NONLINEAR SYSTEM IDENTIFICATION USING NEURAL NETWORKS

P.P.Mohanlal* Rekha Sebastian ** S.Dasgupta*

*Control and Guidance Design Group, VSSC , Trivandrum, India.
** Lecturer, The Indian Engineering College, Vadakankulam, Tamil Nadu, India.

Abstract: Non-linear system identification is essential for the indirect adaptive control of such systems. Feed forward neural networks are general non-linear static maps. They can approximate input-output maps of non-linear dynamic systems using static back propagation algorithm, if the outputs of the target dynamic systems are observable. Static back propagation , Dynamic back propagation(DBP) and Adjoint back propagation(ABP) are applied to general non-linear dynamic system identification problems. DBP and ABP have to be used for recurrent structures. Examples are constructed to demonstrate the different methods for identification using neural networks.

1.INTRODUCTION

The standard back propagation algorithm is well known for training feed forward neural networks to deal with static function mapping problems such as pattern recognition. This computationally efficient algorithm is used in many neural identification applications. Neural networks for system identification is widely recognised as useful for identifying non-linear systems with uncertainties. Recurrent neural networks have better representational abilities than feed forward networks. Training of recurrent neural networks, considering it as a non-linear dynamic system is complex and computationally intensive. But, it becomes necessary to use recurrent neural networks to infer internal dynamics of systems when the relation between internal dynamics and measured outputs is not invertible. In this paper, three examples are chosen to demonstrate the applicability of neural networks for non-linear system identification. The first example uses static back propagation, the second uses dynamic back propagation [2] and the third uses the adjoint back propagation [3]. The DBP and ABP are essential for training recurrent neural networks.

2.DYNAMIC BACK PROPAGATION FOR NONLINEAR DYNAMIC SYSTEMS

In a causal dynamic system, a change in a parameter at time, k, will produce a change in the output y(t) for all t>=k.For example, given a non-linear dynamic system,

$x(k+1) = \phi(x(k),u(k),\theta)$; $y(k) = \Psi(x(k))$............(1)

where, θ is a parameter , u is the input and x is a state vector, then the partial derivative of y(k) with respect to θ can be obtained by solving the following linear state equations,

$z(k+1) = A(k)\, z(k) + V(k)$; $z(0) = 0$; $w(k) = C(k)\, z(k)$.

where, $z(k) = \delta x(k) / \delta \theta$; $A(k) = \phi_x(k)$; $V(k) = \phi_\theta(k)$; $w(k) = \delta y(k) / \delta\theta$; $C(k) = \Psi_x(k)$.

ϕ_x and Ψ_x are the Jacobian matrices and the vector ϕ_θ represents the partial derivatives of ϕ with respect to θ . In the above equations, if A(k) , V(k) and C(k) can be computed , w(k) can be obtained as the output of the dynamic sensitivity model which is the linearised version of the original non-linear dynamic system around a nominal trajectory and input. Thus, the required gradient computation becomes possible for recurrent network which is a non-linear dynamic system. If there are M adjustable parameters, this method requires M separate sensitivity networks to be generated to get all the M gradient components of the vector. However, this is a forward method and is not the generalisation of static back propagation.

3.ADJOINT BACK PROPAGATION FOR NON-LINEAR DYNAMIC SYSTEMS [3]

The co-state equations associated with any optimisation problem actually represent the adjoint of the original system. This is also known as the dual of the original system. The co-states are the states of the adjoint system. The adjoint mathematics is advantageously used for sensitivity analysis for tactical missile guidance applications where the forward system is a linear time varying system [4]. Sensitivities are observed from the adjoint system to calculate the control for the forward system. Viewing the static back propagation algorithm in this way, actually it uses the adjoint network of the forward network for computing the necessary gradients. Extending this idea to recurrent networks requires to unfold the network in time backwards for a finite duration for the construction of the adjoint , from which the necessary gradients of the forward network can be computed. The back propagation through time (BTT) [5,6] is closely related to this idea.

Consider a non-linear dynamic system,

$y(k) = f(y(k-1), y(k-2), \ldots, y(k-n), u(k), u(k-1), \ldots, u(k-m), W_f(k))$ is represented by a recurrent neural network whose parameters at time, $k = W_f(k)$.

Let the scalar cost be $J(k) = \frac{1}{2} e(k)^t e(k)$

Where $e(k) = y(k) - y_p(k)$; $y_p(k)$ = target o/p ; $y(k)$ =network o/p.

If the error is back propagated through the adjoint model whose initial conditions are set to zero, with the adjoint input = e(k) at time k and zero elsewhere , then, the adjoint states are given by

$\lambda(0) = e(k)$; $\lambda(i) = \Im^T_{y(k-i)} f(k-i+1)\, \lambda(i-1)$; $i=1,2,\ldots\varphi$ and the gradients are given by

$\nabla_{wf(k-\varphi)} J(k) = \Im^T_{wf(k-\varphi)} f(k-\varphi)\, \lambda(\varphi)$ ……………..(2)

where , $\Im^T_{y(k-i)} f(k-i+1)$ and $\Im^T_{wf(k-\varphi)} f(k-\varphi)$ are the Jacobians obtainable from map f(.) at different time instances.

The gradient obtained $\nabla_{wf(k-\varphi)} J(k)$ is the gradient of the current performance with respect to an earlier parameter. This can cause problems in the update algorithm if proper care is not taken. However, if we keep the parameters of the network constant for a certain finite duration, then the difficulty can be overcome by sacrificing some information which otherwise could have been used [3].

This method of obtaining the gradients using ABP is computationally advantageous compared to DBP since one single adjoint network gives all the required gradient components. Also, it is the true generalisation of the static back propagation.

4. IDENTIFICATION WITH STATIC BACK PROPAGATION

The general non-linear dynamic plant to be identified is

$y_p(k) = f(y_p(k-1), y_p(k-2), y_p(k-3), u(k), u(k-1))$.

$$f(.) = (y_p(k-1)*y_p(k-2)*y_p(k-3)*u(k-1)*(y_p(k-3)-1) + u(k)) / (1+y_p(k-2)\hat{}2 + y_p(k-3)\hat{}2).$$

The neural network N(.) with 3 layers , 20 neurons in the first layer with tanh sigmoids, 10 neurons in the second layer with tanh sigmoids and 1 neuron in the output layer with linear activation is used to represent the series-parallel identification model [1] used to identify the plant above. There are five inputs for the network. The identification model is represented as

$y_p(k) = N(y_p(k-1), y_p(k-2), y_p(k-3), u(k), u(k-1), \theta)$, where θ is the parameter vector of the network. For general training, uniformly distributed true random input sequence of 1000 samples in the amplitude range (-1,1) was chosen. Batch updating with static back propagation was used to adjust the weights of the neural network.

The responses of the trained identification model and the plant are given in figures 1 – 4. The identification model responses for the untrained random and other inputs are very good and indicates convergence of the identification process. Here, since series-parallel identification model is used, the identification could be done using the static back propagation algorithm where the error is back propagated through the moving average part only. As mentioned earlier, this series-parallel model cannot give information about the internal dynamics of a system when the relation between output and internal dynamics is not invertible. If such a requirement arises, then we have to use a parallel identification model in which case static back propagation cannot be used. This is the motivation to use dynamic back propagation and adjoint back propagation in the following examples.

5. IDENTIFICATION USING DYNAMIC BACK PROPAGATION

The plant to be identified here is chosen as

$y_p(k) = 1.82 * y_p(k-1) - 0.83 * y_p(k-2) + 0.007 * f(u(k))$

Where $f(u) = (u-0.8)*u*(u+0.8)$.

A parallel identification model is chosen here to demonstrate the dynamic back propagation method. The model is

$\hat{y}_p(k) = 1.82*\hat{y}_p(k-1) - 0.82 * \hat{y}_p(k-2) + N(u(k))$.

Where, N(.) represents the neural network as in the example: 1 , except for that now it has only one input. The sensitivity network is constructed as in fig 5 to get all the $\delta y_p / \delta\theta_j$ where, θ_j is a typical parameter of the network. For each parameter in the network separate sensitivity networks were formed. Observation time length is chosen as 2, which is same as the order of the system [3]. Training input is a uniformly distributed, amplitude limited, true random signal.

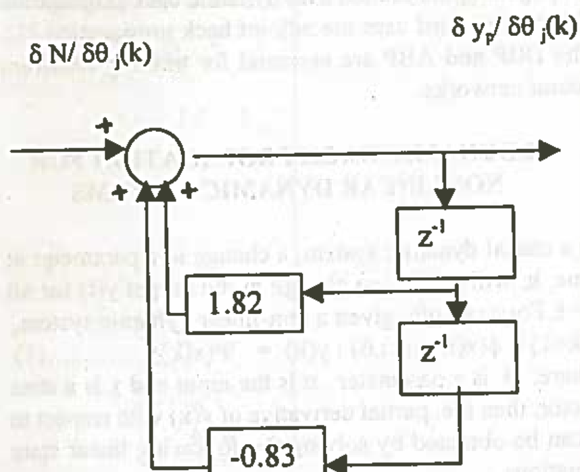

Figure 5

Using $\delta y_p / \delta\theta_j(k)$ and error at each instant, the required gradients are computed for adjusting the parameters of the network. The method of constructing the forward sensitivity networks and obtaining gradients in this way is the dynamic back propagation method. Fig 6 - 8 show the responses of the identification model as compared to the actual plant for trained and untrained inputs. The identification process is seen to be converged sufficiently well.

6. IDENTIFICATION USING ADJOINT BACK PROPAGATION

Here a general non-linear MIMO dynamical system described by the following equation is the plant to be identified.

$$\begin{bmatrix} y_{p1}(k) \\ y_{p2}(k) \end{bmatrix} = \begin{bmatrix} f_1(.) \\ f_2(.) \end{bmatrix} \quad \text{where,}$$

$$f_1(.) = [\, y_{p1}(k-1) * y_{p1}(k-2) * y_{p1}(k-3) * u(k-1) \{ y_{p1}(k-3)-1\} + u(k) + y^2_{p2}(k-1) * \{ y^2_{p2}(k-2)+y^2_{p2}(k-3)\}] / [\, 1+ y^2_{p1}(k-2) + y^2_{p1}(k-3) \,] \quad \text{and}$$

$$f_2(.) = [\, y_{p2}(k-1) * y_{p2}(k-2) * y_{p2}(k-3) * u(k-1) \{ y_{p2}(k-3)-1\} + u(k) + y^2_{p1}(k-3) * \{ y^2_{p1}(k-1) + y^2_{p1}(k-2) \}] / [\, 1 + y^2_{p2}(k-2) + y^2_{p2}(k-3)].$$

The identification model is represented by the recurrent structure

$$\hat{y}_{p1}(k) = N_1\{\hat{y}_{p1}(k-1),\hat{y}_{p1}(k-2),\hat{y}_{p1}(k-3),\hat{y}_{p2}(k-1),\hat{y}_{p2}(k-2), \hat{y}_{p2}(k-3),u(k),u(k-1)\}$$

$$\hat{y}_{p2}(k) = N_2\{\hat{y}_{p1}(k-1),\hat{y}_{p1}(k-2),\hat{y}_{p1}(k-3),\hat{y}_{p2}(k-1),\hat{y}_{p2}(k-2), \hat{y}_{p2}(k-3),u(k),u(k-1)\}$$

A recurrent neural network with 3 layers, 8 inputs and 2 outputs N1,N2 is used to approximate the above vector function with random signal as the training input and using adjoint back propagation as the training algorithm as explained in section:3. The observation time length was chosen as 3 which is equal to the order of the system. The number of neurons in the first and second layers are the same as in earlier examples. Figures 9 – 16 show the identified model response in comparison with actual plant.

7.DISCUSSION AND CONCLUSIONS

The three different gradient learning methods are demonstrated through these examples for system identification purposes. The dynamic back propagation method in example2 and the adjoint back propagation in example3 demonstrate their applicability for recurrent identification structures. Uniformly distributed , amplitude limited , true random signal is a good general input for effective identification of non-linear systems in a domain of interest.

Static back propagation can be used for identifying non-linear dynamic systems in general, but state information cannot be obtained from the input-output model identified if state-output relation is not invertible. However, this method is computationally the most efficient.

Dynamic back propagation or adjoint back propagation has to be used for identifying non-linear dynamic systems for obtaining state information of the system when state-output relation is not invertible. ABP is computationally more efficient than DBP if the identification structure is properly selected. Also ABP is the true generalisation of the static back propagation for dynamic systems. DBP is a forward method and not a backward method. ABP is closely related to back propagation through time (BTT).

Even though, in all the examples, the learning rate was selected small enough to ensure convergence, in an on-line identification problem, divergence can occur. Developing new algorithms and modifying existing algorithms to ensure uniform convergence is a major area of current research. Such works are currently investigated.

References:

1. K.S. Narendra and K. Parthasarathy, "Identification and Control of Dynamical systems using neural networks: IEEE Trans on neural network, vol. I , no. I, March 1990,pp 4-27.

2. K.S. Narendra and K.Parthasarathy , " Gradient methods for the optimisation of Dynamical systems containing neural networks ", IEEE Trans on neural networks , vol.2 , no.2, March 1991 pp 252-262

3. B.Srinivasan , U.R.Prasad and N.J. Rao, " Back propagation through adjoint for the identification of non-linear systems using recurrent neural models", IEEE Trans on neural networks , vol 5 , no. 2 , March 1994 , pp 213-228.

4. P.Zarchan, " Strategic and tactical missile guidance ' , vol 124, AIAA publication , 1990.

5. P.J.Werbos , " Backpropagation through time: what it does and how to do it" ,Proc IEEE , vol 78, Oct 1990, pp 1550 - 1560 .

6. D.Nguyen and B.Widrow , " Neural networks for self learning control systems",IEEE control systems magazine , vol .10 , no.3 , April 1990, pp 18 - 23 .

7. Rekha Sebastian ," Neural networks for non-linear dynamic system identification" M.Tech thesis, University of Kerala , Dec 1997.

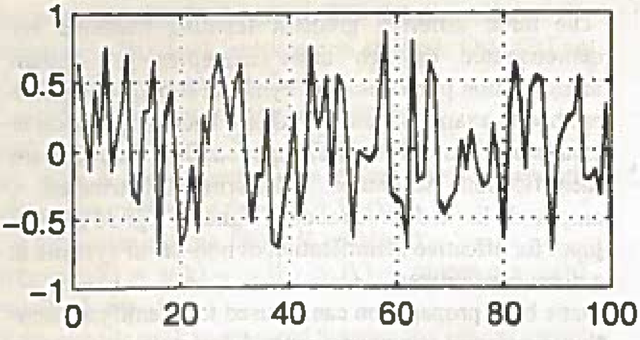

Figure 1

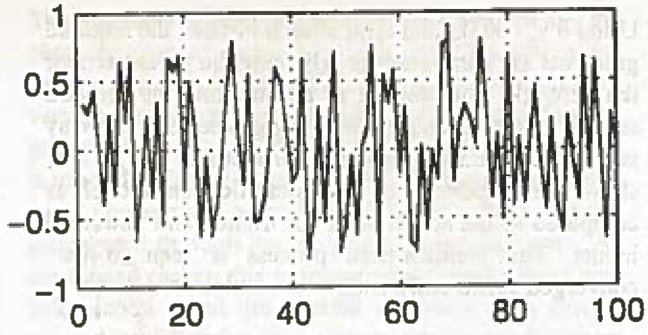

Figure 2

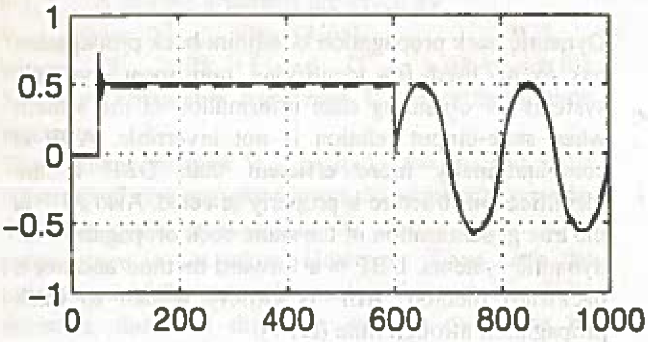

Figure 3

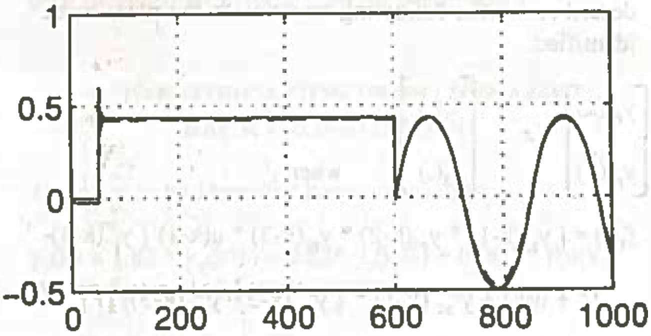

Figure 4

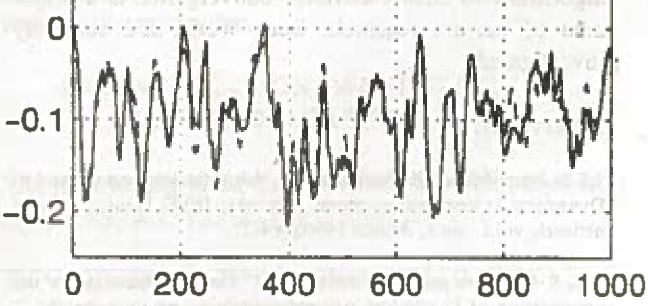

Figure 6

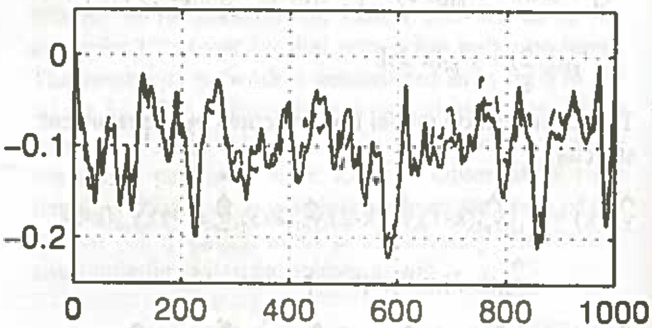

Figure 7

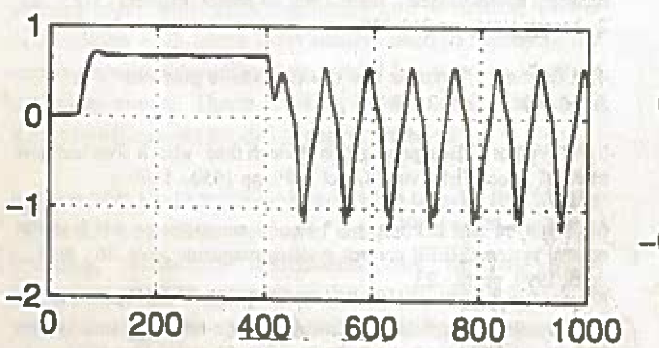

Figure 8

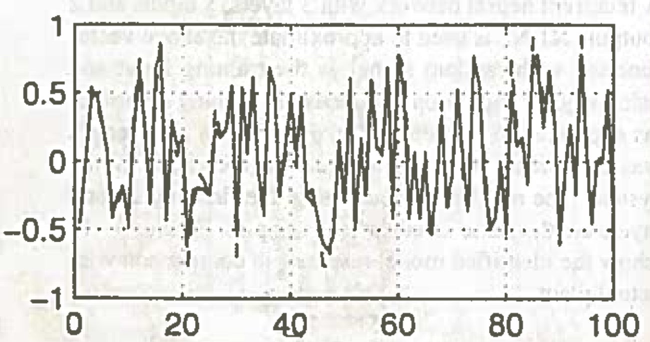

Figure 9

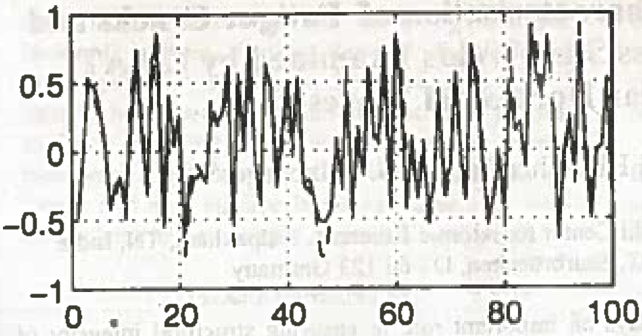

Figure 10

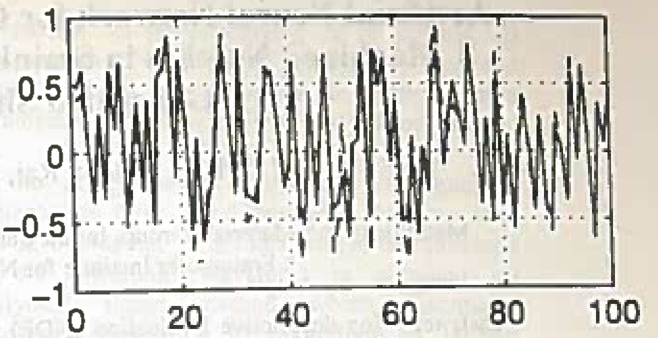

Figure 11

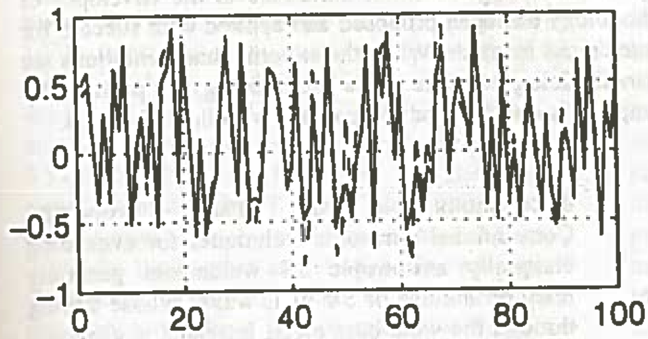

Figure 12

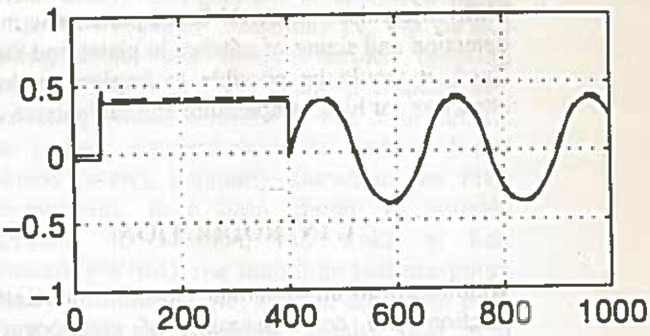

Figure 13

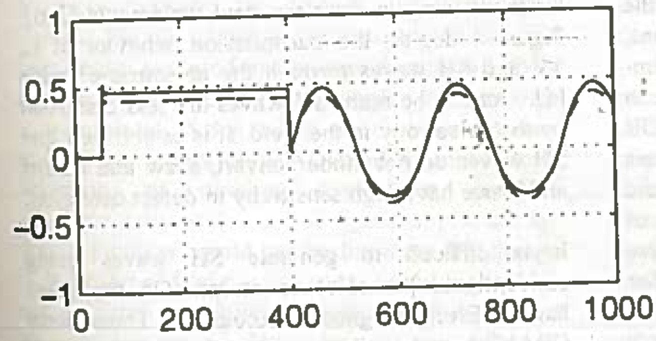

Figure 14

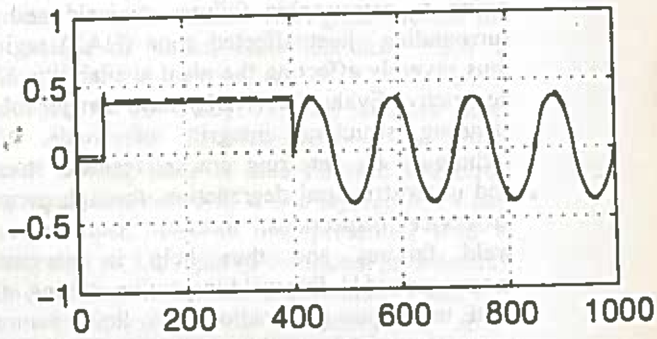

Figure 15

— reference output

- model output

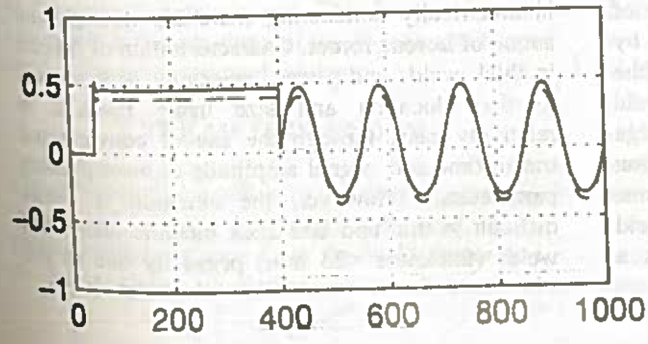

Figure 16