

Identificação Neural com Uso do Sinal de Erro de Predição

R. C. L. de Oliveira¹, F. M. de Azevedo², J. M. Barreto³

¹Lab. de Controle e Sistemas, Depart. de Engenharia Elétrica, Univ. Federal do Pará
Caixa Postal 8619, 66.075-970, Belém, Pará, Brasil
limao@gpeb.ufsc.br, jalb@guama.cpgee.ufpa.br

²Grupo de Pesq. em Eng. Bioméd., Depart. de Engenharia Elétrica, Univ. Federal de Santa Catarina
Caixa Postal 476, 88.040-900, Florianópolis, Santa Catarina, Brasil
azevedo@gpeb.ufsc.br

³Depart. de Informática e Estatística, Univ. Federal de Santa Catarina
88.040-900, Florianópolis, Santa Catarina, Brasil
barreto@inf.ufsc.br

Abstract

This work shows a neural net model with dynamic neurons where the dynamics is represented by state space approach. This neural model is utilized to identify non-linear dynamic systems with non-linear measure noise. In the identification is used the prediction error how a new neural net input.

1. Introdução

Os procedimentos de identificação paramétrica para sistemas lineares são bem consolidados e apresentam inúmeros resultados teóricos e práticos [1]; para sistemas não-lineares o mesmo não acontece [2]. Mais recentemente se tem utilizado o paradigma conexionista das Rede Neurais Artificiais (RNA's) para a tarefa de identificação não-linear com as mesmas sendo tratadas como modelos não-lineares gerais, sem a distinção de classes [3][4][5].

Normalmente, na identificação neural não-linear, tem-se utilizado modelos neurais onde a informação da dinâmica da mesma está representada na forma externa. Esta representação é efetivada através do conjunto de sinais $u(.)$ e $y(.)$ (entrada e saída do processo) atrasados no tempo na entrada da RNA, acarretando em uma representação entrada/saída para o modelo [3][4]. Outra forma de representação para o modelo neural é o espaço de estados, onde o conjunto de sinais atrasados no tempo na entrada da rede é substituído por laços de realimentação na rede neural [5]. Esta última representação tem a vantagem de utilizar um menor número de parâmetros, de utilizar o conceito de estado, apresentando a desvantagem de necessitar de algoritmos de aprendizado que atentem para o problema de instabilidade do modelo neural.

2. Rede Neural com Neurônios Dinâmicos

A figura (1) mostra um modelo geral do neurônio artificial, desenvolvido por De Azevedo[6].

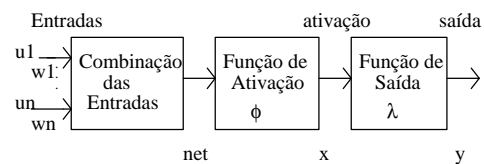


Figura 1 - Modelo de neurônio artificial

Este modelo possibilita a inclusão de dinâmica através da função de ativação, responsável pelo estado do neurônio, e apresenta a função de saída estática que gera o sinal de saída do neurônio. As equações que regem o funcionamento deste neurônio são mostradas a seguir;

$$net(t) = \sum_{i=1}^n w_i u_i \quad (1)$$

$$x(t+h) = \mathbf{f}[x(t), w_i, u_i, t, h] \quad (2)$$

$$y(t) = \mathbf{I}(x(t), t) \quad (3)$$

Se o neurônio é estático, então a função de ativação é do tipo identidade, ou seja;

$$x(t) = net(t) \quad (4)$$

Com este formalismo do neurônio artificial, este mesmo trabalho [6] define uma RNA como sendo um grafo, onde os vértices são os neurônios e os arcos que ligam os vértices são os pesos da rede.

A partir das definições de De Azevedo [6], este trabalho utiliza um novo modelo de RNA, visto na figura (2), onde a dinâmica do processo é representada por uma realimentação linear do estado do neurônio e com a não-linearidade sendo representada pela função de saída do mesmo [5]. O treinamento é realizado por um algoritmo de aprendizado, do tipo *backpropagation*. O modelo neural utilizado tem como diferenças

principais, em relação aos modelos neurais localmente recorrentes apresentados na literatura [7] : a definição de um estado para o neurônio, que é diferente do sinal de saída do mesmo; a utilização deste estado na realimentação do próprio neurônio; esta realimentação apresentar uma ponderação variável; a utilização de um modelo discreto para a rede neural e a possibilidade de conexões entre neurônios de uma mesma camada.

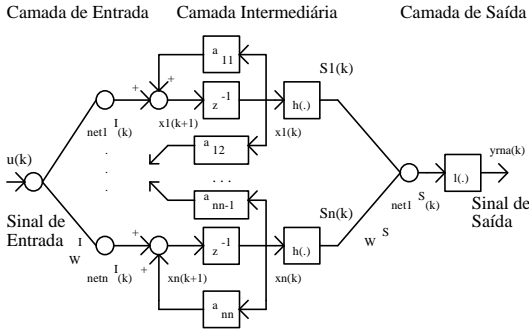


Figura 2 - RNA com uma camada intermediária de neurônios dinâmicos e uma camada de saída com neurônios estáticos

3. Identificação Neural

Atualmente as RNA's têm sido bastante exploradas na identificação de sistemas dinâmicos não-lineares [5][6][7], devido as mesmas serem inerentemente modelos caixa-preta não-lineares e também por terem a habilidade de aproximar complexos mapeamentos não-lineares. Este mapeamento não-linear, realizado pelas RNA's, é baseado em medidas de entrada e de saída do processo a ser modelado. As RNA's utilizadas na identificação não-linear preservam esta relação entrada/saída, mas, normalmente, não preservam a estrutura interna do processo identificado.

Muitas das aplicações de RNA's na identificação de sistemas dinâmicos não-lineares ignora o fato de que o ruído irá sempre estar presente se os dados forem coletados de um sistema real. A menos que os efeitos do ruído sejam bem compreendidos e possam ser apropriadamente compensados, modelos incorretos ou polarizados serão encontrados nos procedimentos de modelagem e identificação.

A polarização é um conceito da teoria de estimação bem conhecido e, muitas das vezes, o seu efeito indesejável pode ser eliminado somente com o uso e identificação de um modelo do ruído. Se o sistema a ser modelado é não-linear, não existe razão para assumir que o ruído irá ser puramente linear.

A polarização é difícil de detectar, isto ocorre porque mesmo que um modelo neural polarizado seja obtido o mesmo irá apresentar uma boa previsão para o conjunto de dados utilizados no treinamento. Uma maneira de evitar a polarização é estimar o modelo do ruído. Esta estimação pode ser feita através da inclusão do erro de previsão no conjunto de sinais de entrada da rede neural [3]. Quando o modelo for utilizado após o

procedimento de identificação paramétrica do mesmo, o erro de previsão é substituído por uma sequência de ruído branco. O ruído branco é utilizado porque no procedimento de identificação paramétrica, esta é a suposição utilizada.

Para determinar a qualidade do modelo, face a presença de ruído nas medidas do processo, utiliza-se testes de validação. Estes testes de validação do modelo são procedimentos criados para avaliar a inadequabilidade de um modelo, independentemente das discrepâncias encontradas nos modelos neurais. Aqui será usado um conjunto de condições desenvolvido por Billings que faz medidas de funções correlações cruzadas como teste de validação do modelo neural [3][5]. Estas medidas de correlações são utilizadas em situações onde o ruído inerente ao processo se apresenta de forma não-linear e primordialmente este procedimento de validação de modelos é utilizado para validar modelos não-lineares.

4. Exemplos

Nesta seção, os exemplos simulados servem para mostrar a utilidade do modelo de RNA da seção (2) na tarefa de identificar sistemas dinâmicos que apresentem ruído de medida. Os exemplos consideram ruído colorido influenciando linearmente a saída do processo, ou ruído branco influenciando não-linearmente a saída do processo. O procedimento de identificação do modelo segue o padrão determinado na figura a seguir, onde $\varepsilon(k)$ é o sinal de erro de previsão, ou resíduo.

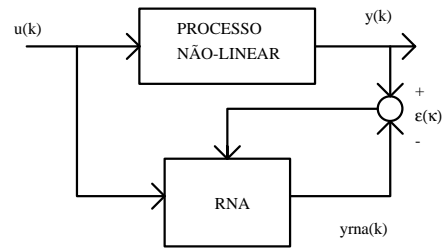


Figura 3 - Identificação neural série-paralelo

O sistema dinâmico não-linear estudado é representado pelas equações abaixo.

$$\begin{aligned} x_1(k+1) &= -0.7x_2(k) + x_3(k) \\ x_2(k+1) &= \tanh[0.3x_1(k) + x_3(k) + (1 + 0.3x_2(k))u(k)] \\ x_3(k+1) &= \tanh[-0.8x_1(k) + 0.6x_2(k) + 0.2x_2(k)x_3(k)] \end{aligned} \quad (4)$$

$$y(k) = [x_1(k)]^2 + d(k) \quad (5)$$

onde k é o instante de amostragem, $x(k)$ é o vetor de estados, $u(k)$ é o sinal de entrada, $y(k)$ é o sinal de saída e $d(k)$ é a sequência de ruído. O conjunto de treino é formado por $N=50$ amostras coletadas para um sinal de entrada do tipo senoidal com mostrado abaixo.

$$u(k) = \text{sen}\left(\frac{2\pi k}{N}\right) \quad (6)$$

4.1. Utilizando Ruído Colorido

Nestas simulações utiliza-se o sinal de erro $d(t)$, da equação (5), como sendo um ruído colorido do tipo, $d(k) = e(k) + 0.6e(k-1)$, com $e(k)$ sendo uma sequência ruído branco de média zero e variância 0.001.

4.1.1. RNA sem Erro de Predição

A RNA utilizada é formada por 4 camadas de neurônios, onde :

- a primeira camada apresenta um único neurônio com função ativação identidade, significando um neurônio estático, e função de saída do tipo tangente hiperbólica.
- a segunda camada tem seis neurônios com função ativação do tipo equação a diferenças de primeira ordem, onde a entrada do neurônio é afetada pelos estados dos neurônios vizinhos mais próximos conforme equação (7), que equivale a matriz A mostrada na equação (8), e função de saída do tipo tangente hiperbólica,.

$$x_i(k+1) = a_{i-1i}x_{i-1}(k) + a_{ii}x_i(k) + a_{i+1i}x_{i+1}(k) + net_i(k) \quad (7)$$

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & 0 & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45} & 0 \\ 0 & 0 & 0 & a_{54} & a_{55} & a_{56} \\ 0 & 0 & 0 & 0 & a_{65} & a_{66} \end{bmatrix} \quad (8)$$

- a terceira camada contém doze neurônios com função ativação identidade, significando neurônios estáticos, e função de saída do tipo tangente hiperbólica.
- a última camada tem um único neurônio com função ativação identidade, significando um neurônio estático, e função de saída do tipo tangente hiperbólica.

A identificação se deu em duas etapas : uma sessão de treinamento, onde a RNA teve os seus pesos ajustados em função dos dados coletados, e outra sessão onde foi calculada a estimativa das funções correlações amostradas. Nesta duas etapas utiliza-se o sinal $y(k)$ normalizado. Os resultados são vistos na figura (4), que mostra o comportamento da RNA no início do treinamento, na figura (5), que mostra o comportamento da RNA após 18350 iterações de treinamento, e na figura (6), que mostra as medidas de correlações cruzadas.

O modelo será adequado se todas as medidas de correlação ficarem dentro do limite imposto pela linha tracejada, que é a banda de confiança utilizada [3]. Este limiar é determinado a partir de propriedades estatísticas sempre encontradas em procedimentos de identificação de processos. Como observa-se da figura (6), o modelo neural encontrado apresenta algumas deficiências, indicado pelas curvas Phi3 e Phi5.

A topologia da RNA usada foi a que mais de adequou ao problema em questão. Foram testadas dezenas de topologias antes que se chegasse a esta utilizada.

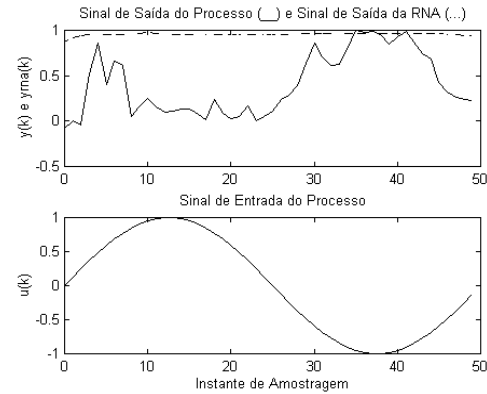


Figura 4 - RNA no início do treinamento

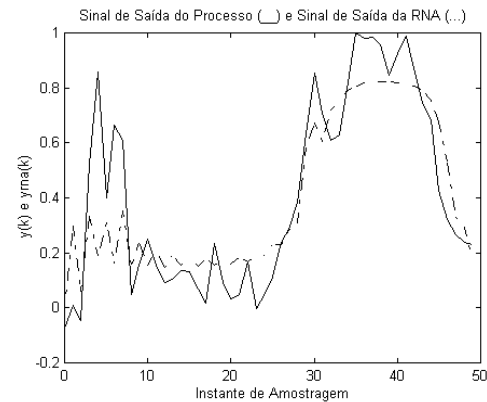


Figura 5 - RNA após o treinamento

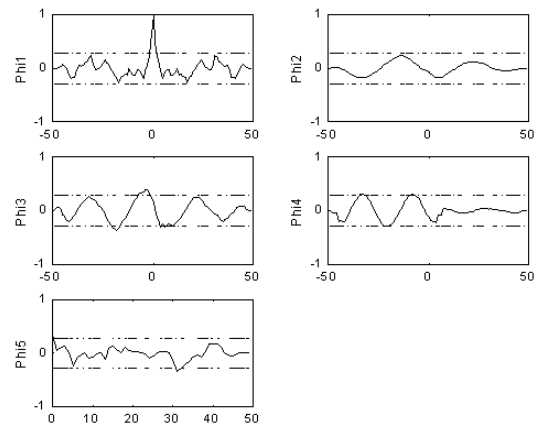


Figura 6 - Funções Correlações Cruzadas : $\Phi_1 = \Phi_{ee}(t)$; $\Phi_2 = \Phi_{ue}(t)$; $\Phi_3 = \Phi_{u^2e}(t)$; $\Phi_4 = \Phi_{u^2e^2}(t)$; $\Phi_5 = \Phi_{e(eu)}(t)$

4.1.2. RNA com Erro de Predição

Neste caso utiliza-se o sinal de erro de predição $\varepsilon(k)$ como um outro sinal de entrada da rede neural. A RNA utilizada é formada por 4 camadas de neurônios, onde :

- a primeira camada apresenta dois neurônios com função ativação identidade e função de saída do tipo tangente hiperbólica (um neurônio recebe o sinal $u(k)$ e o outro neurônio recebe o sinal $\varepsilon(k)$).

- a segunda camada tem três neurônios com função ativação do tipo equação a diferenças de primeira ordem, onde a entrada do neurônio é afetada pelos estados de todos os outros neurônios conforme equação (9), que equivale a matriz A mostrada na equação (10), e função de saída do tipo tangente hiperbólica.

$$x_i(k+1) = a_{i1}x_1(k) + a_{i2}x_2(k) + \dots + a_{ii}x_i(k) + \dots + a_{in}x_n(k) + net_i(k) \quad (9)$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (10)$$

- a terceira camada contém doze neurônios com função ativação identidade e função de saída do tipo tangente hiperbólica.

- a última camada tem um único neurônio com função ativação identidade e função de saída do tipo tangente hiperbólica.

Como no caso anterior, a identificação se deu em duas etapas: uma sessão de treinamento, e outra sessão onde foi calculada a estimativa das funções correlações amostradas. Também aqui, o sinal $y(k)$ utilizado está normalizado. Os resultados são vistos nas figuras (7), (8) e (9). A figura (8) é o resultado de 154886 iterações de treinamento.

Como observa-se da figura (9), consegue-se um modelo adequado para o processo identificado. A topologia da RNA utilizada, também foi resultado de várias tentativas na busca de uma topologia que apresentasse bons resultados. O número de iterações de aprendizado, necessário para que se chegasse a situação mostrada, é maior do que o caso anterior, mas isto não é importante porque o treinamento foi feito *off-line*, com os dados coletados sendo apresentados exaustivamente à RNA, até que a mesma apresente um desempenho adequado.

O número de iterações, para este caso, é aproximadamente oito vezes maior do que o caso anterior (sem o uso do sinal de erro de predição), mas este número não é levado em conta como critério de parada da sessão de treinamento. O que é considerado para interromper o treinamento é o conjunto de medidas das funções correlações cruzadas. Este cálculo é feito após um certo número de iterações, pré-definido, e se as funções correlações não satisfizerem a banda de confiança, o treinamento continua. A causa desta discrepância entre o número de iterações nestes dois casos é a inicialização dos pesos da rede neural. Em outras palavras, em cada uma das situações, o conjunto de pesos iniciais do modelo neural é diferente.

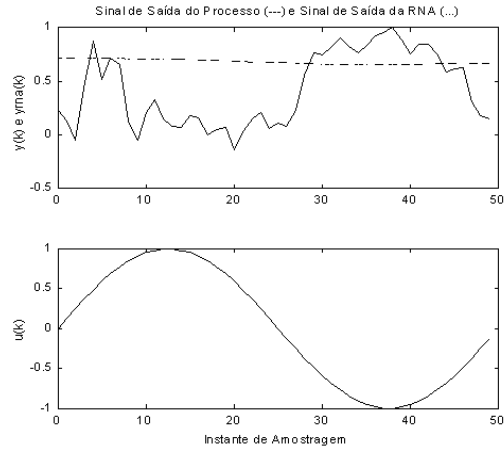


Figura 7 - RNA no início do treinamento

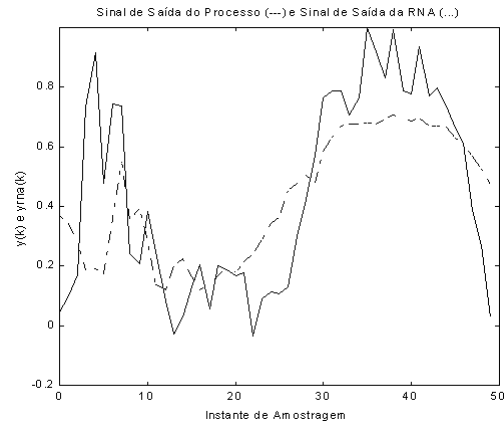


Figura 8 - RNA após o treinamento

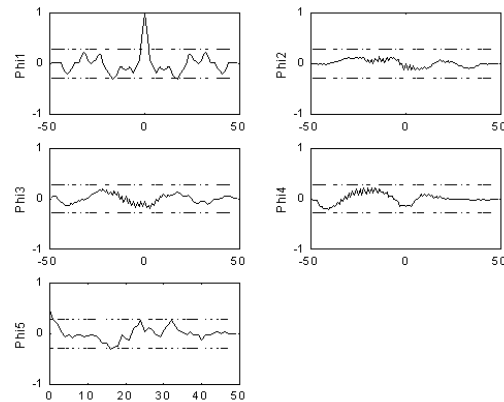


Figura 9 - Funções Correlações Cruzadas : $\Phi_{ee}(t)$; $\Phi_{ue}(t)$; $\Phi_{u^2e}(t)$; $\Phi_{ue^2}(t)$; $\Phi_{e(eu)}(t)$

4.2. Ruído Branco Não-Linear

Nestas simulações o ruído de medida $d(k)$ influencia a saída do processo de uma forma não-linear, ou seja,

$$y(k) = [x_1(k) + d(k)]^2 \quad (11)$$

com $d(k) = e(k)$. $e(k)$ é uma sequência ruído branco de média zero e variância 0.001.

4.2.1. RNA sem Erro de Predição

A RNA utilizada é idêntica aquela utilizada na seção (4.1.1).

Como nos casos anteriores, a identificação se dá em duas etapas : uma de treinamento e outra de cálculo das funções correlações (sinal $y(k)$ normalizado). Os resultados são vistos nas figuras (10), (11) e (12). A figura (11) é o resultado de 246 iterações de treinamento.

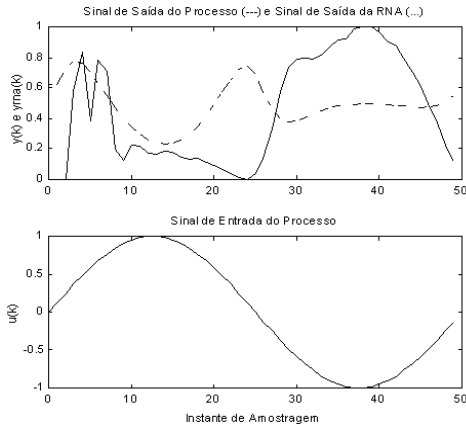


Figura 10 - RNA no início do treinamento

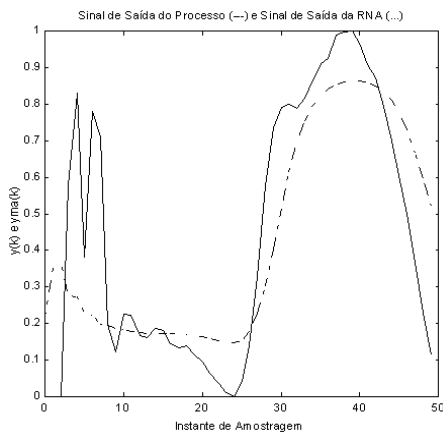


Figura 11 - RNA após o treinamento

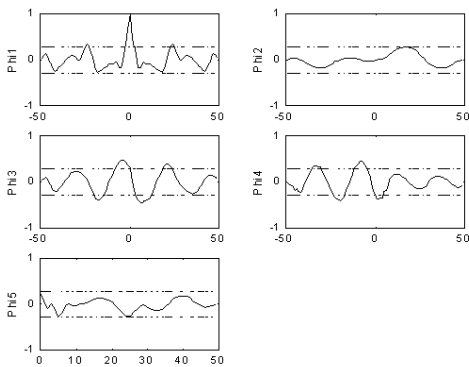


Figura 12 - Funções Correlações Cruzadas : $\Phi_{ee}(t)$; $\Phi_{ue}(t)$; $\Phi_{u'e'}(t)$; $\Phi_{u''e''}(t)$; $\Phi_{e(eu)}(t)$

Como observa-se da figura (12), o modelo neural encontrado apresenta algumas deficiências, indicado

pelas curvas Phi1, Phi3 e Phi4. Como nos casos anteriores, a topologia da RNA usada foi a que mais de adequou ao problema em questão.

4.2.2. RNA com Erro de Predição

A RNA utilizada é idêntica aquela utilizada na seção (4.1.2).

Como nos casos anteriores, a identificação se dá em duas etapas : uma de treinamento e outra de cálculo das funções correlações (sinal $y(k)$ normalizado). Os resultados são vistos nas figuras (13), (14) e (15). A figura (14) é o resultado de 249 iterações de treinamento.

Como observa-se da figura (15), consegue-se um modelo adequado para o processo identificado. A topologia da RNA utilizada, também foi resultado de várias tentativas na busca de uma topologia que apresentasse bons resultados. O número de iterações de aprendizado, necessário para que se chegasse a situação mostrada, é bem próximo do que o caso anterior sem o erro de predição.

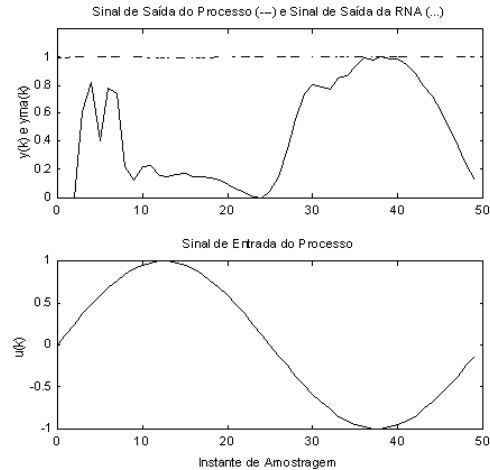


Figura 13 - RNA no início do treinamento

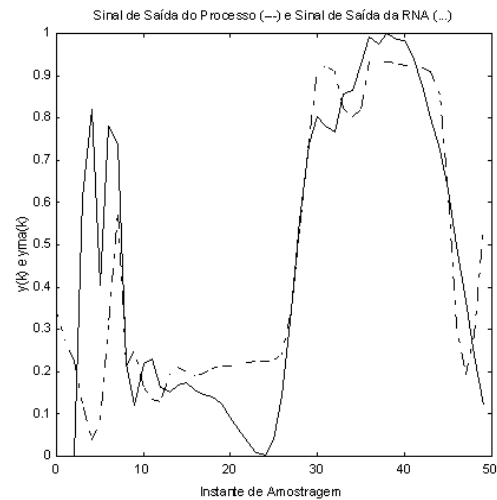


Figura 14 - RNA após o treinamento

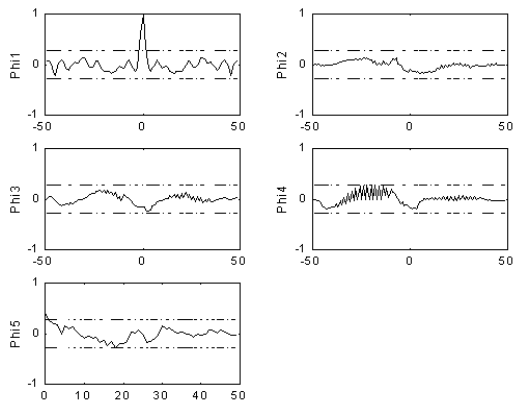


Figura 15 - Funções Correlações Cruzadas : $\text{Phi1} = \Phi_{ee}(t)$; $\text{Phi2} = \Phi_{ue}(t)$; $\text{Phi3} = \Phi_{u^2e}(t)$; $\text{Phi4} = \Phi_{u^2e^2}(t)$; $\text{Phi5} = \Phi_{e(eu)}(t)$

5. Discussões

Este trabalho mostra a utilidade do modelo de RNA, formada por neurônios dinâmicos lineares e por neurônios estáticos não-lineares, na tarefa de identificação de processos não-lineares. O modelo neural utilizado conserva apenas a relação não-linear entre o sinal de entrada e o sinal de saída do processo, não preservando a estrutura interna do mesmo, como a imensa maioria dos modelos neurais existentes na literatura.

Com a presença de ruído de medida colorido ou não-linear, corrompendo o sinal de saída do processo, a RNA só apresenta um desempenho aceitável quando incorpora o erro de predição como um sinal de entrada. A influência desta incorporação é significativa para o bom desempenho do modelo neural; sem o mesmo, a RNA apresenta deficiências na sua modelagem. O comportamento da saída da rede neural não acompanha o sinal de saída do processo porque este está corrompido pelo ruído. Ou seja, o modelo não deve ser guiado pela informação do ruído de medida do processo.

Referências

- [1] L. Ljung. System Identification : Theory for the user. Prentice-Hall, Englewood Cliffs, NJ, 1987.
- [2] S. A. Billings. Identification of Nonlinear Systems - A Survey. Proc. IEE, Pt. D, 127(6):272-284, 1980.
- [3] S. A. Billings, H. B. Jamaluddin, and S. Chen. Properties of Neural Networks with Applications to Modelling Nonlinear Dynamical Systems. Int. J. Control, 55(1):193-224, 1992.
- [4] U. Levin and K. S. Narendra. Control of Nonlinear Dynamical Systems using Neural Networks - Observability, Identification and Control. IEEE Trans. on Neural Networks, 7(1):30-42, 1996.
- [5] R. C. L. de Oliveira, F. M. de Azevedo, e J. M. Barreto. Rede Neural com Dinâmica Linear Representada no Espaço de Estados. Anais do III Congresso Brasileiro de Redes Neurais, Florianópolis, SC, páginas 115-120, 1997.

- [6] F. M. De Azevedo. Uma Proposta de Modelos Formais de Neurônios e Redes Neurais. Anais do III Congresso Brasileiro de Redes Neurais, Florianópolis, SC, páginas 140-145, 1997.
- [7] A. C. Tsoi and A. D. Back. Locally Recurrent Globally Feedforward Networks : A Critical Review of Architectures. IEEE Transactions on Neural Networks, 5(2):229-239, 1994.