Recurrent neural networks for identi¯cation of nonlinear systems ¤

Xuemei Ren
Department of Automatic Control, Beijing Institute of Technology
Beijing,100081, P.R.China (Email: xuemeiren @263.net)

Shumin Fei
Research Institue of Automation, Southeast University
Nanjing 210096, P R China (Email: infcon@seu.edu.cn)

Abstract

A new type of recurrent neural network is discussed in this paper, which provides the potential for the modelling of unknown nonlinear systems with multi-inputs and multi-outputs. The proposed network is a generalization of the network described by Elman. It is shown that the proposed network with appropriate neurons in the context layer can model unknown nonlinear systems. Based on the PID-like training objective function, the learning algorithm of the proposed network is considerably faster through the introduction of dynamic backpropagation, which is used to estimate the weights of both the feedforward and feedback connections. The techniques have been successfully applied to the modelling nonlinear plants and simulation results are included.

1 Introduction

The feedforward neural networks have been widely applied to dynamic system identi¯cation in recent years (Billings et al: 1992, Narendra and Parthasarathy 1990, Reynold and Tenorio 1990). However, the feedforward network can not represent by itself a dynamic system without the help of tapped delay lines since it is a static mapping. The drawbacks associated with the use of feedforward networks for system identi¯cation are a long computation time, easily being a®ected by external noises and di± culty in obtaining an independent system simulator (Pham and Liu 1992).

Recurrent neural networks have been paid more attention in the identi¯cation of dynamic systems since they have a dynamic memory not found in feedforward networks, especially their ability in modelling nonlinear systems without requiring external feedback lines. Recently, various recurrent networks have been o®ered to remedy the limitations of the feedforward networks. Werbos (1988) and Bass (1990) utilized the feedback of past values of all system outputs and certain auxiliary variables to represent a class of nonlinear systems. Ku and Kwang (1995) proposed a diagonal recurrent neural network with the hidden layer consisting of self-recurrent neurons for the identi¯cation and control of dynamic nonlinear systems. The Elman network (Elman 1990) is an important recurrent network, where the outputs of neurons in the hidden layer at the previous time step are fed back to the context neurons. Scott and Ray (1993) demonstrated the performance of the Elman network for nonlinear process modelling. Pham and Karaboga (1999) developed a modi¯ed version of the Elman network to facilitate its application in dynamic system identi¯cation.

This paper proposes a new type of recurrent network for system identi¯cation. The proposed network is an extended version of the Elman network. It is shown that the modi¯ed Elman neural network (MENN) is capable of modelling unknown nonlinear systems. In the training of the MENN, a PID-like training objective function is de¯ned, and a new dynamic backpropagation algorithm is proposed to achieve a faster learning.

This paper is organized as follows. Section 2 describes the proposed MENN. The input-output dynamics of the MENN is analysed in Section 3. A dynamic backpropagation learning algorithm on the basis of the PID-like training objective function is proposed in Section 4 for modelling nonlinear systems. Applications of the MENN to the identi¯cation of nonlinear systems are presented in Section 5. Finally, Section 6 concludes this paper.

2 The modi¯ed Elman network

The Elman network has been veri¯ed well suited

¤This project supported by the National Natural Science Foundation of China(69804001, 69604003)

for dynamic system identification. The Elman network has three layers including the input layer, the hidden layer and the output layer. The input layer is composed of the two different groups of neurons, that is, the group of external input neurons and the group of the internal neurons (also called context units). Pham and Liu (1992) investigated the approximation property of the Elman network. It is shown that the Elman network with all feedback connections from the hidden layer to the context layer set to 1 can represent an arbitrary linear system if the linear activation functions are adopted for the hidden neurons.

With the objective of adding dynamic memory capability of the Elman network, we propose a modified Elman network, shown in Figure 1, where arbitrary connections can be allowed from the hidden layer to the context layer. Activation function in the nonlinear hidden layer can be chosen as the hyperbolic tangent function

$$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}};$$

while all neurons in the other layers have linear activation functions. Let $u(k) \in R^p$ and $\hat{y}(k) \in R^q$

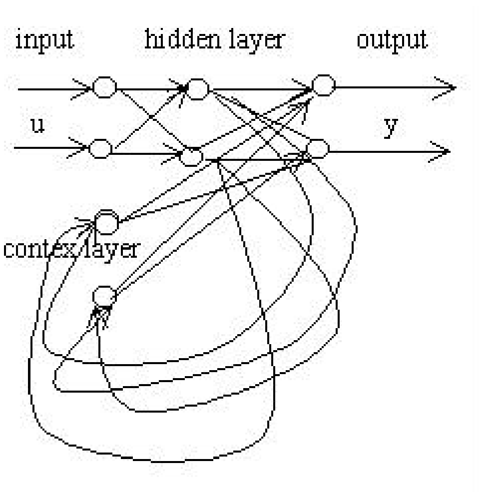

Figure 1: The proposed MENN.

denote the network input and output vectors at the discrete time k, respectively. Let $c(k) \in R^s$ be the output of the context layer and $h(k) \in R^l$, the output of the hidden layer. Therefore, the following relations describing the MENN can be obtained

$$\begin{aligned} \hat{y}(k+1) &= w^o h(k) \\ h(k) &= \sigma(w^c c(k) + w^I u(k)) \\ c(k) &= w^h h(k-1) \end{aligned} \quad (1)$$

where $w^o \in R^{q \times l}; w^c \in R^{l \times s}; w^I \in R^{l \times p}; w^h \in R^{s \times l}$ are respectively weight matrices. The function $\sigma : R^l \to R^l$ is denoted by

$$\sigma((x_1; \cdots; x_l)^T) = (\sigma(x_1); \cdots; \sigma(x_l))^T:$$

From Figure 1, one can see that the MENN can reduce to the Elman network if its feedback matrix w^h is unitary matrix. Therefore, the MENN has more degrees of freedom to represent dynamic systems. In the sense of memory capability, the MENN has more space than the Elman network.

3 Input-output dynamics of the MENN

From (1), the relationship between the network output $\hat{y}(k)$ and state c(k) is derived by

$$\begin{aligned} \hat{y}(k+1) &= w^o \sigma[w^c c(k) + w^I u(k)] \\ c(k) &= w^h \sigma[w^c c(k-1) + w^I u(k-1)] \end{aligned} \quad (2)$$

The above equation can be rewritten as

$$\begin{aligned} \hat{y}(k+1) &= \psi[c(k); u(k)] \\ c(k) &= \phi[c(k-1); u(k-1)] \end{aligned} \quad (3)$$

where the mappings $\psi : R^{s+p} \to R^q; \phi : R^{s+p} \to R^s$ are nonlinear smooth functions. Since $\phi(0; 0) = 0$, the original is an equilibrium state of (2). From (3),we have

$$\begin{aligned} \hat{y}(k+1) &= \psi[c(k); u(k)] = \psi_1[c(k); u(k)] \\ \hat{y}(k+2) &= \psi[\phi(c(k); u(k)); u(k+1)] \\ &= \psi_2[c(k); u(k); u(k+1)] \\ &\cdots \quad \cdots \\ \hat{y}(k+s) &= \psi \cdot \phi^{s-1}[:::] \\ &= \psi_s[c(k); u(k); \cdots; u(k+s-1)] \end{aligned} \quad (4)$$

where $\phi^{s-1}[:::]$ is an $(s-1)$-times iterated composition of ϕ. Denoting $\hat{Y}_s(k) = [\hat{y}(k+1); \cdots; \hat{y}(k+s)]^T$ and $U_s(k) = [u(k); u(k+1); \cdots; u(k+s-1)]^T$, (4) is of the form

$$\hat{Y}_s(k) = \Psi[c(k); U_s(k)] \quad (5)$$

where $\Psi = [\psi_1; \cdots; \psi_m]^T$.If the Jacobian of Ψ with respect to c is nonsingular at $c = 0; U_s = 0$, by using the implicit function theorem that there exists a unique local solution of (5) as follows

$$c(k) = \xi[\hat{Y}_s(k); U_s(k)] \quad (6)$$

in a neighborhood of the equilibrium state $c = 0; U_s = 0$,where $\xi : R^{s(p+q)} \to R^s$ is a smooth function .By (3),the state $c(k+s)$ depends on the state c(k) and the input sequence $U_s(k) = [u(k); u(k+1); \cdots; u(k+s-1)]^T$. Since $\hat{y}(k+s) = \psi(c(k+s-1); u(k+s-1))$,this leads to the input-output representation of the network given by

$$\begin{aligned} \hat{y}(k+1) = \; & G[\hat{y}(k); \cdots; \hat{y}(k-s+1); \\ & u(k); \cdots; u(k-s+1)] \end{aligned} \quad (7)$$

where $G : R^{s(p+q)} \rightarrow R^q$ is a smooth function.

4 Network training

Our interest in this paper is the identification using the proposed MENN for the nonlinear dynamic system expressed by

$$y(k+1) = F[y(k); y(k-1); \cdots; y(k-n+1); u(k); u(k-1); \cdots; u(k-m+1)] \quad (8)$$

where $u(k) \in R^p$ and $y(k) \in R^q$ are the control and output ,respectively. The mapping $F : R^{nq+mp} \rightarrow R^q$ is an unknown continuous function. From (7), we can see that the proposed MENN (1) can be used to learn the input-output behavior of dynamic system (8) if the order n of the system (8) is equal to or less than the number s of unites in the context layer. In many cases, we do not know the order of the plant, the number of the context units may be set sufficiently large enough to meet this condition.

To model an unknown plant, the network only requires a sequence of input signals $u(k)$ and the corresponding plant output signals $y(k+1)$ at the discrete time k. The purpose of the weight learning of the MENN is to estimate the weight such that the output $\hat{y}(k)$ of the network convergences to the plant output $y(k)$ as $k \rightarrow \infty$. As a matter of fact, a simple and natural extension of the static backpropagation algorithm is dynamic backpropagation for recurrent neural networks. A new dynamic backpropagation learning version will be derived in this section.

Based on the idea of PID control, we define the following PID-like training objective function as

$$J = \frac{1}{2}\sum_{k=0}^{N_1} e^T(k+1)e(k+1) + \gamma e_1^T(k+1)e_1(k+1) + \delta e_2^T(k+1)e_2(k+1) \quad (9)$$

such that the convergence of the weights of the MENN can be speeded up, where N_1 represents the total number of data patterns used in the training process. $e(k+1) = y(k+1) - \hat{y}(k+1)$ is the error between the plant and network outputs at time $k+1$; $e_1(k+1) = e(k+1) - e(k)$; $e_2(k+1) = e_1(k+1) - e_1(k)$; $0 \leq \gamma < 1$ and $0 \leq \delta < 1$ are weighting factors. Using the backpropagation algorithm, the weights of the MENN are updated as follows

$$w_{i+1} = w_i - \eta_i \frac{\partial J}{\partial w_i} + \alpha \Delta w_i \quad (10)$$

where i is the iteration number, η_i is a learning rate, α is a momentum factor, Δw_i represents the change in weight in ith iteration. Here, we use an adapted value of η_i as follows:

$$\eta_{i+1} = \begin{cases} 1.05\eta_i & \text{if } J_i \leq 0.99 J_{i-1} \\ 0.85\eta_i & \text{if } J_i \geq 1.02 J_{i-1} \\ \eta_i & \text{otherwise.} \end{cases} \quad (11)$$

η_0 is chosen to lie between 0 and 1. From (10), we can obtain

$$\frac{\partial J}{\partial w_i} = \sum_{k=0}^{N_1} e^T(k+1)\frac{\partial \hat{y}(k+1)}{\partial w_i} + \gamma e_1^T(k+1)\frac{\partial \hat{y}(k+1)}{\partial w_i} + \delta e_2^T(k+1)\frac{\partial \hat{y}(k+1)}{\partial w_i}. \quad (12)$$

In order to compute the gradient $\frac{\partial \hat{y}(k+1)}{\partial w}$, the mathematical model (1) for the MENN can be rewritten as

$$\begin{aligned} \hat{y}_i(k+1) &= \sum_j w_{ij}^o h_j(k) \\ h_j(k) &= \sigma(s_j(k)) \\ s_j(k) &= \sum_r w_{jr}^c c_r(k) + \sum_d w_{jd}^I u_d(k) \\ c_r(k) &= \sum_j w_{rj}^h h_j(k-1). \end{aligned} \quad (13)$$

The output gradients with respect to $w^o; w^c; w^I$ and w^h are given by

$$\frac{\partial \hat{y}_i(k+1)}{\partial w_{ij}^o} = h_j(k) \quad (14)$$

$$\frac{\partial \hat{y}_i(k+1)}{\partial w_{jr}^c} = w_{ij}^o P_{w_{jr}^c}; \quad P_{w_{jr}^c} = \frac{\partial h_j(k)}{\partial w_{jr}^c} \quad (15)$$

$$\frac{\partial \hat{y}_i(k+1)}{\partial w_{jd}^I} = w_{ij}^o P_{w_{jd}^I}; \quad P_{w_{jd}^I} = \frac{\partial h_j(k)}{\partial w_{jd}^I} \quad (16)$$

$$\frac{\partial \hat{y}_i(k+1)}{\partial w_{rj}^h} = w_{ij}^o P_{w_{rj}^h}; \quad P_{w_{rj}^h} = \frac{\partial h_j(k)}{\partial w_{rj}^h} \quad (17)$$

where $P_{w_{jr}^c}; P_{w_{jd}^I}; P_{w_{rj}^h}$ satisfy

$$P_{w_{jr}^c}(k) = \sigma'(s_j)[c_r(k) + w_{jr}^c w_{rj}^h P_{w_{jr}^c}(k-1)]; \quad P_{w_{jr}^c}(0) = 0 \quad (18)$$

$$P_{w_{jd}^I}(k) = \sigma'(s_j)[u_d(k) + w_{jr}^c w_{rj}^h P_{w_{jd}^I}(k-1)]; \quad P_{w_{jd}^I}(0) = 0 \quad (19)$$

$$P_{w_{rj}^h}(k) = \sigma'(s_j)[w_{jr}^c(k)h_j(k-1) + w_{jr}^c w_{rj}^h P_{w_{rj}^h}(k-1)]; \quad P_{w_{rj}^h}(0) = 0 \quad (20)$$

It is shown that if the weight w^h is zero, the MENN (1) becomes the usual feedforward neural network and dynamic recurrent equations (18), (19) and (20) are algebraic.

5 Simulation

In this section, the ability of the MENN to be trained by dynamic backpropagation algorithm to model nonlinear dynamic systems has been demonstrated in simulation. The initial weights are randomly set between (-0.5,0.5) for the purpose of avoiding weight paralysis and speeding up the convergence of the networks. The parameters of the networks are adjusted by (10)-(20). The input sequence is random in the range [-1,1]. To test the performance of the trained network, the network is used in recall mode to produce responses to a sinusoidal input and random input signal in the range [-0.5,0.5]. The performance of the trained network is measured by

$$MSE = \frac{1}{N_2} \sum_{k=1}^{N_2} [y(k) - \hat{y}(k)]^2$$

where N_2 represents the number of data pairs $[u; y]$ in the recall set.

The modelling is carried out using the Elman network and the MENN. For demonstration, the same number of units in the context layer for two types of networks is used.

Example 1 : The plant is described by

$$y(k+1) = \frac{y(k)}{1.2 + y^2(k)} + 0.5u(k) \tag{21}$$

During training, the parameters of the network are given in Table 1,where $\eta_0; \alpha; \gamma; \delta;$ and N_1 ,respectively, denote the initial value of learning rate, momentum term, error derivative factor ,error integral factor, total number of data patterns used in the training process, and i represents the number of iteration when the batch-based training method is used. A random signal in the range [-1,1] is used as the input to the plant, the Elman network and the MENN. In the recall models, responses from the plant and the MENN are obtained for two kinds of signals: $u(k) = 0.62\sin(2\pi/25) + 0.5\sin(2\pi/40)$ and a random input $-0.5 < u(k) < 0.5; k = 1, 2, \cdots, 150$. The simulation results are shown in Figures 2 and 3Ä* * y,{ ŷℭ .

For comparison, the MSE values computed for the Elman network(EN) and the MENN are presented in Table 2. In this example, both types of networks capture the dynamics of the given system. However, the MENN gives a better performance.

Parameters						
Example	η_0	α	γ	δ	N_1	i
1	0.001	0.95	3.2	1.4	200	2000
2	0.001	0.55	1.6	0.4	300	4000
3	0.003	0.65	1.9	0.7	300	6000

Table 1. The training parameters of the MENN.

Example (Input)	MSE(EN)	MSE(MENN)	
1 (Sinusoidal input)	0.000611	0.000575	
1 (Random input)	0.000401	0.000273	
2 (Sinusoidal input)	0.0053	0.0025	
2 (Random input)	0.000909	0.000756	
3 (Sinusoidal input)	0.1041	0.0423	
3 (Random input)	0.0233	0.0040	

Table 2. Performance indices for different examples.

Example 2: The plant has the following

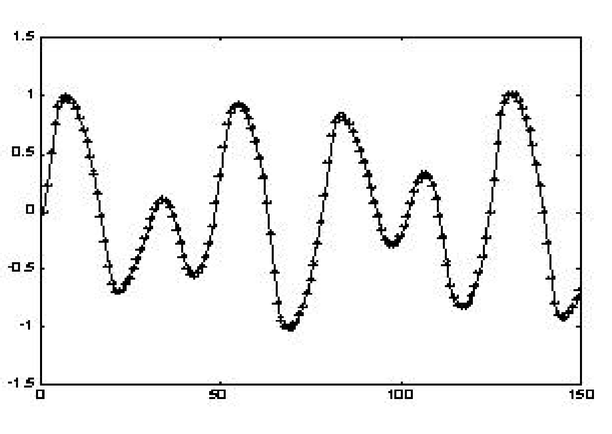

Figure 2: Sinusoidal responses of plant (21) and the MENN.

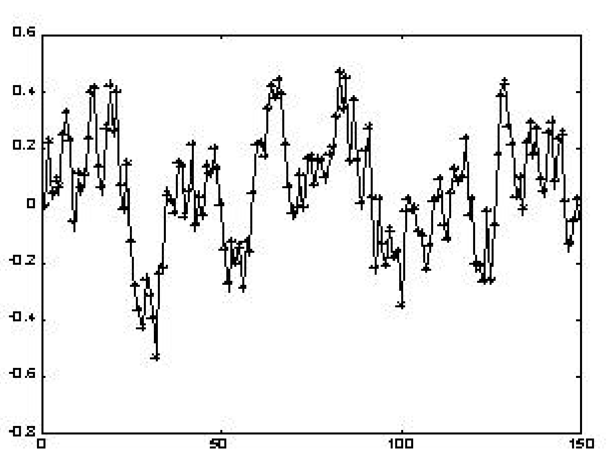

Figure 3: Random responses of plant (21) and the MENN.

equation of the form

$$y(k+1) = 1.532y(k) - 0.845y(k-1) + 0.06u(k) + 0.01u^2(k)/(1 + y^2(k)) \tag{22}$$

The training parameters for the MENN are given in Table 1. A random input signal between +1 and -1 (the same as in Example 1) is applied as the training signal. Figures 4 and 5 show the responses of the MENN during recall for the case where the input was, respectively, a sinusoidal function $u(k) = 0.3\sin(2\pi/15) + 0.5\sin(2\pi/25) + 0.23\sin(2\pi k/50)$ and a random input in the range [-0.5,0.5]. Table 2 gives the MSE performance indices of the Elman network and the MENN.

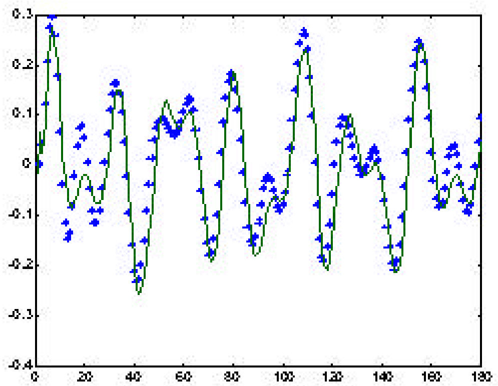

Figure 4: Sinusoidal responses of plant (22) and the MENN.

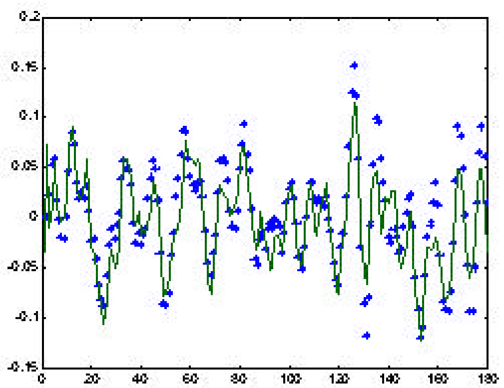

Figure 5: Random responses of plant (22) and the MENN.

Example 3: The system can be found in the paper by Sales and Billings (1990) and is described by

$$\begin{aligned} y(k+1) &= 0.9722y(k) + 0.3578u(k) - 0.1295u(k-1) \\ &- 0.3103y(k)u(k) - 0.04228y^2(k-1) \\ &+ 0.1663y(k-1)u(k-1) + 0.2573y(k-1)e(k) \\ &- 0.03259y^2(k)y(k-1) - 0.3513y^2(k)u(k-1) \\ &+ 0.3084y(k)y(k-1)u(k-1) + 0.2939y^2(k-1)e(k) \\ &+ 0.1087y(k-1)u(k)u(k-1) + 0.4770y(k-1)u(k)e(k) \\ &+ 0.6389u^2(k-1)e(k) + e(k+1) \end{aligned} \tag{23}$$

The model used in the experiment was identified from a laboratory scale liquid level system. The system consists of a DC water pump feeding a conical flask which in turn feeds a square tank, giving the system second-order dynamics. The input is the voltage to the pump motor and the output of the system is the height of water in the conical flask. The noise sequence $e(k)$ is gaussian and its variance is 0.05. The training parameters for the MENN are shown in Table 1. The training conditions are the same for Example 2. The results obtained are demonstrated in Figures 6 and 7. MSE performance indices for the Elman network and the MENN are given in Table 2. These results show the effectiveness of the proposed scheme in modelling a practical system.

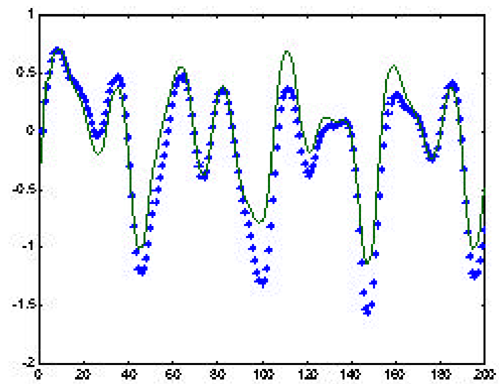

Figure 6: Sinusoidal responses of plant (23) and the MENN.

6 Conclusion

This paper investigates the use of dynamic backpropagation to train the proposed MENN for the identification of dynamic systems. It is shown that the MENN can model nonlinear system under the assumption that the order of the system is equal to or less than the number of the context units in the MENN. The dynamic backpropagation algorithm based on the PID-like training objective function can

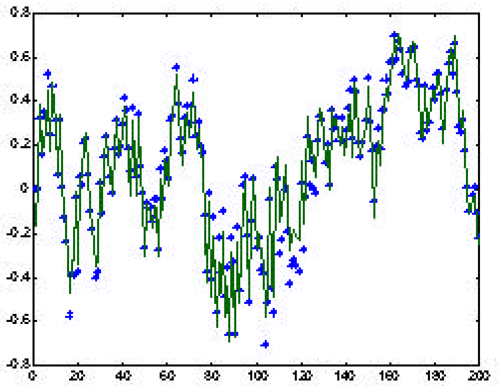

Figure 7: Random responses of plant (23) and the MENN.

be speeded up the convergence of the MENN. The identi¯cation results obtained for di®erent systems have demonstrated the theory. We also conclude that the MENN has a better modelling performance than the Elman network.

References

[1] Bass, R.W.,1990, Optimal recurrent backpropagation networks for real-time identi¯cation. Proceedings of the 29th IEEE Conference on Decision and Control, pp.2756-2765.

[2] Billings, S. A., Jamaluddin, H. B.,and Chen,S.,1992, Properties of neural networks with applications to modelling nonlinear systems.International Journal of Control,55,193-224.

[3] Elman, J. L.,1990, Finding structure in time.Cognitive Science,14,179-211.

[4] Funahashi, K.,1989, On the approximate realization of continuous mappings by neural networks. Neural Networks, 2, 183-192.

[5] Ku, C. C., and Kwang, Y. L.,1995, Diagonal recurrent neural networks for dynamic systems control. IEEE Trans. on Neural Networks,6,(1),144-155.

[6] Narendra, K. S.,and Parthasarathy, K.,1990, Identi¯cation and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks,1,4-27.

[7] Pham ,D. T., and Liu, X.,1992, Dynamic system identi¯cation using partially recurrent neural networks. Journal of Systems Engineering, 2,(2),90-97.

[8] Pham, D. T., and Karaboga, D.,1999, Training Elman and Jordan networks for system identi¯cation using genetic algorithms. Arti¯cial Intelligence in Engineering, 13,107-117.

[9] Sales, K. R., and Billings, S. A.,1990, Self-tuning control of nonlinear ARMAX models. Int. J. Control,51,(4),753-769.

[10] Scott, G. M., and Ray, W. H.,1993, Creating e± cient nonlinear network process models that allow model interpretation.J Process Control,3,(3),163-178.

[11] Werbos,P.J.,1988, Generalization of backpropagation with application to a recurrent gas market model. Neural Networks,1,339-356.

[12] Williams, R.,and Zipser, D.,1989, A learning algorithm for continually running fully recurrent neural networks. Neural Computation,1,270-280.