergência e também possuem processo de treinamento mais estável que a forma tradicional. As comparações mostram que o FKE tem maior capacidade de aprendizado, melhor propriedade de convergência e maior velocidade de treinamento que o BP. Percebe-se também que o treinamento em lote mostra maior convergência e processo de treinamento mais estável que o treinamento padrão.

O artigo de (GANG; YU, 2005) apresenta o *Node Decoupled Extend Kalman Filter* (NDEFK ou FKE Disjunto, com os pesos acoplados por nós) para treinar uma RN híbrida auto-regressiva. A RN é utilizada para identificação de categorias de motores. A principal diferença do FKE disjunto para o FKE padrão é a linearização da equação de espaço de estados. A função de transição é transformada em uma matriz de derivadas, onde cada posição pode ser obtida pela regra da cadeia. O artigo mostra um experimento comparando o aprendizado do NDEKF com o do BP, com o NDEKF convergindo em poucas iterações. A comparação do NDEKF com o BP indica que o NDEKF converge mais rapidamente, está menos suscetível a mínimos locais e tem melhor capacidade de generalização.

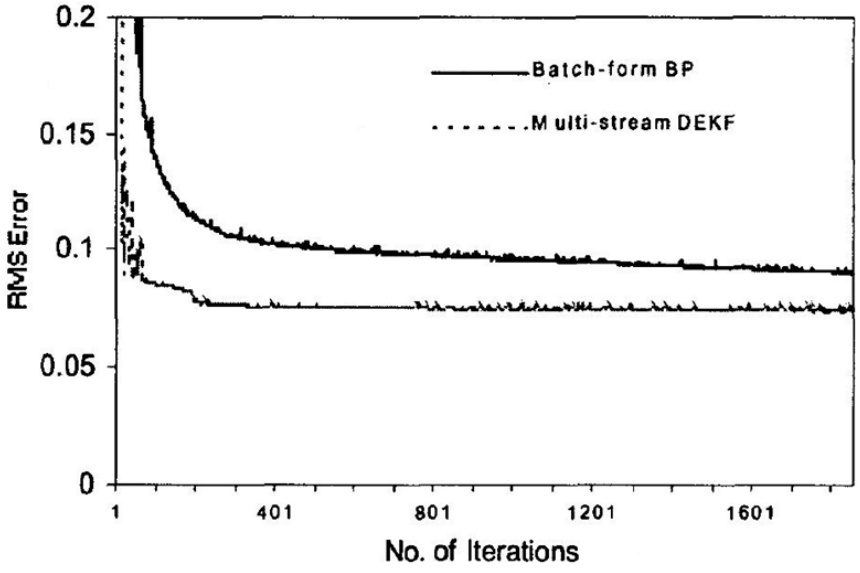

Figura 5.11: Comparação de taxa de erro do BP e FKE em forma de lote

6 PROPOSTA DO MÉTODO NEURO ESTATÍSTICO

Este capítulo apresenta a proposta de um método neuro-estatístico, unindo as características de uma rede neural de múltiplas camadas com o Filtro de Kalman Estendido. Apresenta-se a motivação e a justificativa para essa proposta; os modelos e formalismos utilizados, mostrando a relação entre a RN e o FKE e a explicação de todo o algoritmo de funcionamento do método.

6.1 Motivação

Esta proposta trata da construção de um método híbrido de uma rede neural de múltiplas camadas com o método estatístico Filtro de Kalman Estendido para aplicações de predição de séries temporais. A justificativa para a junção dessas abordagens é o fato de possuírem características complementares, no que se refere à regressão (previsão) em séries com presença de ruído e que seguem dinâmicas desconhecidas e não-lineares. A seguir são comentados os motivos pelos quais a hibridização de redes neurais com o método estatístico é desejável.

O método estatístico Filtro de Kalman (KALMAN, 1960), consegue minimizar a influência do ruído, trabalhando com a variância do ruído nos dados extraídos do sistema real (ruidoso). Essa variância é utilizada para melhorar a predição, juntamente com a covariância do erro de predição. O motivo da utilização do Filtro de Kalman Estendido (em vez do FKD) é o tratamento de não-linearidades no modelo gerador da série e a possibilidade de interagir com a RN na geração das jacobianas. As não-linearidades tratadas pelo FKE são apenas de primeira ordem (suaves) e as matrizes jacobianas são as responsáveis pelo tratamento dessas não-linearidades. As jacobianas podem ser calculadas diretamente a partir de valores internos de uma rede neural de múltiplas camadas alimentada adiante. Essa característica do FKE torna-o muito indicado para uso juntamente com RNs. Com a interação da RN com o FKE, no modelo proposto, visa-se resolver uma das limitações do FK, abordada por (DECRUYENAERE; HAFEZ, 1992): o tratamento de não-linearidades. A outra limitação apontada, a suposição de o ruído obedecer distribuição gaussiana, não é abordada neste trabalho.

Os métodos estatísticos necessitam conhecer o modelo estatístico gerador (função) das séries. O principal problema dos métodos estatísticos como o FKE é a dificuldade de se criar uma abordagem complexa pela falta de compreensão de certos modelos reais, onde muitas características e parâmetros não são conhecidos. Daí advém a necessidade de testes de muitas hipóteses e combinações através de massivos processos estatísticos, o que em muitas vezes não é viável (MORETTIN; TOLOI, 2004). O Filtro de Kalman necessita conhecer uma função $\mathbf{f}$ que descreva o modelo gerador do sistema. Como na predição de séries temporais o objetivo é exatamente a aproximação do modelo gerador

desconhecido das séries, o FK não pode ser utilizado isoladamente para PST.

As redes neurais executam as previsões sem a necessidade de conhecimento das funções complexas dos sistemas. As RNs aprendem a partir de amostras dos próprios dados, fazendo ajustes de maneira gradual, aproximando a função do sistema. Com isso, as RNs não necessitam conhecer previamente o modelo estatístico gerador das séries. As RNs apresentam uma não-linearidade de um tipo especial: presente em cada um de seus neurônios. A combinação de não-linearidades de vários neurônios de camadas ocultas ou de sucessivas camadas torna as RNs muito poderosas, proporcionando o tratamento de altos graus de não-linearidades nas séries (HAYKIN, 2001a).

A dificuldade existente nas RNs encontra-se no fato de que elas, como não modelam ruído, possam confundir o sinal (função original do sistema) com o ruído. Experimentos deste trabalho comprovam a dificuldade que as RNs apresentam na predição, à medida que insere-se ruído nos dados. A intenção deste trabalho é unir a capacidade de modelagem de ruído (presente no FKE), com adaptação a modelos desconhecidos e tratamento de não-linearidades (presentes nas redes neurais). O principal objetivo do método proposto é apresentar melhores resultados que os métodos puramente neurais e melhor aplicabilidade que os métodos puramente estatísticos, na predição de séries temporais. As séries tratadas são ruidosas (não possuindo dados livres de ruído para treinamento), não possuem modelo gerador conhecido, e apresentam grandes não-linearidades.

6.2 Modelos de Entrada-Saída Utilizados

Os modelos de entrada-saída, utilizados para a RN do modelo neuro-estatístico, são baseados nos modelos NARX e NOE, mostrados na seção 2.4.2. O modelo baseado no NARX não apresenta entradas exógenas e pode ser chamado de NAR (*Nonlinear Auto-regressive* - Modelo Auto-regressivo Não-linear). O modelo NAR estabelece uma relação entre as saídas passadas e a saída prevista na seguinte forma:

$$\hat{y}(n+1) = F(y(n), \cdots, y(n-T+1)) \tag{6.1}$$

Onde $y(n), \cdots, y(n-T+1)$ são os valores anteriores de saída, medidos diretamente do sistema e $\hat{y}(n+1)$ é o valor estimado da próxima saída, calculado a partir dessas saídas atrasadas. O modelo não apresenta entradas exógenas pois dispõe apenas do histórico da série temporal como entrada. Um modelo NAR de ordem $T = 2$ é mostrado na figura 6.1.

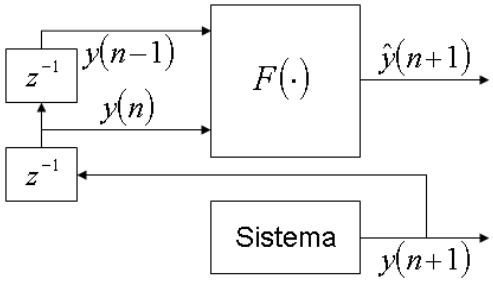

Figura 6.1: Modelo NAR

Utiliza-se também um modelo correspondente ao modelo NOE, sem as entradas exógenas. Esse modelo possui a relação de entrada-saída:

$$\hat{y}(n+1) = F(\hat{y}(n), \cdots, \hat{y}(n-T+1)) \tag{6.2}$$

Onde $\hat{y}(n), \cdots, \hat{y}(n-T+1)$ são estimativas passadas de saída. A diferença desse modelo para o NAR é a realimentação com as próprias saídas previstas. Em ambos modelos não há a presença de entradas exógenas, para tratar séries em que não dispõe-se de ação ou outra entrada, apenas os valores anteriores da mesma. A figura 6.2 mostra um modelo NOE sem entradas exógenas de ordem 2.

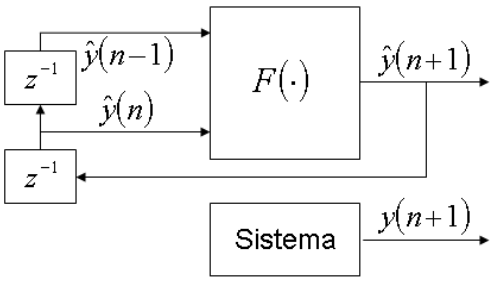

Figura 6.2: Modelo NOE sem entradas exógenas

6.3 Explicação do Modelo Proposto Baseado no Modelo do Filtro de Kalman

O modelo proposto envolve o uso de uma rede neural como processo do FKE, fazendo a tarefa de predição, substituindo a função **f**. O emprego da RN elimina a necessidade de conhecimento prévio da função de transição de estados. O restante do método funciona como sendo um FKE, trabalhando com as covariâncias dos ruídos e erros de predição, para melhorar a qualidade da solução do método.

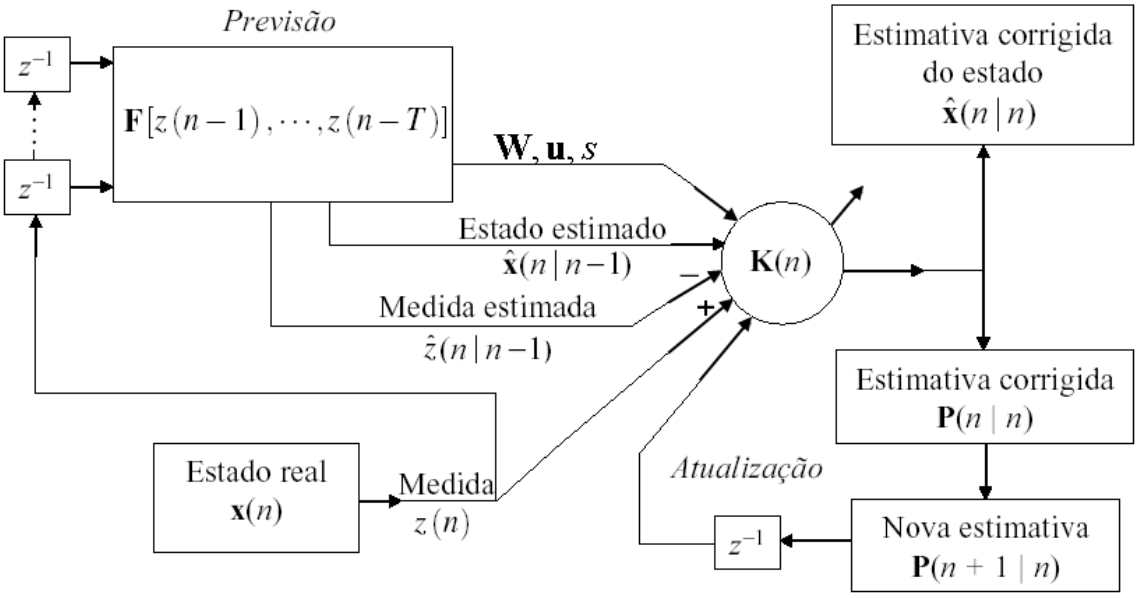

Figura 6.3: Modelo neuro-estatístico sem realimentação da saída

São criadas duas variações do modelo proposto, baseadas nos modelos de entrada-saída da rede neural, mostrados na seção anterior. Com o NAR, as saídas do método neuro-estatístico não realimentam a entrada da rede, como mostrado na figura 6.3. Nesse

caso, a RN recebe como entrada um vetor com T medidas anteriores, na forma:

$$\mathbf{F}\left[z\left(n-1\right),\cdots,z\left(n-T\right)\right] \tag{6.3}$$

Onde $z\left(n-1\right),\cdots,z\left(n-T\right)$ são as T medidas anteriores e $\mathbf{F}\left[\cdot\right]$ representa a função formada pela RN, onde a nova posição do estado é prevista. A medida representa a posição atual da série (ruidosa). A rede prevê a primeira posição do vetor de estados. As demais posições são atrasadas em uma posição no tempo, formando um novo vetor $\mathbf{x}$.

Com o modelo de entrada-saída NOE, as saídas do método neuro-estatístico realimentam a entrada da rede, como mostrado na figura 6.4. Nesse caso, a RN recebe como entrada as T posições do vetor de estados estimados, na forma:

$$\mathbf{F}\left[\hat{x}_1,\hat{x}_2,\cdots,\hat{x}_T\right] \tag{6.4}$$

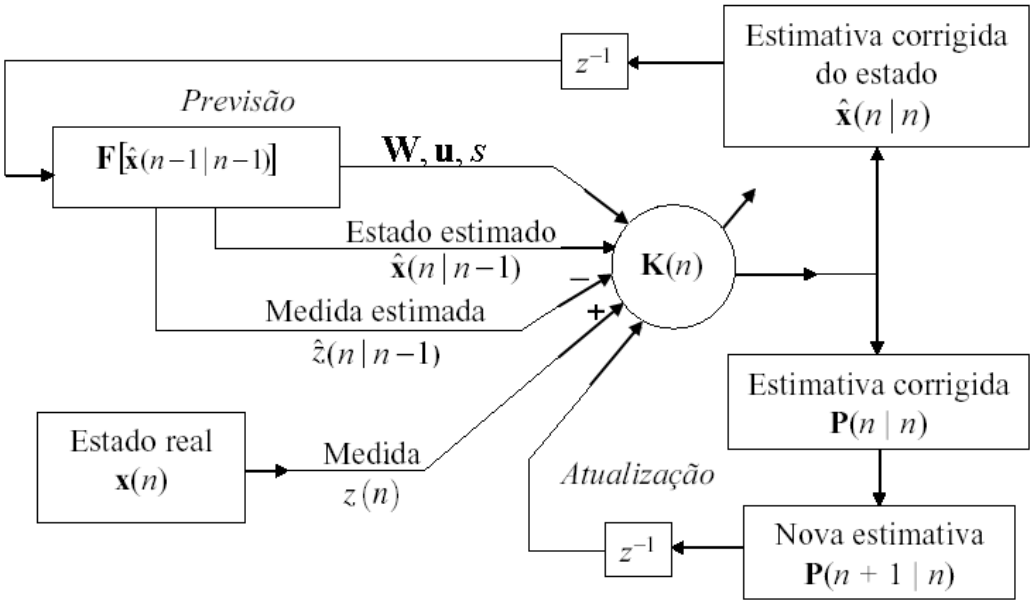

Figura 6.4: Modelo neuro-estatístico com realimentação da saída

No sub-modelo apresentado na figura 6.4, a rede neural recebe como entrada as estimativas anteriores do filtro, representando os valores da série filtrados ou suavizados. Com a entrada de valores suavizados, espera-se que a RN melhore a sua saída (predição). A diferença entre os dois sub-modelos é que sem a realimentação para a entrada, a predição a *priori* será a mesma de uma RN sozinha. Nesse caso o modelo melhorará apenas a tarefa da filtragem, em relação a RN pura. Os resultados de predição podem ser melhorados se a rede neural do método for treinada novamente com os valores filtrados.

As figuras também mostram a RN passando os pesos de todas as camadas ($\mathbf{W}$), os valores de saída da camada oculta ($\mathbf{u}$) e o valor da camada de saída (s) (além das entradas da rede) para serem utilizados na fase de atualização. Esses dados vindos da RN são utilizados para a computação das matrizes jacobianas, responsáveis pelo tratamento de não-linearidades do FKE. As jacobianas são utilizadas para calcular as matrizes de covariâncias dos erros, utilizadas para computar o Ganho de Kalman (módulo K nas figuras 6.3 e 6.4). A utilização da RN como processo torna possível aproximar uma função não-linear desconhecida $\mathbf{f}$. O cálculo das matrizes de derivadas parciais (jacobianas), a partir das informações de cada camada da RN, possibilita que essas matrizes reflitam o

mapeamento de entrada-saída da rede (com suas possíveis não-linearidades). Com isso, a RN e o FKE fazem um tratamento conjunto de não-linearidades.

A figura 6.5 mostra a estrutura da rede neural utilizada. A rede recebe como entrada os T últimos valores da série temporal, que são também as T posições do vetor de estados $\mathbf{x}$. A RN possui uma única saída para prever o valor de x_1, que é a posição atual da série.

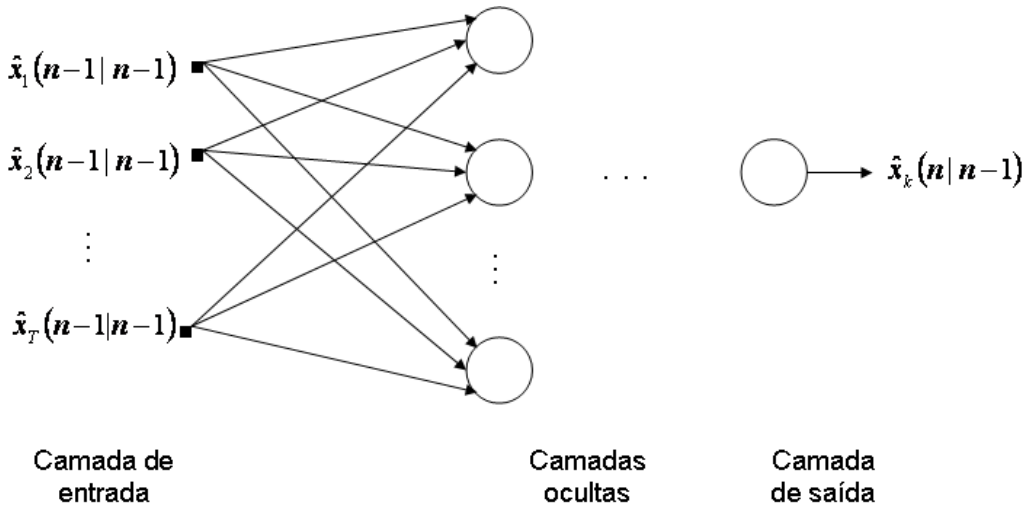

Figura 6.5: Estrutura da rede neural

A estimativa corrigida do estado representa a posição atual da série filtrada (última posição do vetor de estados), mas o modelo realiza também (automaticamente) a predição do estado seguinte, antes da medida. O valor previsto é a própria saída da função de transição de estados (rede neural). Então o mesmo modelo serve tanto para predição como para filtragem, só mudando o local de onde o estado é observado (a *priori* ou a *posteriori*).

6.4 Formalismo do Método Proposto

A explicação matemática do modelo neuro-estatístico será baseada na formulação do algoritmo do Filtro de Kalman, mostrada na seção 3.2, utilizando o Modelo de Espaço de Estados (MEE) e o modelo de entrada-saída NOE sem entradas exógenas (realimentação da entrada), explicado nas seções 6.2 e 6.3. A predição da série se dará pela estimação do vetor de estados $\mathbf{x}$ assim formado:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_T \end{bmatrix} \tag{6.5}$$

A inicialização do método é feita da seguinte forma:

- Treina-se a rede neural *off-line*, usando trechos da série temporal;
- Estima-se a variância do ruído de processo Q com medidas estatísticas do erro do processo (rede neural);
- Estima-se a variância do ruído de medida R através da aplicação de um filtro *off-line* em medidas ruidosas;

- Inicializa-se a estimativa de estado $\hat{\mathbf{x}}$ do instante anterior com medidas anteriores (ruidosas) da série temporal:

$$\hat{\mathbf{x}}(n-1|n-1) = \begin{bmatrix} z(n-1) \\ z(n-2) \\ \vdots \\ z(n-T) \end{bmatrix} \tag{6.6}$$

- A matriz de covariância do erro $\mathbf{P}$ do instante anterior é inicializada com uma matriz quadrada de zeros, com número de linhas e colunas igual à quantia de termos de $\hat{\mathbf{x}}$:

$$\mathbf{P}(n-1|n-1) = (\hat{\mathbf{x}} \cdot 0)(\hat{\mathbf{x}} \cdot 0)^T \tag{6.7}$$

6.4.1 Fase de Predição do Estado

O estado é projetado adiante (previsto) pela rede neural, que funciona como função de transição de estados. Para o modelo com realimentação da entrada, a atualização do estado será dada por:

$$\hat{\mathbf{x}}(n|n-1) = \mathbf{F}[\hat{\mathbf{x}}(n-1|n-1)] \tag{6.8}$$

Onde $\hat{\mathbf{x}}(n|n-1)$ é a estimativa do vetor de estado para o tempo atual (n), realizada no instante anterior ($n-1$). $\mathbf{F}$ é a função de transição de estados, com a rede neural. Para o modelo sem realimentação da entrada, a função $\mathbf{F}$ receberá sempre as medidas anteriores:

$$\hat{\mathbf{x}}(n|n-1) = \mathbf{F}[z(n-1), z(n-2), \cdots, z(n-T)] \tag{6.9}$$

Como deseja-se apenas calcular a posição atual da série (x_1), a RN fará a predição dessa posição e as demais serão apenas deslocadas. Para o modelo com realimentação da entrada, tem-se:

$$\begin{aligned}
&\hat{x}_1(n|n-1) = RN[\hat{x}_1(n-1|n-1), \hat{x}_2(n-1|n-1), \cdots, \hat{x}_T(n-1|n-1)] \\
&\hat{x}_2(n|n-1) = \hat{x}_1(n-1|n-1) \\
&\hat{x}_3(n|n-1) = \hat{x}_2(n-1|n-1) \\
&\vdots \\
&\hat{x}_T(n|n-1) = \hat{x}_{T-1}(n-1|n-1)
\end{aligned} \tag{6.10}$$

Como no modelo sem realimentação da entrada a RN recebe as medidas anteriores, cada posição do vetor de estimativa de estados é calculado da seguinte forma:

$$\begin{aligned}
&\hat{x}_1(n|n-1) = RN[z(n-1), z(n-2), \cdots, z(n-T)] \\
&\hat{x}_2(n|n-1) = \hat{x}_1(n-1|n-1) \\
&\hat{x}_3(n|n-1) = \hat{x}_2(n-1|n-1) \\
&\vdots \\
&\hat{x}_T(n|n-1) = \hat{x}_{T-1}(n-1|n-1)
\end{aligned} \tag{6.11}$$

Uma rede neural propragada adiante de uma camada oculta, utilizada para prever a posição atual da série, é mostrada na figura 6.6. Essa RN recebe as entradas, no modelo com realimentação da entrada (NOE). A RN equivalente para o modelo NAR é mostrada na figura 6.7.

A partir da propagação do estado adiante, as demais equações seguem a forma do FKE (para ambos modelos de entrada-saída), sendo que algumas dessas equações são

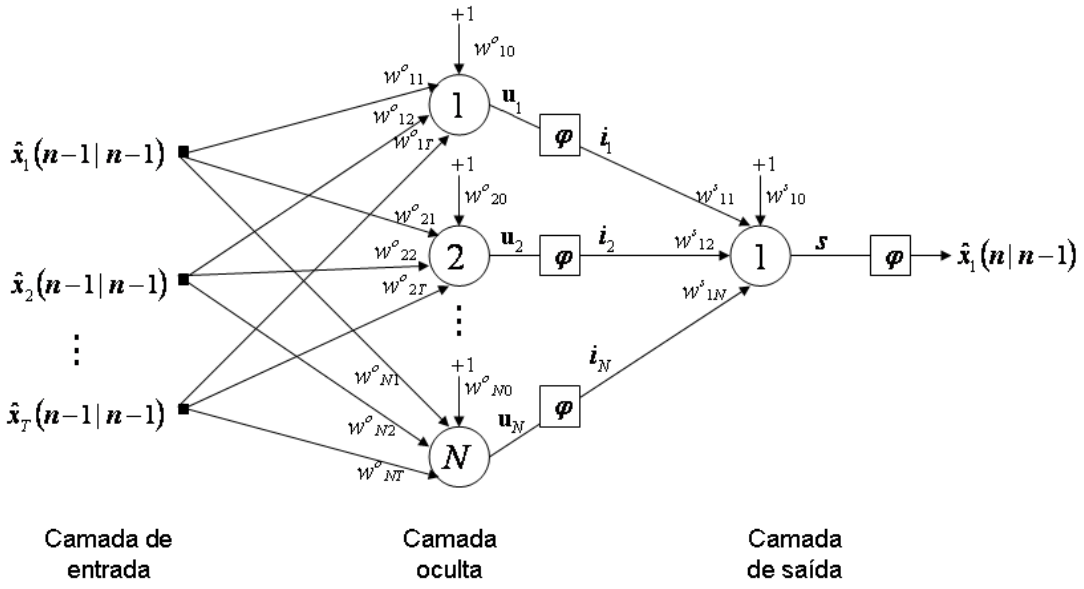

Figura 6.6: Rede neural para previsão da primeira posição do vetor de estados, no modelo NOE

simplificadas para a aplicação de predição de séries temporais. A estimativa da medida é descrita pela função:

$$\hat{z}(n|n-1) = h[\hat{\mathbf{x}}(n|n-1), 0] \tag{6.12}$$

Como o que está se buscando prever é a posição atual da série (x_1) e as medidas são os próprios valores da série ruidosos, o valor da função de estimativa da medida é a primeira posição do vetor de estados:

$$\hat{z}(n|n-1) = \hat{x}_1(n|n-1) \tag{6.13}$$

A matriz de covariância do erro de estimação $\mathbf{P}$ também é projetada adiante (atualizada para o novo instante de tempo), ficando da mesma forma que o FKE:

$$\mathbf{P}(n|n-1) = \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \mathbf{P}(n-1|n-1) \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right)^T + \frac{\partial \mathbf{F}}{\partial \mathbf{w}} \mathbf{Q}(n-1) \left(\frac{\partial \mathbf{F}}{\partial \mathbf{w}}\right)^T \tag{6.14}$$

Onde:

- $\mathbf{P}(n-1|n-1)$ é a matriz de covariância do erro calculada no tempo $n-1$;
- $\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$ e $\frac{\partial \mathbf{F}}{\partial \mathbf{w}}$ representam as jacobianas da função $\mathbf{F}$ em relação a $\mathbf{x}$ e a $\mathbf{w}$, respectivamente;
- $\mathbf{Q}(n-1)$ é a variância do ruído de processo. $\mathbf{Q}(n-1)$ representa a variância no instante anterior, mas pode ser mantida com valor fixo em todo o processo ou atualizada de tempos em tempos. A variância Q será relativa ao ruído do mecanismo de estimativa, ou seja, a imprecisão da rede neural em estimar o estado. O valor da variância R normalmente é mantido fixo, pois representa o nível de ruído nos dados da série, devido à imprecisões na obtenção dos dados. Normalmente esse ruído é fixo na grande maioria das séries temporais.

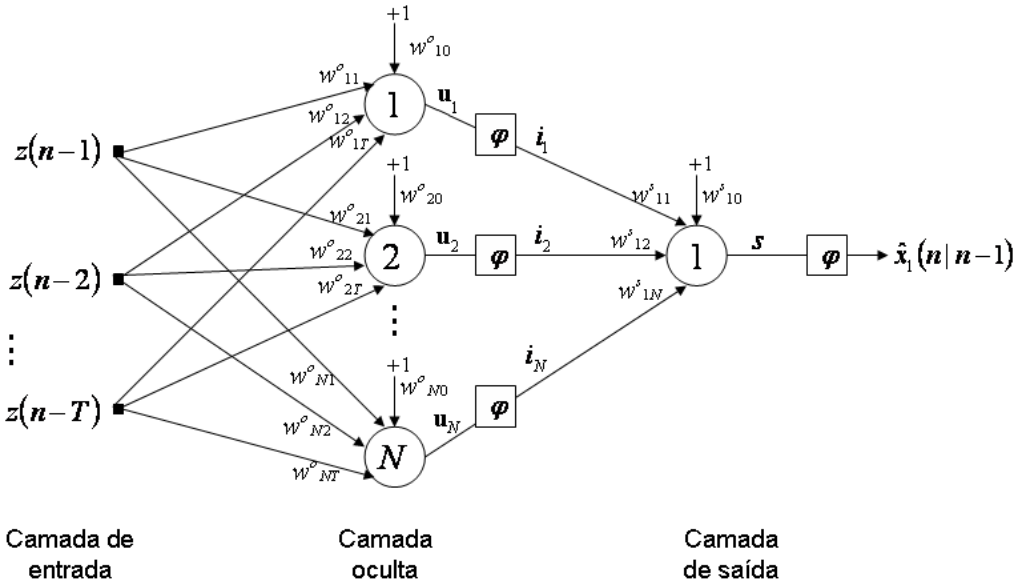

Figura 6.7: Rede neural para previsão da primeira posição do vetor de estados, no modelo NAR

6.4.2 Fase de Atualização do Estado

A matriz de variância do erro de estimação da medida S_{zz} possui dimensão $[1 \times 1]$ pois o erro de medida é um escalar.

$$S_{zz}(n|n-1) = \frac{\partial h}{\partial \mathbf{x}} \mathbf{P}(n|n-1) \left(\frac{\partial h}{\partial \mathbf{x}}\right)^T + \frac{\partial h}{\partial v} R(n) \left(\frac{\partial h}{\partial v}\right)^T \tag{6.15}$$

Onde:

- $\frac{\partial h}{\partial \mathbf{x}}$ é a jacobiana da função de medida h em relação ao estado,
- $\frac{\partial h}{\partial v}$ é a jacobiana da função h em relação do ruído de medida.

A matriz de covariância do erro da estimativa do estado pelo erro de estimativa da medida $\mathbf{S_{xz}}$ possui dimensão $[T \times 1]$, onde T é o número de termos do estado ($\hat{\mathbf{x}}$).

$$\mathbf{S_{xz}}(n|n-1) = \mathbf{P}(n|n-1) \left(\frac{\partial h}{\partial \mathbf{x}}\right)^T \tag{6.16}$$

O Ganho de Kalman ($\mathbf{K}$) também possuirá dimensão $[T \times 1]$:

$$\mathbf{K}(n) = \mathbf{S_{xz}}(n|n-1) {S_{zz}}^{-1}(n|n-1) \tag{6.17}$$

A atualização da estimativa de estado também será da mesma forma que o FK, sendo calculada a partir do Ganho de Kalman e da inovação:

$$\hat{\mathbf{x}}(n|n) = \hat{\mathbf{x}}(n|n-1) + \mathbf{K}(n)(z(n) - \hat{z}(n|n-1)) \tag{6.18}$$

E, por fim, a atualização da matriz de covariância do erro de predição do estado:

$$\mathbf{P}(n|n) = \mathbf{P}(n|n-1) - \mathbf{K}(n) S_{zz}(n|n-1) \mathbf{K}^T(n) \tag{6.19}$$

6.4.3 Matrizes Jacobianas

Nesta seção é mostrado o processo de obtenção das matrizes jacobianas pelo método neuro-estatístico. Essas jacobianas são utilizadas posteriormente no cálculo das matrizes de covariância do erro das estimativas. Como a função de estimativa do estado seguinte **F** é baseada na própria rede neural, a jacobiana da saída desta linearizará o estado da RN em cada predição. O cálculo das derivadas parciais é feito utilizando todas as camadas da RN, aproveitando toda a capacidade da rede de prever não-lineridades. A jacobiana da função **F** em relação ao estado é baseada na rede neural. Então essa jacobiana não será simplesmente calculada sobre uma função estimada, como no caso do FKE. Neste caso, a função é aproximada pela RN e, consequentemente, a jacobiana refletirá toda a capacidade de mapeamento de entrada-saída da RN. A seguir são mostradas as quatro matrizes jacobianas: jacobianas das saídas da função de processo **F** em relação ao estado e ao ruído de processo; e das saídas da função de medida *h* em relação ao estado e ao ruído de medida.

*6.4.3.1 Jacobiana da função **F** em relação ao estado x*

Esta jacobiana é formada pelas derivadas parciais de todas as saídas da função **F** em função das entradas (posições do vetor de estados no instante anterior). Como a primeira posição do vetor de saída da função é calculada pela rede neural, a primeira linha desta jacobiana será obtida em função dos pesos e valores intermediários da RN. Cada derivada parcial (célula da primeira linha da jacobiana) é a derivada da saída da rede das figuras 6.6 e 6.7 (para obter $\hat{x}_1(n|n-1)$) em função de uma das entradas da rede. Por isso que, de acordo com a explicação da seção sobre o aprendizado em redes MLP, a derivada parcial será a multiplicação da derivada da função de ativação da saída pelo somatório das derivadas de cada caminho até chegar nos nós de entrada.

A jacobiana de **F** em função de **x** será uma matriz $[T \times T]$, onde T é o número de entradas da rede (tamanho de **x**). A saída de **F** (vetor $\hat{\mathbf{x}}(n|n-1)$) possui T posições e a entrada (vetor $\hat{\mathbf{x}}(n-1|n-1)$) também possui T posições. A jacobiana será então:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} = \begin{bmatrix} (1,1) & (1,2) & (1,3) & \cdots & (1,T-1) & (1,T) \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{6.20}$$

Então, a primeira linha é extraída da rede neural para obter $\hat{x}_1(n|n-1)$ e possui derivadas parciais em relação a suas duas entradas:

- Derivada parcial em relação a $\hat{x}_1(n-1|n-1)$:

$$(1,1) = \varphi'(s) \sum_{i=1}^{N} w_{1i}^{s} \varphi'(u_i) w_{i1}^{o} \tag{6.21}$$

  Onde:

  - $\varphi'(s)$ é a derivada da função de ativação do neurônio da camada de saída sobre o valor de saída desse neurônio;

  - w^s_{1i} é o peso da ligação do neurônio da camada de saída com o neurônio i da camada oculta;
  - $\varphi'(u_i)$ é a derivada da função de ativação do neurônio i da camada oculta sobre o valor de saída desse neurônio;
  - w^o_{i1} é o peso da ligação do neurônio i da camada oculta com o neurônio 1 da camada de entrada.

- Seguindo da mesma forma, a derivada parcial em relação a $\hat{x}_2(n-1|n-1)$ será:

$$(1,2) = \varphi'(s) \sum_{i=1}^{N} w^s_{1i} \varphi'(u_i)\, w^o_{i2} \tag{6.22}$$

- Generalizando, a derivada parcial em relação a $\hat{x}_T(n-1|n-1)$ será:

$$(1,T) = \varphi'(s) \sum_{i=1}^{N} w^s_{1i} \varphi'(u_i)\, w^o_{iT} \tag{6.23}$$

A segunda linha da matriz é extraída da fórmula para obter $\hat{x}_2(n|n-1)$, a linha 3 é relação $\hat{x}_3(n|n-1)$, a linha T, em relação a $\hat{x}_T(n|n-1)$. Como $\hat{x}_2$ é igual a $\hat{x}_1$ no instante anterior, $\hat{x}_3$ é igual a $\hat{x}_2$, $\hat{x}_T$ igual a $\hat{x}_{T-1}$, tem-se que as posições $(2,1)$, $(3,2)$, $\cdots$, $(T,T-1)$ possuirão valor 1 e as demais posições (das linhas 2 a T) possuirão valor 0.

6.4.3.2 Jacobiana da função **F** *em relação ao ruído de processo* w

Essa jacobiana será uma matriz $[T \times 1]$, porque as T saídas de **F** serão em relação ao valor de w (escalar):

$$\frac{\partial \mathbf{F}}{\partial w} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6.24}$$

A primeira linha da matriz é a derivada parcial da saída $\hat{x}_1(n|n-1)$ em relação a entrada $w(n-1)$. Como são diretamente relacionadas, o valor é 1. As demais linhas são as derivadas parciais das saídas $\hat{x}_k(n|n-1)$ (com k variando de 2 a T) em relação a essa mesma entrada. Como não são relacionadas, o valor é 0.

6.4.3.3 Jacobiana da função h *em relação ao estado x*

Essa jacobiana será uma matriz $[1 \times T]$, pois a saída ($\hat{z}(n|n-1)$) tem 1 posição e a entrada (vetor de estimativa de estado) tem T posições :

$$\frac{\partial h}{\partial \mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{6.25}$$

O primeiro elemento da matriz é a derivada parcial da estimativa de medida $\hat{z}(n)$ em relação a $\hat{x}_1(n|n-1)$, que são diretamente relacionados. Os demais são as derivadas parciais de $\hat{z}(n)$ em relação a $\hat{x}_k(n|n-1)$ (com k variando de 2 a T), que não são relacionados.

6.4.3.4 *Jacobiana da função* h *em relação ao ruído de medida v*

Essa jacobiana será uma matriz $[1 \times 1]$ (escalar), porque tanto a saída como a entrada (ruído de medida v) possuem 1 posição:

$$\frac{\partial h}{\partial v} = [1] \tag{6.26}$$

O único elemento dessa jacobiana é a derivada parcial de $\hat{z}(n|n-1)$ em relação a $v(n)$. Como o valor de saída da função h de medida é diretamente relacionada ao ruído de medida, o valor dessa derivada é 1.

6.5 Comparações com os Trabalhos Correlacionados

O treinamento de redes neurais com Filtro de Kalman e o ajuste de parâmetros do FK com RN são trabalhos "indiretamente correlacionados" a este trabalho. Essas abordagens foram relatadas no capítulo anterior para dar uma visão geral da utilização em conjunto das duas técnicas (RN e FK). As abordagens de treinamento de RNs com FK são mais tradicionais e existem de maneira mais abundante que os métodos denominados híbridos. Alguns dos métodos que podem ser comparados de maneira mais próxima ao modelo neuro-estatístico aqui proposto são os baseados no Neural Extended Kalman Filter (NEKF). No NEKF a RN está na saída do FKE, recebendo apenas o erro deste. Para modelos parcialmente conhecidos essa estrutura funciona satisfatoriamente. Porém, se o modelo real for totalmente desconhecido, a RN terá a função de estimar todo esse modelo desconhecido, de maneira *on-line*, dispondo apenas do erro do FKE a cada iteração. Trabalhando dessa maneira e ainda tendo uma estrutura simples (para poder ser usada de maneira *on-line*), a tarefa da predição se torna difícil para a RN. Essa é uma explicação para o NEKF não ser usado em modelos desconhecidos e que não se conhece a ação tomada (u) em cada instante de tempo, para a construção da trajetória.

No caso das séries temporais abrangidas por este trabalho, o modelo estatístico não é conhecido e não dispõe-se de informações adicionais, como a ação tomada u. Nesses casos também não existe a necessidade de treinamento da rede *on-line*. Assim, pode-se utilizar um conjunto de dados de treinamento, retirado do histórico da série. Com a disponibilidade de um conjunto de dados para treinamento *off-line*, pode-se fazer um treinamento muito mais completo e utilizar uma poderosa estrutura de rede neural (pois a realização de treinamento instantâneo não é mais uma imposição), para obter resultados mais precisos na predição. Outra diferença importante entre o NEKF e o novo método neuro-estatístico é a forma como as não-linearidades são tratadas. No NEKF as não-linearidades passam primeiramente pelo FKE (e por suas jacobianas), sendo que as não-linearidades tratadas pelo FKE são de primeira ordem. Os próprios autores do NEKF comentam que, em dinâmicas altamente não-lineares, o erro cresce significativamente (KRAMER; STUBBERUD, 2005). No método neuro-estatístico, a estimativa de estado (valor desejado da série) é feita primeiramente pela RN, que tem boa capacidade para tratamento de grandes não-linearidades. O fato de a RN simular o processo e de ter acesso a um conjunto de treinamento também auxilia no tratamento de não-linearidades.

O trabalho de (WAN; MERVE, 2000) utiliza uma abordagem semelhante ao método neuro-estatístico, no que se refere à colocação da RN como função de transição de estados do FK e à aplicação para predição de séries temporais. A principal diferença é que nesse trabalho supõe-se a disponibilidade prévia da série limpa (não-ruidosa), para treinamento da RN. Em situações reais, normalmente a série não-ruidosa não está disponível. O artigo

também supõe que o mapeamento de entrada-saída da RN é totalmente conhecido, como se fosse uma função definida f do FK, por exemplo. Os valores exatos dos parâmetros de covariância dos ruídos também são considerados conhecidos *a priori*. Essa aplicação do UKF com RN supõe a presença de condições ideais, facilitando a comparação do UKF com FKE, seu principal objetivo. O artigo indica muitas dificuldades do FKE nos experimentos relatados. No método neuro-estatístico as jacobianas do FKE são calculadas com o mapeamento da RN (construido com os dados de todas as camadas da RN) a cada passo de predição, com a rede sendo treinada com o histórico da série ruidosa. O artigo não mostra esse tipo de estratégia, podendo ser este o motivo das dificuldades do FKE. O método neuro-estatístico, aqui proposto, utiliza condições mais realistas, considerando a não disponibilidade de dados livres de ruído e o não conhecimento dos parâmetros exatos do método.

7 EXPERIMENTOS

Os experimentos, utilizados nas comparações, envolvem o método neuro-estatístico e a rede neural, uma vez que o Filtro de Kalman Estendido necessita da função de transição de estados, não conhecida nas séries tratadas. Para fins de comparação, sempre são utilizados os mesmos modelos e configurações de RN, tanto na rede utilizada isoladamente, como na RN utilizada no método híbrido. Os experimentos foram realizados com duas séries temporais: a série caótica de Mackey-Glass e uma série combinada de funções senos. A diferença para experimentos tradicionais é que ambas as séries são acrescidas de ruído, impondo um desafio extra para a predição e gerando a necessidade de filtragem. São usados dados ruidosos tanto no treinamento da rede, como nas medidas efetuadas durante a execução do método.

7.1 Predição e Filtragem da Série Caótica de Mackey-Glass Acrescida de Ruído

A predição de séries temporais com dinâmicas caóticas é algo muito desafiador para todas as linhas de pesquisa em PST. Mesmo em problemas difíceis como esse, as redes neurais têm apresentado desempenho satisfatório, como no trabalho de (JANG, 1993). A adição de ruído nesse tipo de série torna-se uma novidade ainda mais desafiadora. Uma das séries caóticas, utilizada como *benchmark* (ponto de referência) para a comparação de métodos, é a série de Mackey-Glass (MACKEY; GLASS, 1977), apresentada na figura 7.1.

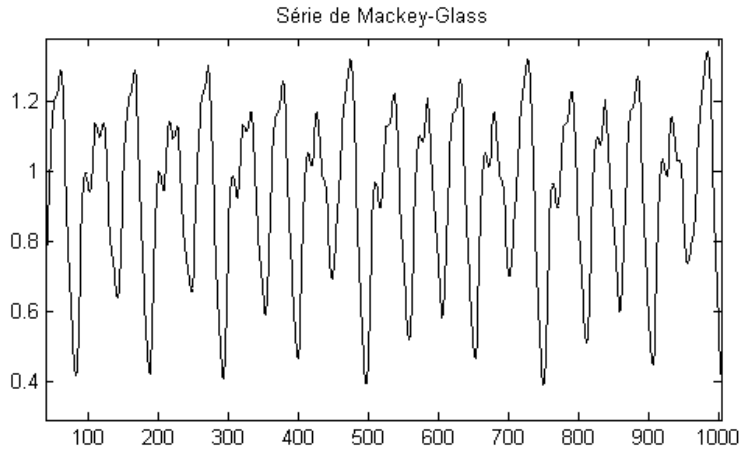

Figura 7.1: Série temporal caótica de Mackey-Glass

A série de Mackey-Glass aqui empregada segue a dinâmica básica utilizada nos trabalhos de (CROWDER, 1991) e (JANG, 1993), em que a variação entre uma posição da série e a próxima é descrita por:

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{7.1}$$

Onde $\dot{x}(t)$ é a variação de valor da série no instante t, comparando-se com o instante anterior; $x(t-\tau)$ é o valor da série, τ posições atrás. O trecho da série de Mackey-Glass mostrado na figura 7.1 possui $\tau = 17$.

O problema considerado como *benchmark* para pesquisadores conexionistas é a predição de valores futuros dessa série, no instante $k = t+P$, sendo P um valor inteiro positivo (JANG, 1993). Para a predição de um valor futuro, P posições a frente, são utilizados D amostras anteriores, espaçadas em Δ posições entre elas.

7.1.1 Configurações Utilizadas nos Experimentos

Para que os experimentos fossem apresentados de maneira próxima às definições originais de (CROWDER, 1991) e (JANG, 1993), adotou-se muitas configurações desses trabalhos. Configurou-se $D = 4$, ou seja, são utilizados 4 valores de posições anteriores (4 entradas para a rede neural). O valor de P foi escolhido como 6, então o valor previsto será 6 posições a frente do atual ($t+6$). Também foi configurado $\Delta = 6$ como o espaçamento entre as posições de entrada. Atribuiu-se também $\tau = 17$ (definindo a periodicidade e complexidade da série).

Gerou-se um conjunto de dados, para extração das amostras de treinamento e teste, com $0 \leq t \leq 1617$ (assumiu-se que a série possui valores nulos para $t < 0$). Extraiu-se 1000 amostras para treinamento com $118 \leq t \leq 1117$, seguindo os trabalhos citados. Cada amostra de treinamento, segue o seguinte formato:

$$[x(t-18), x(t-12), x(t-6), x(t); x(t+6)] \tag{7.2}$$

Onde os 4 primeiros valores servem como entrada e o último como saída desejada. As 500 amostras de teste foram extraídas com $1118 \leq t \leq 1617$, tendo o mesmo formato dos dados de treinamento, porém sem o valor desejado (último valor). A RN utiliza a função de ativação tangente hiperbólica. Tanto no treinamento quanto no teste, a RN recebe os dados (ruidosos) da própria série como entrada. Ou seja, utilizou-se o método neuro-estatístico com modelo de entrada NAR, da figura 5.3. Nesse caso, a predição (resultado a priori do método) será a mesma da RN sozinha. Os valores melhorados são os da saída a priori (filtragem). Os dados filtrados podem ser usados para fazer um retreinamento da RN em outros experimentos.

7.1.2 Predição da Série Sem Ruído

Na predição de séries sem ruído não é necessária a utilização do método neuro-estatístico. Apenas aplicou-se uma rede neural, nesse caso, para avaliar o poder preditivo desse modelo de RN na predição da série de Mackey-Glass convencional. Com isso, pode-se comparar o erro deste experimento com os erros que a incidência de ruído provoca na RN, justificando a necessidade da filtragem do método neuro-estatístico (para posterior retreinamento da RN).

Para esse experimento, a rede MLP utilizada contém 10 neurônios na camada oculta e 400 épocas de treinamento. Todas as configurações desse capítulo de experimentos foram

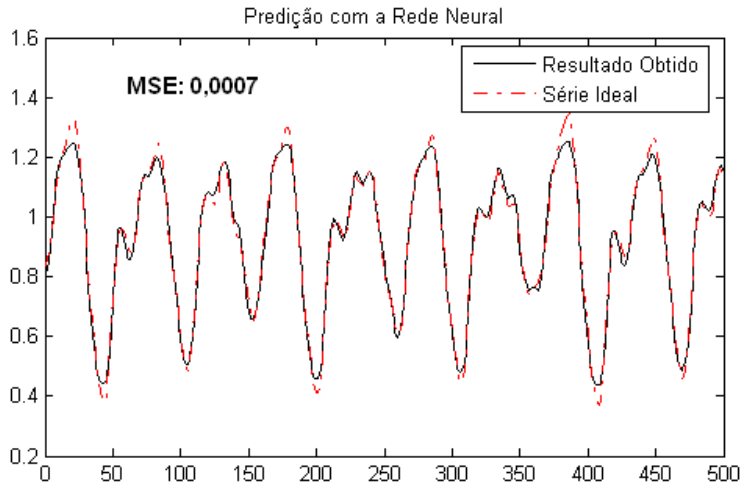

Figura 7.2: Predição da série de Mackey-Glass não-ruidosa com a rede neural

escolhidas a partir de grande número de execuções experimentais dos métodos, com faixas de cada parâmetro a ser escolhido. A figura 7.2 mostra o resultado da predição da RN, com erro médio quadrado de 0,0007. Pode-se perceber uma boa capacidade preditiva por parte de uma rede MLP com algoritmo *backpropagation*, com apenas uma camada oculta e poucos neurônios.

7.1.3 Utilização do Método Neuro-estatístico com Ruído Pequeno

O ruído utilizado neste trabalho é *gaussiano branco*, isto é, com distribuição normal e média zero. O ruído é aditivo (somado aos valores da série) e possui variância R. A adição de ruído serve para simular as imprecisões na obtenção dos dados. Nos exemplos das figuras 7.3 e 7.4, a variância R tem valor 0,01, tendo então desvio padrão de 0,1.

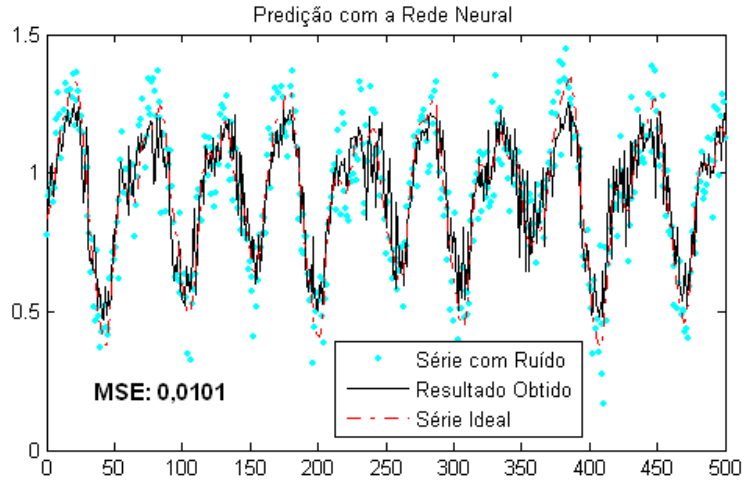

Figura 7.3: Resultado da predição da rede neural para a série com 0,01 de variância de ruído

A RN para predição da série de Mackey-Glass (agora ruidosa) possui a mesma configuração que o exemplo da série não ruidosa, com 10 neurônios na camada oculta, 400 épocas de treinamento, taxa de aprendizado 0,1 e coeficiente de Momentum 0,5. Essa configuração também é utilizada na RN que faz parte do NE. Os resultados da predição

da série, apenas com a RN são mostrados na figura 7.3. O erro da RN piora bastante apenas com a adição de um pouco de ruído. Percebe-se também uma tendência de generalização, aplainando os máximos e mínimos da função da série.

O método neuro-estatístico é então aplicado para filtragem, com a sua parte "rede neural" configurada da mesma forma que a RN mostrada anteriormente. A parte "Filtro de Kalman Estendido" do NE necessita apenas da configuração dos parâmetros Q e R, variância de ruído de processo e variância de ruído de medida. O valor de R é configurado com 0,01 (valor do ruído verdadeiro). A opção por colocar os valores exatos do parâmetro R é para possibilitar a análise das demais características do método, sem o viés dos mecanismos de escolha (medida) desse parâmetro. Na seção 7.3 são apresentadas opções de medida desse parâmetro diretamente dos dados e realiza-se uma análise sobre os erros dessa medida. O valor de Q é configurado com um valor um pouco acima da média de erros da RN, nesse caso foi configurado como 0,013. A justificativa para esse tipo de escolha também é mostrada na seção 7.3.

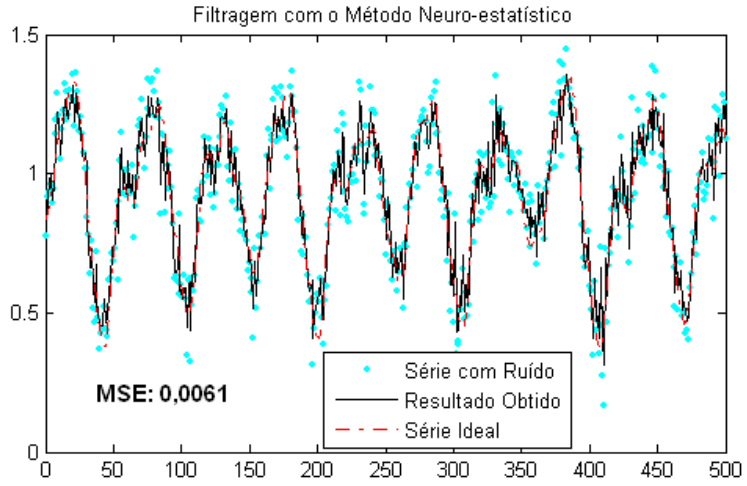

Figura 7.4: Resultado da filtragem do método neuro-estatístico para a série com 0,01 de variância de ruído

Os resultados da filtragem do NE são mostrados na figura 7.4. Pode-se observar uma significativa diminuição do Erro Médio Quadrado (MSE), comparando-se com os resultados da rede neural. Observa-se também que o MSE dos valores filtrados pelo NE fica bem abaixo da variância do ruído de medida, diminuindo consideravelmente o grau de ruído nos dados.

7.1.4 Utilização do Método Neuro-estatístico com Ruído Médio

O "ruído médio", aqui denominado, é gaussiano branco com variância 0,04. Esse ruído é mostrado juntamente com os resultados das figuras 7.5 e 7.6, representado pelos pontos nos gráficos. A rede neural, em ambos métodos, também possui a mesma configuração dos exemplos anteriores, com 10 neurônios na camada oculta e 400 épocas de treinamento. Os resultados da aplicação da RN com essa configuração são mostrados na figura 7.5. O erro cresce bastante com o aumento do ruído, ocorrendo uma tendência de arredondamento de curvas. Pode-se perceber também, com ruído maior, que a RN antecipa ou aumenta algumas curvas da série.

A configuração dos parâmetros Q e R do método NE são feitas da mesma forma que o experimento anterior. A variância do ruído de medida R foi agora configurada como

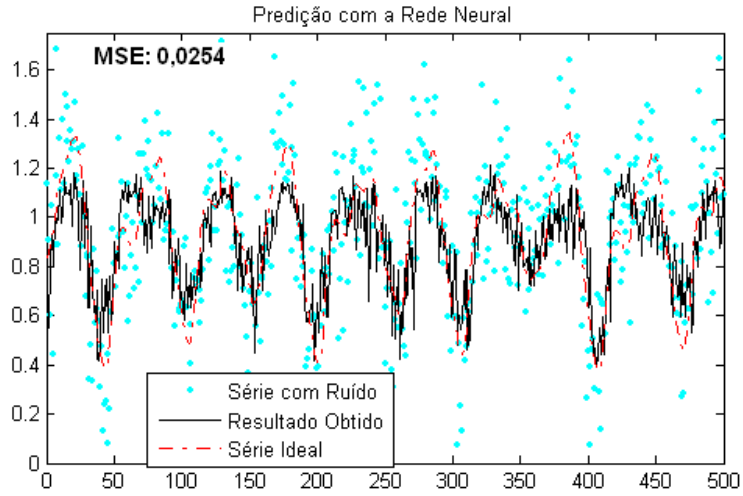

Figura 7.5: Resultado da predição da rede neural para a série com 0,04 de variância de ruído

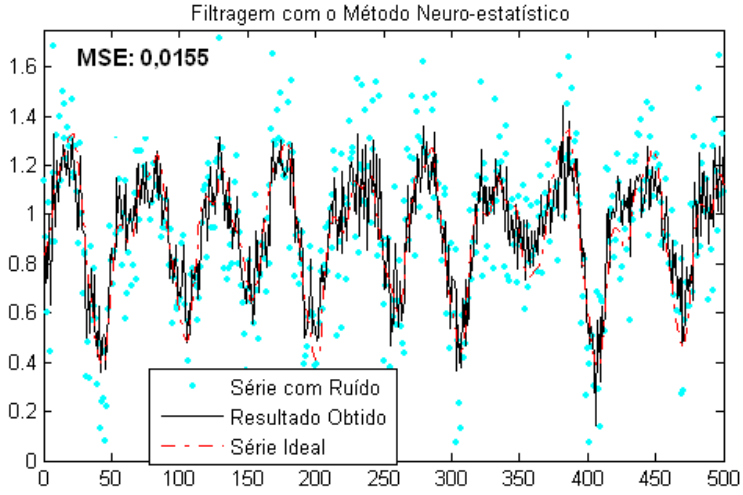

Figura 7.6: Resultado da filtragem do método neuro-estatístico para a série com 0,04 de variância de ruído

sendo 0,04 (ruído da série) e a variância do ruído de processo foi configurada como 0,03 (um pouco acima do erro médio esperado da RN). A figura 7.6 mostra o gráfico de valores filtrados pelo NE. Observa-se que o erro médio foi bem menor que o encontrado pela RN, conseguindo acompanhar a trajetória da série, mesmo com o ruído presente nos dados medidos. O erro (0,0155) também está consideravelmente menor que a covariância do ruído de medida (0,04), significando uma boa eficiência na filtragem, mesmo com o erro da RN sendo mais alto.

7.1.5 Utilização do Método Neuro-estatístico com Ruído Grande

O ruído, aqui chamado "grande" também é gaussiano branco, agora com variância 0,09. O aumento do grau de ruído (variância) serve para uma melhor análise das dificuldades que as incertezas causam na predição por redes neurais. Percebe-se pela distância dos pontos até a curva ideal, nas figuras 7.7 e 7.8, a grande incidência de ruído nesses exemplos. A configuração utilizada pela RN é a mesma dos experimentos anteriores.

O desempenho da predição da RN na série bastante ruidosa é mostrado na figura 7.7.

Percebe-se novamente que a RN suaviza demasiadamente em certos pontos (considerando parte da série verdadeira como sendo ruído) e considera demais o ruído em outros pontos (considerando o ruído como parte da série). O desempenho da filtragem do NE é mostrado na figura 7.8, apresentando uma melhora nessas características e obtendo mais uma vez erro (0,0274) significativamente menor, comparando com a RN e com a variância do ruído. Isso se deve à ponderação dada pelo método para a predição e para a medida. Se fosse dada uma importância muito grande para a predição, o erro ficaria muito próximo de 0,0364 (erro de saída da RN). Por outro lado, se fosse dada uma importância muito grande para a medida o erro ficaria muito próximo de 0,09 (covariância do ruído de medida).

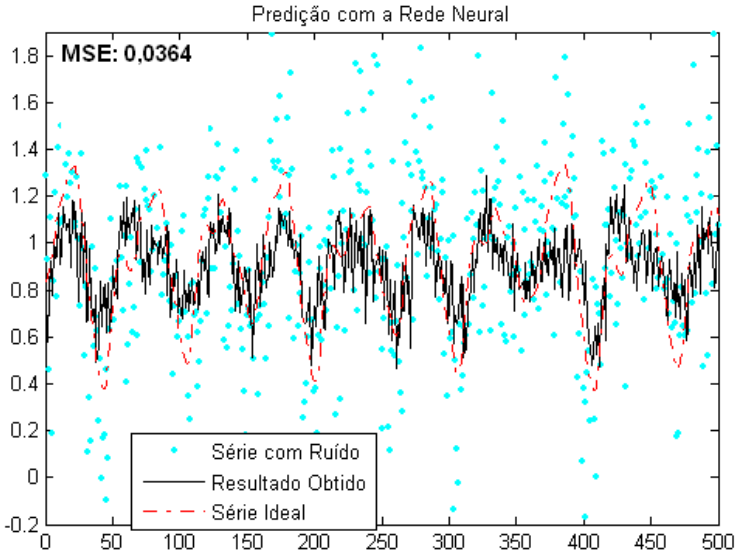

Figura 7.7: Resultado da predição da rede neural para a série com 0,09 de variância de ruído

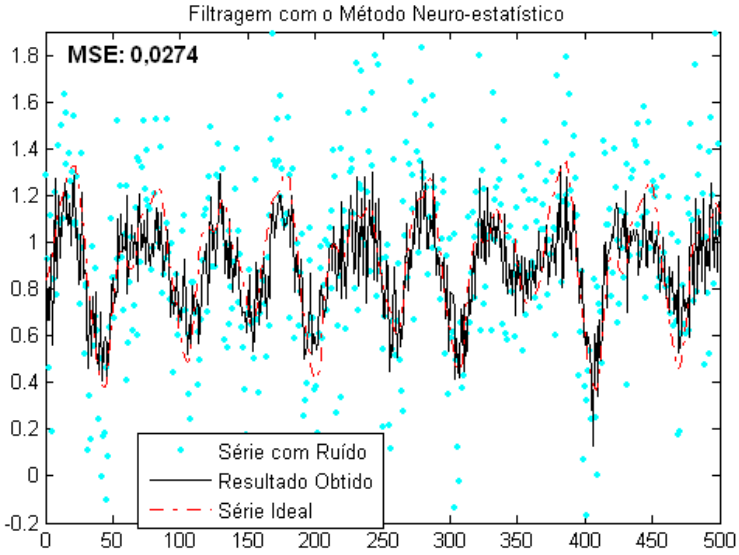

Figura 7.8: Resultado da filtragem do método neuro-estatístico para a série com 0,09 de variância de ruído

Para analisar e confirmar o maior erro da RN em regiões de picos da série temporal, gerou-se os gráficos de erro da RN e do método neuro-estatístico. O gráfico de erro da

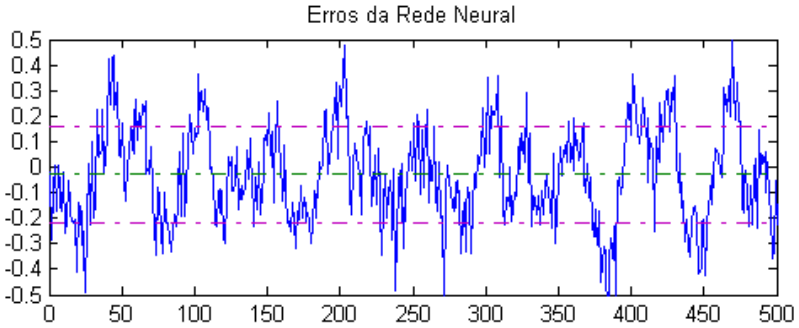

Figura 7.9: Gráfico de erro da rede neural

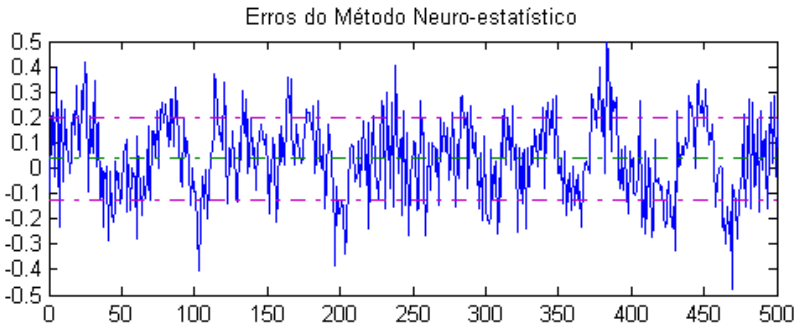

Figura 7.10: Gráfico de erro do método neuro-estatístico

RN é mostrado na figura 7.9, apresentando regiões específicas com erros maiores. Essas regiões normalmente coincidem com as inversões de tendência da série. A RN apresenta então regiões com tendência de erros maiores. O gráfico de erro da filtragem do NE é mostrado na figura 7.10, onde nota-se uma regularidade maior na amplitude do erro ao longo da série. As linhas tracejadas também indicam um menor desvio padrão por parte do NE.

7.1.6 Resumo dos Resultados para a Série Mackey-Glass

Os resultados aqui sintetizados foram gerados com os métodos RN e NE, com as mesmas configurações das subseções anteriores, computando-se os resultados de 10 execuções diferentes para cada método, em cada nível de ruído. Computou-se a média e o desvio padrão de cada conjunto de 10 execuções, para proporcionar maior confiabilidade estatística nos valores.

Os valores de ruído aleatório são diferentes para cada par de execuções. Gera-se o ruído, executa-se os dois métodos (RN e NE), geram-se valores de ruído novamente, e assim por diante. A mudança dos valores aleatórios de ruído a cada execução serve para melhorar a qualidade estatística das comparações, sendo que a execução para a par (RN e NE com os mesmos valores de ruído em cada vez) proporciona maior imparcialidade. Em todos os gráficos mostrados para a série de Mackey-Glass, a saída filtrada do método neuro-estatístico (estimativa a posteriori) é comparada com a saída da RN pura, que é a mesma saída a priori do método, pois utilizou-se o modelo NAR para a entrada-saída. A saída a posteriori do método poderia ser utilizada para retreinar a sua RN.

A tabela 7.1 mostra um resumo dos erros médios e desvios padrões do erro, para as predições da rede neural e do método neuro-estatístico, nos três níveis de ruído. O NE consegue melhorar bastante os resultados da RN em todos os casos. Tanto a RN quanto

Tabela 7.1: Média dos erros e desvios padrões do erro para a RN e o NE

Variância do ruído	MSE médio RN	Desvio MSE RN	MSE médio NE	Desvio MSE NE
0,01	0,0098	0,0009	0,0053	0,0004
0,04	0,0283	0,0024	0,0177	0,0018
0,09	0,0371	0,0028	0,0268	0,0020

o método NE conseguem ter MSEs abaixo da variância do ruído. Os desvios padrões do erros permanecem proporcionais aos valores absolutos desses erros, em todos os casos. O erro da RN cresce muito no início da adição de ruído. Quando o ruído torna-se muito grande, o erro da RN tende a estabilizar com a adição de mais ruído, pois o pior caso é considerar todos os dados como ruído e prever o ponto central. Mostrou-se que a RN possui dificuldades nos pontos extremos das curvas (picos), tendendo a achatá-los ou a criar falsos picos. O método neuro-estatístico ameniza bastante esses problemas.

7.2 Predição de Série de Combinação de Senos Acrescida de Ruído

A criação de séries a partir de uma composição de funções trigonométricas (principalmente seno e cosseno) proporciona o aparecimento de séries difíceis com não-linearidades bastante complexas. Uma série apresentada por (HAYKIN, 2001a) como desafio para a área de redes neurais é a série dada pela seguinte função:

$$x(n) = \sin\left(n + \sin\left(n^2\right)\right) \tag{7.3}$$

Onde $\sin(\cdot)$ representa a função seno. A função combina o seno de um valor inteiro n acrescido do seno desse mesmo valor ao quadrado. A figura 7.11 mostra um trecho dessa série, com $1 \leq n \leq 100$. A inserção do seno de n^2 dentro de outro seno cria uma série de difícil predição. A existência de ciclos muito curtos também aumenta a dificuldade.

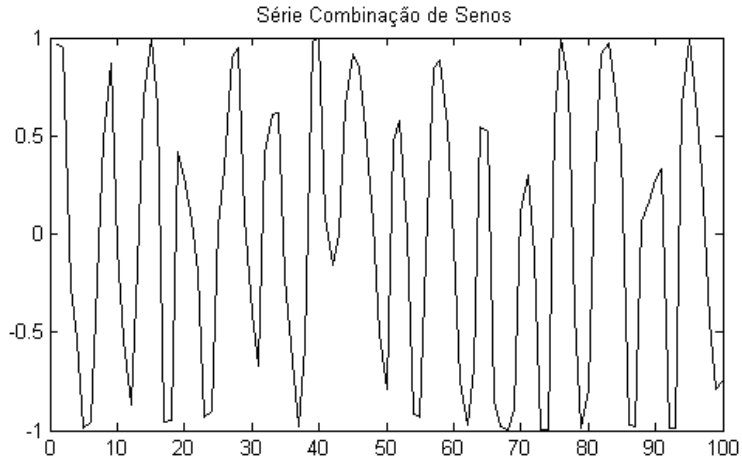

Figura 7.11: Série temporal não-linear gerada a partir de combinação de senos

A predição da série dos senos combinados, acrescida de ruído, gerou a justificativa inicial desse trabalho, pois experimentos iniciais com redes neurais já indicavam a sua grande dificuldade. Para a predição dessa série sem ruído, a RN necessita de uma grande estrutura e consegue fazer predições razoáveis. Porém, quando essa série é acrescida de ruído, a RN possui grande dificuldade de predição, apresentando erros muito altos.

7.2.1 Configurações e Estratégias Utilizadas nos Experimentos

O conjunto de treinamento, para a RN e para a rede do NE, é composto de amostras seqüenciais, no seguinte formato:

$$[x(t-T), x(t-T+1), \cdots, x(t-2), x(t-1); x(t)] \tag{7.4}$$

Onde T é o número de valores anteriores (atrasos) que a RN recebe como entrada e $x(t)$ é o valor da série no instante atual. Esse valor do instante atual é ruidoso (quando passado como entrada para o treinamento), sendo posteriormente suavizado pelos métodos. Escolheu-se o número de 13 atrasos para a realização dos experimentos. Então a RN recebe 13 valores seqüenciais de entrada e um valor desejado no treinamento. Para o teste, a RN receberá apenas os valores de entrada (diferentes do treinamento):

$$[x(t-13), x(t-12), \cdots, x(t-2), x(t-1)] \tag{7.5}$$

Gerou-se 1000 amostras para a extração do conjunto de treinamento e 500 amostras para o conjunto de teste. Para o conjunto de treinamento, utilizou-se $14 \leq t \leq 1013$ e para o conjunto de teste, $1014 \leq t \leq 1513$. Todos os experimentos foram realizados com a RN tendo a configuração de 35 neurônios na camada oculta, 400 épocas de treinamento, taxa de aprendizado de 0,1, coeficiente de Momentum de 0,5 e função de ativação tangente hiperbólica. Também utilizou-se a estratégia de colocação de 2000 amostras intercaladas de treinamento (com posições escolhidas aleatoriamente) e a realimentação da RN com valores já suavizados pela parte do FKE. Essa abordagem de realimentação segue o modelo apresentado na figura 6.4. Essas duas estratégias são comentadas a seguir.

Para uma maior eficiência do treinamento da RN, em vez de passar a seqüência de amostras de treinamento na ordem original, passam-se as amostras em ordem aleatória. Estabelece-se um certo valor de amostras a serem utilizadas em cada época e escolhe-se aleatoriamente o t (dentro do limite definido para treinamento) para cada uma dessas amostras. Assim o conjunto de treinamento fica muito mais heterogêneo, melhorando significativamente a qualidade de treinamento da RN. Também pode ser aumentado o número de amostras, podendo diminuir o número de épocas de treinamento. Comprovou-se, nos experimentos, que o aumento da quantidade de amostras por época gera melhores resultados que o aumento do número de épocas.

Outra estratégia utilizada é a realimentação do método NE com os valores anteriores previstos pelo próprio método. Em vez de receber as entradas, a RN do método receberá os valores já suavizados nos instantes anteriores (menos ruidosos). Essa variação na entrada-saída no método corresponde ao modelo apresentado na figura 5.4. Comprovou-se que, mesmo que a RN tenha sido treinada com dados altamente ruidosos, essa RN apresentará melhores resultados quando receber dados com menor grau de ruído.

7.2.2 Predição da Série Sem Ruído

Para a predição da série combinada de senos sem ruído, utilizou-se apenas a rede neural, com a configuração otimizada por experimentos, conforme descrito anteriormente. Os resultados da predição da série não-ruidosa, através da RN isoladamente, servem para mostrar a grande diferença de erro quando o ruído for adicionado (mesmo em pequena quantidade). A predição dessa série (sem incidência de ruído) pela RN possui uma taxa aceitável de erro e é mostrada na figura 7.12.

Pode-se perceber, na figura 7.12, alguns erros nas regiões de inversão de tendência (picos) da série. Observa-se também na figura, que ocorrem poucos retardos ou antecipa-

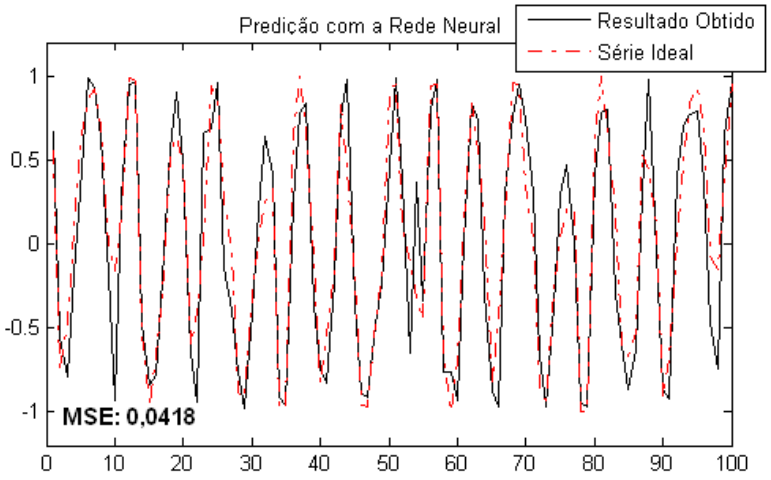

Figura 7.12: Predição da série não-ruidosa com uma rede neural

ções de tendências, concentrando os erros apenas nos picos. Esses outros tipos de erros serão observados nas seções seguintes, quando a série é ruidosa.

7.2.3 Comparações Utilizando Ruído Pequeno

Esta subseção inicia as comparações da RN com o método neuro-estatístico com presença de ruído, na série combinada de senos. Todas as comparações utilizam a mesma rede (com os mesmos pesos) no método NE também, para uma maior confiabilidade nas comparações. Todos os ruídos destas comparações são gaussianos brancos aditivos. O ruído aqui considerado pequeno também possui variância de 0,01. O parâmetro R do método neuro-estatístico é configurado como 0,01, enquanto o Q é configurado como 0,09.

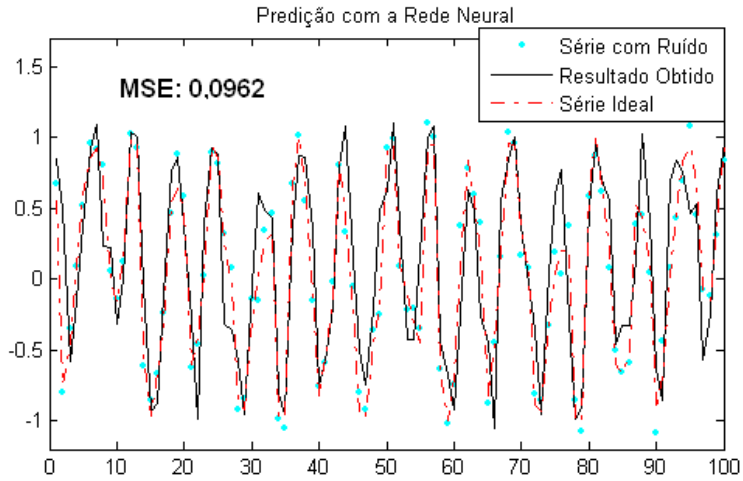

Figura 7.13: Resultado da RN na predição da série com 0,01 de variância de ruído

O gráfico de resultados obtidos e desejados da RN é mostrado na figura 7.13. O acréscimo desse pequeno grau de ruído causa um grande aumento do erro, em relação à série sem ruído. Além dos erros na intensidade das tendências detectadas na predição sem ruído, também ocorrem agora antecipações e retardo na detecção dessas tendências.

A figura 7.14 mostra os resultados da filtragem do método neuro-estatístico, que apresenta um erro médio quadrado muito mais baixo: 0,0081. A obtenção de erros muito abaixo dos da RN, mesmo utilizando a própria rede como processo, se deve aos mecanis-

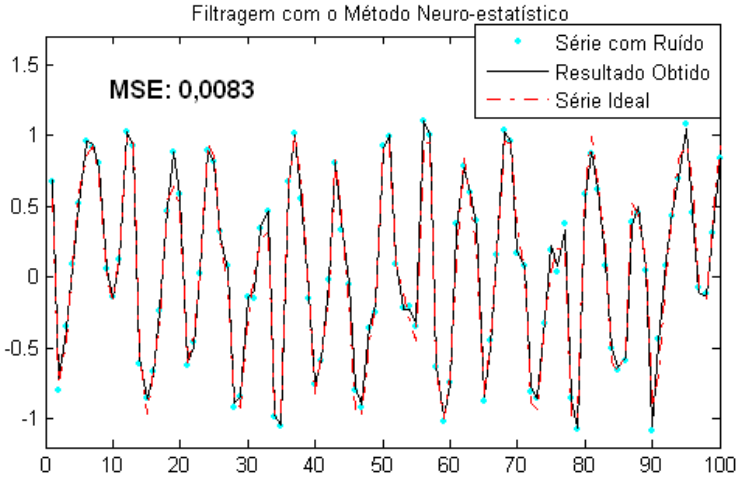

Figura 7.14: Resultado do NE na filtragem da série com 0,01 de variância de ruído

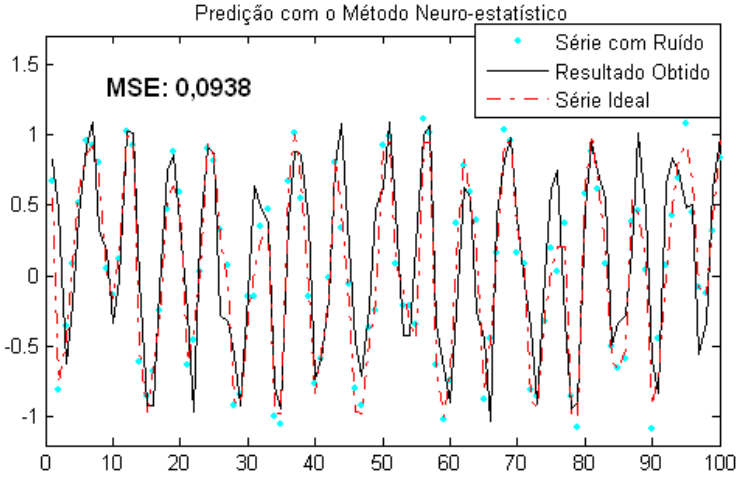

Figura 7.15: Resultado do NE na predição da série com 0,01 de variância de ruído

mos de ponderação pelo Ganho de Kalman, internos do NE. Como a RN apresenta um erro muito alto, o NE dá uma importância maior para as medidas, "situando" a previsão da RN quando esta está muito longe da série. Observa-se que, mesmo dando uma importância maior para a medida, o NE consegue ter um erro menor que a variância do ruído de medida, conseguindo realizar a filtragem da série. Na figura 7.15 é mostrada a predição do NE, utilizando os valores filtrados na entrada da RN, em vez de usar as medidas ruidosas. Percebe-se uma leve diminuição do erro da RN do NE em relação a RN sem realimentação, recebendo os valores filtrados. Essa diminuição ocorre porque a RN recebe entradas com variância de erro de 0,0083 em vez de receber entradas com erro de 0,01 (ruído de medida).

7.2.4 Comparações Utilizando Ruído Médio

O ruído "médio" possui variância de 0,04. As configurações da RN são as mesmas do experimento anterior e o NE possui o parâmetro R configurado como 0,04 e o Q como 0,15 (um pouco acima da variância aproximada do erro da RN). A figura 7.16 mostra os resultados da RN. Percebe-se que o erro da RN cresce bastante com o aumento do ruído. Em relação ao ruído menor, crescem os picos de erros nas inversões de tendências.

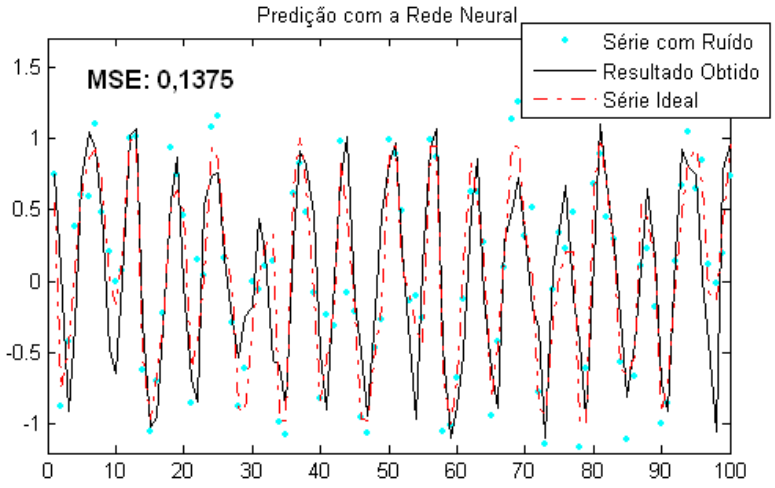

Figura 7.16: Resultado da predição pela RN para a série com 0,04 de variância de ruído

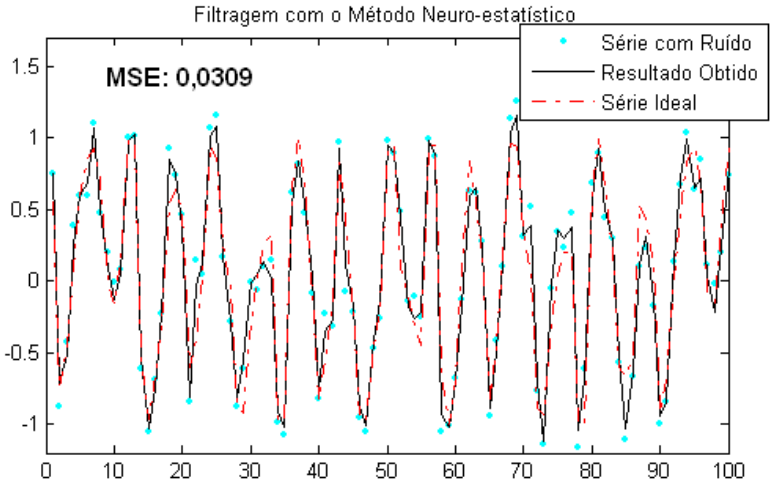

Figura 7.17: Resultado da filtragem pelo NE, para a série com 0,04 de variância de ruído

O desempenho da filtragem do NE para esse nível de ruído é mostrado na figura 7.17. Observa-se que o NE diminui o erro da RN em todas as partes da trajetória, sem deixar picos de erro. Em regiões da série onde o erro da RN é muito grande, o valor calculado pelo NE fica muito próximo da medida ruidosa. Essa opção de ajuste é calculada automaticamente pelo método, através do Ganho de Kalman, gerado a partir das covariâncias dos erros. A escolha de dar maior importância para a medida é feita de acordo com o crescimento do erro do processo (RN) e permanece até o erro diminuir. A preparação do NE para um possível grande erro da RN vai ocorrendo gradativamente ao longo das iterações. Por exemplo, em um primeiro grande erro da RN para menos, o filtro do NE irá compensar parte do erro. Se, depois disso, houver outro grande erro no mesmo sentido do primeiro, o erro será mais fortemente compensado. O método parecerá "vacinado" contra o erro. Da mesma forma, ocorre também gradativamente o esquecimento do erro (na matriz de covariância **P**).

O resultado da predição do método neuro-estatístico é mostrado na figura 7.18. Percebe-se uma melhora no desempenho da RN do NE ao receber os dados com ruído menor. A RN permanece com o mesmo treinamento (realizado com ruído de variância 0,04), mas passa a receber como entrada os dados filtrados (com erro de variância 0,0309). Essa

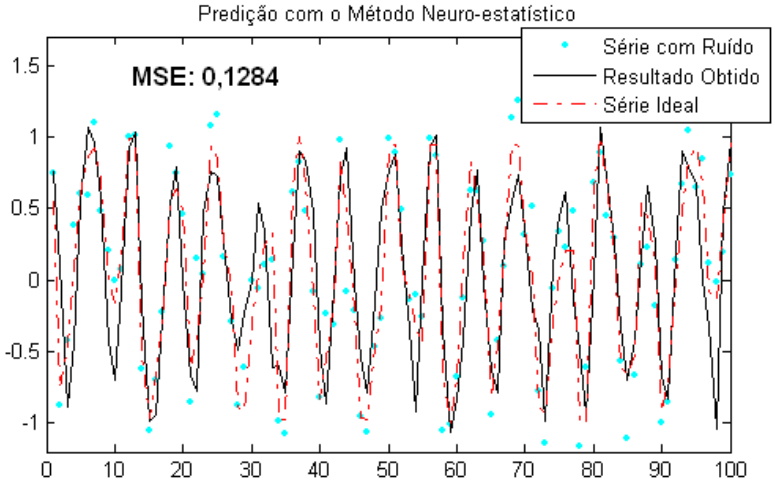

Figura 7.18: Resultado da predição pelo NE, para a série com 0,04 de variância de ruído

diferença de ruído faz a RN melhorar o MSE de 0,1375 para 0,1284.

7.2.5 Comparações Utilizando Ruído Grande

O ruído "grande" possui variância de 0,09. Seguindo a mesma linha que as configurações dos demais experimentos, os valores de R e Q do NE foram respectivamente 0,09 e 0,22 (próximo do MSE estimado da RN). O resultado da aplicação da RN é mostrado na figura 7.19. Como mostrado na figura, o erro da RN cresceu ainda mais com o aumento do ruído, tendendo a aplainar algumas curvas da trajetória e desviar outras. Quando mais cresce o ruído, percebe-se que a RN (atuando isoladamente) tende a simplificar a série, desconsiderando comportamentos (e curvas) mais complexos.

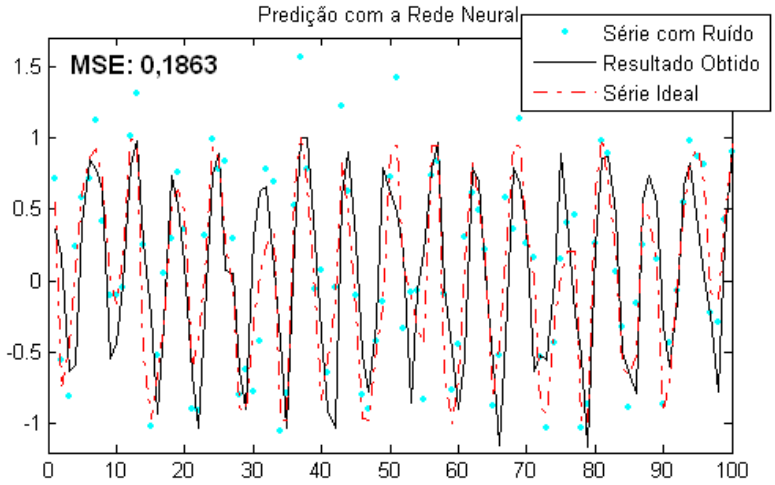

Figura 7.19: Resultado da RN para a série com 0,09 de variância de ruído

Na figura 7.20 está o gráfico de resultados da filtragem do método neuro-estatístico, aplicado na série acrescida de ruído grande. Percebe-se um melhor ajuste à verdadeira série. Ocorre um bom nível de filtragem em relação à variância do ruído, mesmo com o grande erro da RN.

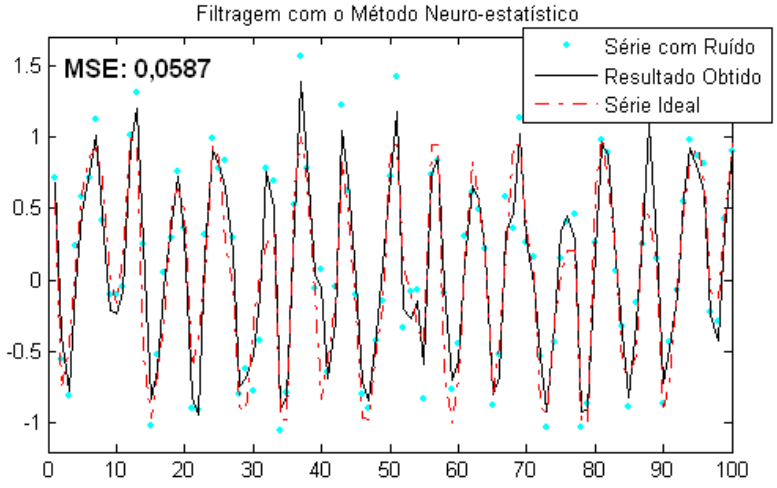

Figura 7.20: Resultado do NE para filtragem da série com 0,09 de variância de ruído

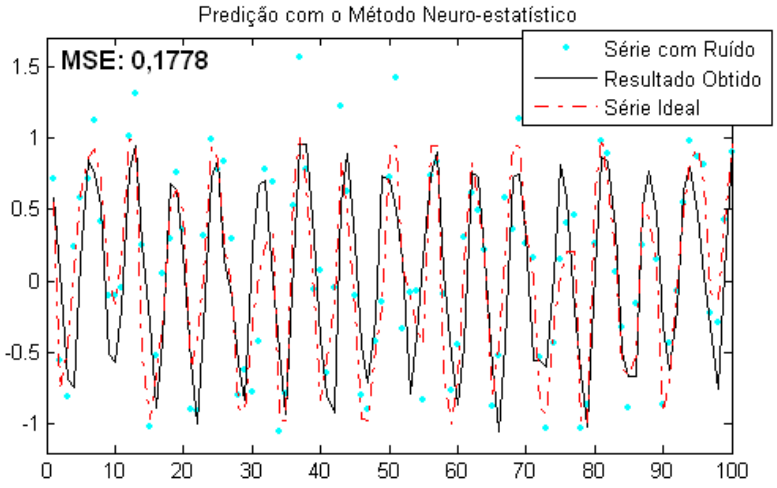

Figura 7.21: Resultado do NE para predição da série com 0,09 de variância de ruído

7.2.6 Resumo dos Resultados da Série

Os resultados para a série combinada de senos são sintetizados na tabela 7.2. A tabela apresenta uma comparação dos erros de predição da RN e dos erros de predição e filtragem do NE, para 3 níveis de ruído. Percebe-se uma melhora no desempenho da RN do NE ao receber os dados com ruído menor. A RN permanece com o mesmo treinamento (realizado com os dados ruidosos), mas passa a receber como entrada os dados filtrados (com variância de erro menor).

Tabela 7.2: Erros Médios Quadrados para a RN e o NE

Variância do ruído	Predição RN	Predição NE	Filtragem NE
0,01	0,0962	0,0938	0,0083
0,04	0,1375	0,1284	0,0309
0,09	0,1863	0,1778	0,0587

O funcionamento do FKE em conjunto com a RN propicia que um método passe progressivamente resultados melhores para o outro. No início das iterações do NE a RN

começa a gerar estimativas com MSE correspondente ao seu próprio erro (quando atuando isoladamente). A parte "filtro" do NE irá melhorar a estimativa da rede e irá passar esse valor melhorado como uma das entradas da RN para o passo seguinte. Depois de T iterações, a RN já estará recebendo todos os valores filtrados e passará valores ainda melhores para o filtro. O ciclo se repete com o filtro conseguindo estimativas ainda melhores e passando para a RN prever também ainda melhor. Relembra-se que, para todos os experimentos com essa série, o método neuro-estatístico utilizou sempre a mesma RN com a qual foi comparado, com os mesmos pesos do treinamento. Esses efeitos observam-se na predição do NE, mostrada na tabela 7.2. Com ruído de variância 0,09, o NE melhorou o MSE de 0,1863 para 0,1778. A RN do NE recebe dados com ruído de variância 0,0587 (MSE da filtragem) em vez de 0,09, melhorando o seu desempenho.

7.3 Análise Prática sobre o Ajuste dos Parâmetros Q e R

O ajuste correto de parâmetros é importante para o bom funcionamento do FK e, conseqüentemente, do método neuro-estatístico. O parâmetro Q representa a covariância do ruído de processo, ou seja, as imprecisões do processo em relação ao verdadeiro modelo da série. Como o processo do método neuro-estatístico é a própria RN, o ruído de processo será o MSE da rede em relação à série filtrada (sem ruído). O parâmetro R é a covariância do ruído de medida, ou seja, o MSE entre as medidas ruidosas e a série ideal (não ruidosa). Os valores exatos desses parâmetros não são conhecidos, sendo possível fazer estimativas sobre eles. Nesta seção é mostrada uma análise sobre a conseqüência dos erros de estimação desses parâmetros e são apresentadas algumas medidas estatísticas para estimá-los.

7.3.1 Análise Sobre Ajustamento Não Otimizado de Parâmetros

Como mostrado anteriormente, o valor de ruído (erro) da RN não será sempre conhecido exatamente, mas pode ser colocado um valor aproximado como parâmetro Q. Para visualizar as repercussões de se atribuir um valor menor ou maior que o ideal para esse parâmetro, realizaram-se experimentos utilizando uma grande quantidade de valores diferentes de parâmetros. O experimento foi realizado com o mais ruidoso de todos os exemplos tratados neste capítulo: a série composta de senos com variância de ruído de 0,09. As configurações utilizadas são as mesmas relatadas nos demais experimentos com essa série. Na execução da RN atuando isoladamente para esse problema obteve-se MSE de 0,2069. O método neuro-estatístico utilizou a mesma RN treinada (não alterando os pesos) para todos as execuções.

A figura 7.22 mostra os valores de erro da filtragem do NE para configurações de Q variando de 0,13 até 0,31. Observa-se que os menores valores de MSE do método estão com Q entre 0,19 e 0,20. O valor ótimo de Q nesse caso é um pouquinho abaixo do próprio MSE da RN, pois a rede diminui um pouco o erro ao longo das iterações do NE, como explicado anteriormente. Observa-se também que o aumento do MSE é maior quando o parâmetro Q é configurado abaixo do ideal que quando configurado acima do ideal. Para confirmar essa tendência foram realizadas execuções com valores extremos (muito grandes e muito pequenos) de Q.

A figura 7.23 mostra os testes com valores muito pequenos para Q. Observa-se que o erro cresce exponencialmente até estabilizar em um valor muito alto (próximo do erro que uma previsão de linha reta no ponto médio do eixo y da série geraria). A figura 7.24 mostra a utilização de valores muito grandes para Q. O erro cresce mais suavemente, que

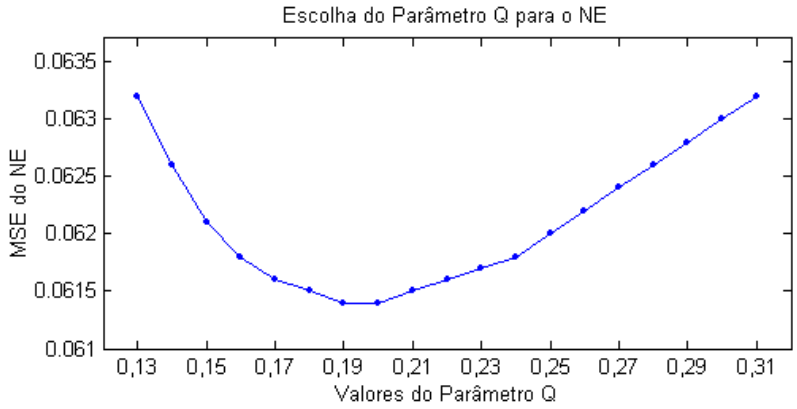

Figura 7.22: Curva de variação do MSE do NE conforme o parâmetro Q

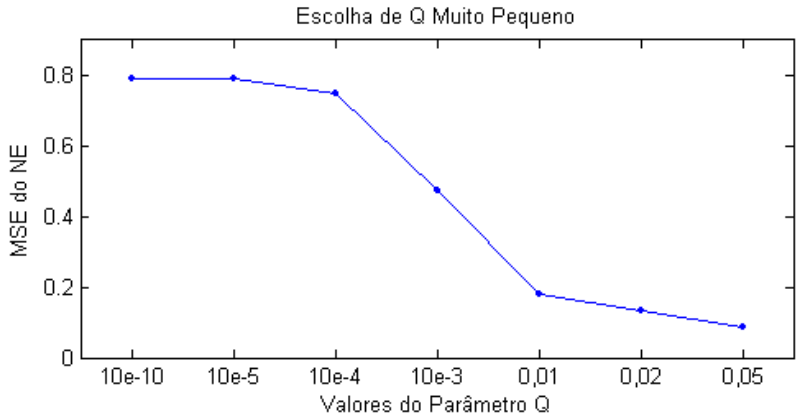

Figura 7.23: Curva do MSE do NE para a escolha de Q muito pequeno

na figura 7.23, com o aumento do valor do parâmetro. Com um valor de Q muito grande, o MSE estabiliza em um valor um pouco abaixo da variância do erro de medida. Esse erro se deve ao fato de o método considerar, nesse caso, quase exclusivamente a medida. Conclui-se que as conseqüências de escolher Q maior que o ideal são menores do que se for escolhido um valor menor que o ideal.

Outra constatação importante, nos experimentos realizados, é que quando sabe-se a relação entre Q e R, a configuração correta dos valores absolutos não é necessária. Os parâmetros $Q = 0.2$; $R = 0.09$ darão o mesmo erro que $Q = 0.4$; $R = 0.18$ ou que $Q = 0.002$; $R = 0.0009$. Essa proporção também pode ser aproximada da ideal, pois somente se a diferença de proporção for muito grande é que haverá alguma diferença significativa no erro final do método NE. Por exemplo se o erro de proporção for 2 (o dobro ou a metade do ideal), o MSE do NE será apenas 5,8% maior que o erro gerado pela configuração ideal.

7.3.2 Medidas Estatísticas para a Especificação de Parâmetros

Como o parâmetro Q refere-se ao erro da RN em relação à série ideal, esse parâmetro pode ser estimado através de medidas estatísticas do erro da rede. Essas medidas podem ser obtidas em execuções da RN para predições de termos da série. Os valores desejados para o cálculo do erro podem ser os valores da série ruidosa, gerando uma especificação mais imprecisa do erro (parâmetro). Também podem ser usados dados suavizados por um filtro ou pelo próprio método neuro-estatístico com parâmetros menos otimizados. Com várias medidas de erro da RN (de várias execuções), pode-se calcular a média dos MSEs

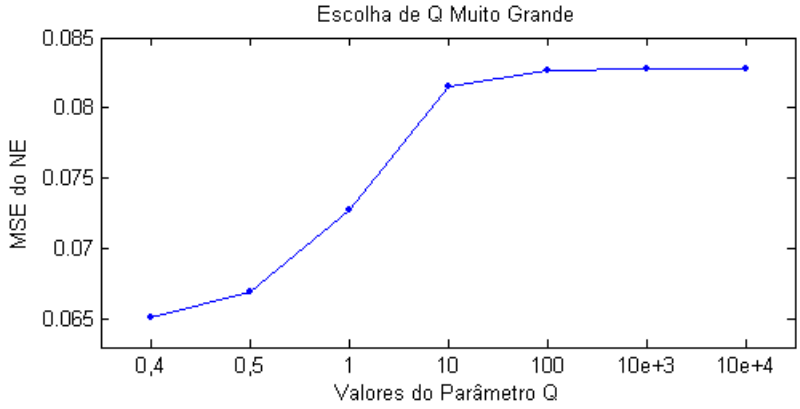

Figura 7.24: Curva do MSE do NE para a escolha de Q muito grande

da rede. O valor escolhido como parâmetro pode ser um pouco acima dessa média, porque as conseqüências (no aumento do erro do método) de configurar para mais o parâmetro são menores do que configurando para menos, conforme mostrado anteriormente.

Quanto ao ajuste do parâmetro R (covariância do ruído de medida), este deve ser estimado a partir de medidas ruidosas da série. Porém não se sabe o que é a série (não ruidosa) e o que é o ruído. A extração da covariância de toda a série ruidosa implicaria em considerar tudo como ruído, inclusive os ciclos da série ideal. Para que esses ciclos não influenciem na medida do ruído, uma solução é utilizar janelas muito curtas, onde a variação do ciclo não seja significativa. O mínimo tamanho possível para o cálculo da covariância é duas posições da série. Depois da escolha do tamanho da janela, desloca-se essa janela por toda a série e calcula-se a média de todas as covariâncias calculadas. Para confirmação dessa estratégia, realizou-se um experimento com a série de Mackey-Glass, com covariância real de 0,01 para o ruído de medida. Testou-se diferentes janelas, de 2 até 20, como mostrado na figura 7.25. Os valores de janela pequena apresentam um valor de parâmetro bem próximo do ideal. Para janelas muito grandes, a covariância estimada é maior pelo fato de estar mais suscetível às oscilações da série.

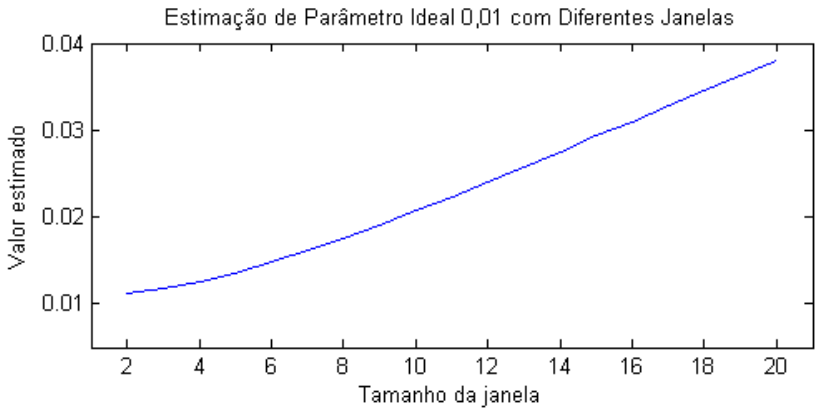

Figura 7.25: Estimação do ruído de medida na série de Mackey-Glass

Para séries de ciclo longo, como a Mackey-Glass, a estratégia de cálculo de covariâncias em janelas pequenas funciona satisfatoriamente, como mostrado na figura 7.25. Em séries com ciclo muito curto, como a combinada de senos, mesmo a mínima janela não será suficiente para evitar oscilações da série. Nessa série há uma oscilação muito grande entre um instante e outro. Como obedece um ciclo semelhante ao seno, em apro-

ximadamente 3 instantes a série variará de -1 para 1. Então em média a série variará aproximadamente 2/3 em 1 passo. A estratégia é descontar a covariância dessa estimativa de tendência da covariância total da janela a cada passo. Por exemplo, a covariância do vetor $0 \quad 0,6666$ é 0,2222. Com esse mecanismo, tornou-se possível estimar, com janela de 2 posições, os valores de parâmetros na série combinada de senos, como mostrado na figura 7.26. O valor estimado do parâmetro mantém-se um pouco acima do ideal, em todos os graus de ruído. Observa-se também que a razão entre os valores de parâmetros diminui com o aumento do ruído, sendo que os ruídos maiores são mais fáceis de medir.

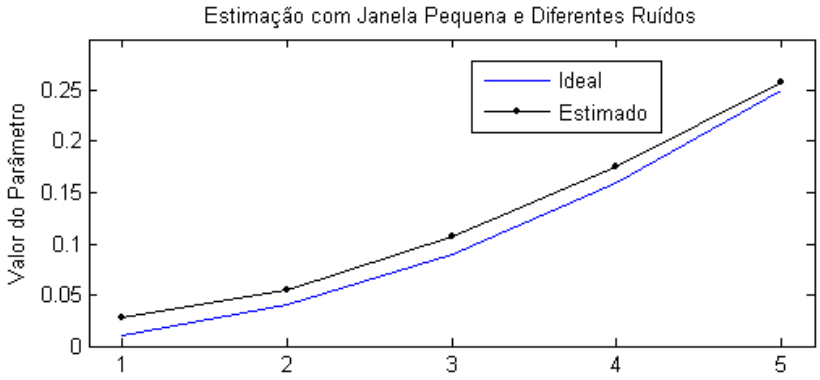

Figura 7.26: Estimação do ruído de medida na série combinada de senos

Essa estratégia de descontar a covariância da variação de amplitude média da série é mais eficiente quando se tem um bom conhecimento dos tamanhos médios dos ciclos. Mesmo com as dificuldades de ajuste exato do parâmetro, a diferença no MSE final do método não é tão grande. Por exemplo, na figura 7.26 é mostrada uma diferença na escolha do parâmetro ideal $R = 0,09$. No exemplo, é atribuído o valor de 0,107. Essa diferença de quase 20% a mais irá produzir um acréscimo no MSE final de apenas 0,3%. Pode-se concluir que erros pequenos (menores que 50%) no ajuste dos parâmetros não causam aumento significativo no erro final. Com isso, as estimativas de parâmetros aqui apresentadas proporcionam desempenhos satisfatórios para o método.

8 CONSIDERAÇÕES FINAIS

Neste capítulo será realizado um apanhado geral das idéias do método proposto e da sua estrutura, ressaltando as principais comparações e sintetizando os resultados. Também são comentadas as sugestões de trabalhos futuros que este trabalho proporcionou.

8.1 Conclusões

Este trabalho apresentou uma nova abordagem para a predição de séries temporais, aplicando conjuntamente uma rede neural de múltiplas camadas com o método estatístico Filtro de Kalman Estendido. O novo método pode ser utilizado em séries com grandes não-linearidades, modelo gerador desconhecido e com incidência de ruído nas medições das entradas. A RN atua como processo previsor do FKE, auxiliando na predição do modelo não-linear desconhecido da série. O restante do FKE filtra o ruído, iterativamente com a RN, melhorando o desempenho de todo o conjunto (FKE e RN juntos) do método.

A utilização de uma RN como processo do FKE aumenta muito a aplicabilidade que o filtro possui isoladamente. O Filtro de Kalman e suas variantes só podem ser aplicados quando o modelo estatístico da série é conhecido. Em séries com modelo parcialmente conhecido e com necessidade de predição em tempo real, pode ser usado um método híbrido como o Neural Extended Kalman Filter (NEKF). O novo método neuro-estatístico (NE) atende a necessidade de predição de séries em que o modelo é totalmente desconhecido, com conjuntos de dados para treinamento *off-line*, como grande parte das séries temporais mais importantes atualmente. Grandes não-linearidades nessas séries também podem ser tratadas mais cuidadosamente pela colocação da RN como centro do processo preditivo e pela possibilidade de a RN possuir uma poderosa estrutura de camadas ocultas e grande quantidade de neurônios nessas camadas, como é feito neste trabalho. O método proposto adapta-se à condições realistas de aplicações, como o treinamento com dados ruidosos e imprecisões nas estimativas dos parâmetros.

Os resultados do método neuro-estatístico em predição e filtragem são comparados com os resultados da mesma arquitetura de rede MLP utilizada na estrutura do método. As comparações são feitas a partir de experimentos em dois modelos de séries temporais, acrescidos de ruído: a famosa série caótica de Mackey-Glass; e uma série combinada de senos, utilizada como desafio na área de redes neurais. Em ambos *benchmarks* o método NE obteve resultados satisfatórios, melhorando o resultado da RN "pura" em todos os experimentos. O método NE também ajustou-se melhor aos picos das séries, detectando melhor as tendências dos ciclos. O erro médio quadrado (MSE) do método também permaneceu sempre abaixo da variância do ruído, podendo ser considerado um bom filtro.

O método NE funciona sem o conhecimento dos valores exatos de ruído de medida

(variância R) e ruído de processo (variância Q). O método depende apenas do conhecimento de uma proporção aproximada entre os parâmetros Q e R. Mostra-se como esses parâmetros podem ser estimados. O parâmetro Q é aproximado pelo MSE da saída da RN e o parâmetro R, pela passagem prévia de um filtro nos dados da série. De acordo com experimentos realizados, mesmo que os parâmetros tenham valores distantes do ideal, o acréscimo no erro final do método será pequeno.

O método NE aprende com os erros da própria rede neural interna, utilizando as matrizes de covariâncias dos erros e Ganho de Kalman. Quando a previsão da RN está desviando em um sentido, as matrizes internas do método ajustam-se para corrigir o erro, adaptando-se novamente quando o erro baixar. O método híbrido proporciona um aprendizado duplo (algoritmo de treinamento da RN e covariâncias do erro do FKE) com os dois métodos alimentando-se mutuamente, explicando os bons resultados obtidos.

8.2 Sugestões de Trabalhos Futuros

Uma das sugestões de trabalhos futuros é a realização de novos treinamentos da RN do método NE, quando o método tiver filtrado parcialmente ou totalmente os dados. Em uma primeira passada, a série poderia ser totalmente filtrada, utilizando também o NE sobre o conjunto de treinamento. Em uma segunda passada, a RN do método seria treinada com parte da série já suavizada, preferencialmente a parte que era conjunto de teste no passo anterior. A segunda passada completaria após prever novamente o agora conjunto de treinamento. A suavização da série de trás para frente também poderia ser realizada, intercalando-se cada passo desses com outra passada na ordem original. As suavizações no sentido inverso serviriam para compensar o período de ajuste do método em cada passada. Sugere-se um estudo dessa abordagem de predições sucessivas para análise de taxa de diminuição do erro a cada nova passada.

Também pode ser feita a tentativa de implementação do método híbrido *on-line*, utilizando uma RN de treinamento instantâneo como a Rede de Elman (ELMAN, 1990). Teoricamente o novo método teria uma menor capacidade preditiva mas poderia ser aplicado em predições de séries em tempo real. Esse método teria aplicações semelhantes às do NEKF e mais algumas, por possuir a RN como processo. Esse método neuro-estatístico *on-line* seria uma alternativa para problemas de detecção de trajetórias em tempo real, quando o modelo de trajetória é totalmente desconhecido.

Outro trabalho proposto é a implementação do método neuro-estatístico com o UKF para comparar com a versão com o FKE. Outra sugestão é a utilização de ruídos com distribuições não-gaussianas no FK. Seria feita uma análise da capacidade de um método neuro-estatístico para tratar esse tipo de ruído e a relevância desse tratamento.

REFERÊNCIAS

BOX, G. E. P.; JENKINS, G. M.; REINSEL, G. C. **Time Series Analysis**: forecasting and control. 3rd ed. New Jersey, USA: Prentice Hall, 1994.

BROWN, R. G. **Smoothing, Forecasting and Prediction of Discrete Time Series**. [S.l.]: Prentice-Hall International, 1963.

CASTRO, M. C. F. de. **Predição Não-Linear de Séries Temporais Usando Redes Neurais RBF por Decomposição em Componentes Principais**. 2001. Tese (Doutorado em Ciência da Computação) — Universidade Estadual de Campinas, Campinas, BR.

CLOUSE, D. S. et al. Time-Delay Neural Networks: representation and induction of finite-state machines. **IEEE Transactions on Neural Networks**, [S.l.], v.8, n.5, p.1065–1070, Sept. 1997.

CORTEZ, P. A. R. **Algoritmos Genéticos e Redes Neuronais na Previsão de Séries Temporais**. 1997. Dissertação (Mestrado em Ciência da Computação) — Universidade do Minho, Braga, PT.

CROWDER, R. S. Predicting the Mackey-Glass Timeseries with Cascade-correlation Learning. In: CONNECTIONIST MODELS SUMMER SCHOOL, 1990. **Proceedings...** San Mateo: CA: Morgan Kaufmann, 1991. p.117–123.

CYBENKO, G. Approximation by Superpositions of a Sigmoidal Function. **Mathematics of Control, Signal and Sytems**, [S.l.], v.2, p.303–314, 1989.

DECRUYENAERE, J. P.; HAFEZ, H. M. A Comparison Between Kalman Filters and Recurrent Neural Networks. In: JOINT CONFERENCE ON NEURAL NETWORKS, IJCNN, 1992, Baltimore, MD. **Proceedings...** [S.l.: s.n.], 1992. p.247–251.

ELMAN, J. L. Finding Structure in Time. **Cognitive Science**, [S.l.], 1990.

ENGEL, P. M. Redes neurais artificiais : uma visão geral das suas potenciais aplicações. In: FóRUM DE INTELIGêNCIA ARTIFICIAL DA REGIãO SUL, 2001, Canoas, BR. **Anais...** Canoas: Ulbra, 2001. 1 CD-ROM.

ENGEL, P. M. **Filtro de Kalman**. Notas de Aula da disciplina de Sistemas Conexionistas Avançados, Segundo semestre de 2005. PPGC da UFRGS.

CROCE FILHO, J. **Estatística II**. Juiz de Fora, BR: Universidade Federal de Juiz de Fora, 2000. Disponível em: <http://twiki.dcc.ufba.br>. Acesso em: dez. 2006.

FISHER, W. A.; RAUCH, H. E. Augmentation of an Extended Kalman Filter with a Neural Network. In: IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE; IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, 1994. **Proceedings...** Piscataway: IEEE, 1994. v.2, p.1191–1196.

GANG, L.; YU, F. A hybrid nonlinear autoregressive neural network for permanent-magnet linear synchronous motor identification. In: INTERNATIONAL CONFERENCE ON ELECTRICAL MACHINES AND SYSTEMS, ICEMS, 8., 2005. **Proceedings...** Beijing: Internacional Academic Publishers, 2005. v.1, p.310–314.

GIRONDEL, V.; CAPLIER, A.; BONNAUD, L. Real Time Tracking of Multiple Persons by Kalman Filtering and Face Pursuit for Multimedia Applications. In: IEEE SOUTHWEST SYMPOSIUM ON IMAGE ANALYSIS AND INTERPRETATION, 6., 2004. **Proceedings...** Piscataway: NJ: IEEE, 2004. p.201–205.

GLYMOUR, C. et al. Statistical Inference and Data Mining. **Communications of the ACM**, New York, v.39, n.11, p.35–41, Nov. 1996.

GUANG-FU, M.; XUE-YUAN, J. Unscented Kalman Filter for Spacecraft Attitude Estimation and Calibration Using Magnetometer Measurements. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS, 2005. **Proceedings...** [S.l.: s.n.], 2005. v.1, p.506–511.

HAYKIN, S. **Redes Neurais**: princípios e prática. 2.ed. Porto Alegre, BR: Bookman, 2001. 900p. Tradução da 2. ed. por Paulo Martins Engel.

HAYKIN, S. **Kalman Filter and Neural Networks**. Ontario, CA: John Wiley & Sons, 2001.

HAYKIN, S. **Communication Systems**. 4th ed. New York, USA: John Wiley & Sons, 2001. 816p.

JANG, J.-S. R. ANFIS: adaptive-network-based fuzzy inference system. **IEEE Transactions on Systems, Man and Cybernetics**, [S.l.], v.23, n.3, p.665–685, May 1993.

JORIS, R. F. **Extração de Conhecimento de Redes Neurais Artificiais Usando Seleção de Atributos**. 2005. Dissertação (Mestrado em Ciência da Computação) — Pontifícia Universidade Católica do Rio Grande do Sul.

KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. **Transactions of the Journal of Basic Engineering, ASME**, [S.l.], v.82, n.Series D, p.35–45, 1960.

KOHONEN, T. The Self-Organizing Map. **Proceedings of the IEEE**, Piscataway, v.78, n.9, p.1464–1480, Sept. 1990.

KORNIYENKO, O. V.; SHARAWI, M. S.; ALOI, D. N. Neural Network Based Approach for Tuning Kalman Filter. In: IEEE INTERNATIONAL CONFERENCE ON ELECTRO INFORMATION TECHNOLOGY, 2005. **Proceedings...** [S.l.: s.n.], 2005. p.1–5.

KOVÁCS, Z. L. **Redes Neurais Artificiais**: fundamentos e aplicações. 3.ed. São Paulo, BR: Livraria da Física, 2002. 174p.

KRAMER, K. A.; STUBBERUD, S. C. Impact Time and Point Predicted Using a Neural Extended Kalman Filter. In: INTERNATIONAL CONFERENCE ON INTELLIGENT SENSORS, SENSOR NETWORKS AND INFORMATION PROCESSING CONFERENCE, 2005. **Proceedings...** [S.l.]:IEEE, 2005. p.199–2004.

LAVIOLA, J. J. A Comparison of Unscented and Extended Kalman Filtering for Estimating Quaternion Motion. In: AMERICAN CONTROL CONFERENCE, 2003. **Proceedings...** [S.l.: s.n.], 2003. v.3, p.2435–2440.

MACHADO, K. F. **Módulo de Auto-Localização para um Agente Exploratório usando Filtro de Kalman**. 2003. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul, Porto Alegre, BR.

MACKEY, M. C.; GLASS, L. Oscillation and Chaos in Physiological Control Systems. **Science**, [S.l.], n.197, p.287–289, July 1977.

MANTOVANI, G. F. **Previsão de Séries Temporais Redes Neurais Artificiais vs. Modelos ARIMA**. 2004. 62f. Monografia (Bacharelado em Estatística) - Instituto de Matemática, UFRGS, Porto Alegre.

MCCULLOCH, W. S.; PITTS, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. **Bulletin of Mathematical Biophysics**, [S.l.], v.5, p.115–133, 1943.

MORETTIN, P. A.; TOLOI, C. M. C. **Modelos para Previsão de Séries Temporais**. Poços de Caldas, MG: 13º Colóquio Brasileiro de Matemática, 1981. v.2.

MORETTIN, P. A.; TOLOI, C. M. C. **Análise de Séries Temporais**. São Paulo, BR: Blücher, 2004. 535p.

NUNES, R. C. **Adaptação Dinâmica do *timeout* de Detectores de Defeitos através do Uso de Séries Temporais**. 2003. Tese (Doutorado em Ciência da Computação) — Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, BR.

NYGREN, I.; JANSSON, M. Terrain Navigation for Underwater Vehicles Using the Correlator Method. **IEEE Journal of Oceanic Engineering**, [S.l.], v.29, n.3, p.906–915, July 2004.

OLIVEIRA, G. A. **Sistema de Controle de Estoques Utilizando a Metodologia Box & Jenkis de Séries Temporais**. 2002. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal do Paraná, Curitiba, BR.

OWEN, M. W.; STUBBERUD, S. C. Interacting Multiple Model Tracking Using a Neural Extended Kalman Filter. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, IJCNN, 1999. **Proceedings...** [S.l.: s.n.], 1999. v.4, p.2788–2791.

OWEN, M. W.; STUBBERUD, S. C. A Neural Extended Kalman Filter Multiple Model Tracker. In: OCEANS, 2003. **Proceedings...** [S.l.: s.n.], 2003. v.4, p.2111–2119.

PAYLE, D. **Data Preparation for Data Mining**. San Francisco, USA: Morgan Kaufmann, 1999. 540p.

RUMELHART, D. E. et al. Learning Internal Representation by Error Propagation. **Parallel Distributed Processing:** explorations in the microstucture of cognition, Cambridge: The MIT Press, 1986. v.1, p.318–362.

RUSSELL, S. J.; NORVIG, P. **Inteligência Artificial**. 2.ed. Rio de Janeiro, BR: Campus, 2004.

RUTGEERTS, J. et al. A Demonstration Tool with Kalman Filter Data Processing for Robot Programing by Human Demonstration. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2005. **Proceedings...** [S.l.: s.n.], 2005.

SHUHUI, L. Comparative analysis of backpropagation and extended Kalman filter in pattern and batch forms for training neural networks. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, IJCNN, 2001. **Proceedings...** [S.l.]: IEEE, 2001. v.1, p.144–149.

STUBBERUD, S. C.; KRAMER, K. A. A 2-D Intercept Problem Using the Neural Extended Kalman Filter for Tracking and Linear Predictions. In: SOUTHEASTERN SYMPOSIUM ON SYSTEM THEORY, SSST, 37., 2005. **Proceedings...** [S.l.: s.n.], 2005. p.367–372.

STUBBERUD, S. C.; LOBBIA, R. N.; OWEN, M. An Adaptive Extended Kalman Filter Using Artificial Neural Networks. In: IEEE CONFERENCE ON DECISION AND CONTROL, 37., 1995, New Orleans, LA. **Proceedings...** [S.l.: s.n.], 1995. v.2, p.1852–1856.

STUBBERUD, S. C.; OWEN, M. W. Targeted On-line Modeling for an Extended Kalman Filter Using Artificial Neural Networks. In: IEEE INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS; IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE, 1998. **Proceedings...** [S.l.: s.n.], 1998. v.2, p.1019–1023.

TAKENGA, C. M. et al. Comparison of Gradient Descent Method, Kalman Filtering and Decoupled Kalman in Training Neural Networks used for Fingerprint-Based Positioning. In: IEEE VEHICULAR TECHNOLOGY CONFERENCE, 60, 2004. **Proceedings...** [S.l.]: IEEE, 2004. v.6, p.4146–4150.

TAYLOR, W. K. Electrical Simulation of Some Nervous System Functional Activities. **Information Theory**, [S.l.], v.3, p.314–328, 1956.

VEPA, R. Application of neuro-Kalman filtering to attitude estimation of platforms and space vehicles. In: IEE COLLOQUIUM ON HIGH ACCURACY PLATFORM CONTROL IN SPACE, 1993. **Proceedings...** [S.l.]: IEE, 1993. v.5, p.1–3.

WAN, E. A. Times Series Prediction by Using a Connectionist Network with Internal Delay Lines. In: NATO ADVANCED RESEARCH WORKSHOP ON COMPARATIVE TIMES SERIES ANALYSIS, 1992, Santa Fe N. M. **Proceedings...** Reading, MA: Addison-Wesley, 1994.

WAN, E. A.; MERVE, R. V. The Unscented Kalman Filter for Nonlinear Estimation. In: IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE. ADAPTIVE SYSTEMS FOR SIGNAL PROCESSING, COMMUNICATIONS, AND CONTROL SYMPOSIUM, 2000, Lake Louise, Alta, Canada. **Proceedings...** [S.l.: s.n.], 2000. p.153–158.

WAN, E. A.; MERVE, R. V.; NELSON, A. T. Dual Estimation and the Unscented Transformation. **Advances in Neural Information Processing Systems**, [S.l.], n.12, p.666–672, 2000.

WELCH, G.; BISHOP, G. **An Introduction to the Kalman Filter**. Chapel Hill: University of North Carolina, 2001. Technical report.

YEE, L.; JIANG-HONG, M.; WEN-XIU, Z. A New Method for Mining Regression Classes in Large Data Sets. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], v.23, 2001.

ZHAN, R.; WAN, J. Neural Network-aided Adaptive Unscented Kalman Filter for Nonlinear State Estimation. **IEEE Signal Processing Letters**, [S.l.], v.13, n.7, p.445–448, July 2006.