Monograph:
Morles, E.C. and Mort, N. (1992) On-Line Control of Dynamic Systems Using Feedforward Neural Networks. UNSPECIFIED. ACSE Research Report 457 . Department of Automatic Control and Systems Engineering

ON-LINE CONTROL OF DYNAMIC SYSTEMS USING FEEDFORWARD NEURAL NETWORKS

BY

ELIEZER COLINA MORLES

AND

NEIL MORT

DEPARTMENT OF AUTOMATIC CONTROL

AND

SYSTEMS ENGINEERING

UNIVERSITY OF SHEFFIELD

Mappin Street

Sheffield S1 3JD

Research Report Number 457

August 1992.

Abstract

An artificial feedforward neural network is used for on-line control purposes of a class of single input single output nonlinear dynamic systems. The neural network's task is to provide one step ahead predictions of the unknown input-output behavior of the dynamic system to the controller in order to drive the system's behavior according to the specifications given by a reference model.

The learning algorithm used to update the weights of the network allows dynamic adjustment of their values to cope with varying signals coming from the unknown plant.

Keywords: Feedforward neural network, On-line control, One step ahead predictor, Reference model, Learning algorithm.

1 INTRODUCTION

An artificial feedforward neural network can be thought of as a topologically layer-like configuration of neurons interconnected by variable weighting values. The outputs of the neurons of one layer might be connected to some or all of the neuron inputs of the next layer. Figures 1 and 2 illustrate a single neuron as an adaptive element, and the feedforward neural network considered in this report, respectively.

A crucial role in the potential capabilities of a neural network is played by its embedded learning algorithm. In a supervised learning enviroment, a very well known procedure to minimize the mean square error between the teaching signal and the actual output in a single layer network is called the delta rule [1]. The generalized delta rule, usually known as the back propagation algorithm [1], is often used for updating the weights in multilayer feedforward networks. Some identification and control of dynamic systems applications of feedforward neural networks trained using back propagation algorithms can be found in several recently published articles [2, 3, 4, 5]. However, a number of difficulties have been reported associated with back propagation algorithms, for example:

- A satisfactory learning process is achieved only after a long training sequence [6].
- Large amount of computing time are required [6].
- Convergence is not guaranteed due to local minima plateaus [6].
- The nonlinear activation functions must be differentiable [7].

The supervised learning algorithms used in this report, operate by making an error between the teaching signal and the neural network output satisfy an asymptotically stable difference equation [7]. These types of algorithms are derived from a variable structure control framework [7]. Unlike back propagation, these algorithms do not need long training sequences, and differentiability along the network signal paths is not required. Different from [7], the effect of considering non-constant teaching signals and a different set of input signals to the neural network is taken into consideration in the learning algorithm discussed here. Under these considerations the neural network performs with higher precision to track or predict the values of the teaching signal. Digital simulations are presented to support this assertion. In this report, the nonlinear systems to be controlled are of the class described by the following nonlinear difference equation:

$$y_p(k+1) = f(y_p(k), y_p(k-1), ..., y_p(k-n+1)) + u(k), \tag{1}$$

where $f(..)$ is an unknown function. In order to achieve the aforementioned objective we shall propose a model reference-neural network-based control scheme and present the results obtained from digital simulations of the scheme. This paper is organized as follows. The next section contains a review of the adaptation algorithm presented in [7], followed by the modification arising from considering varying teaching signals. In section 4 we present a model reference-neural network-based control scheme, and the responses obtained from simulating the scheme using a nonlinear system. Both the original and the modified adaptation algorithms are tested. The last section contains conclusions and recommendations for further research.

2 THE ADAPTATION ALGORITHMS

Here we shall present an overview of the adaptation algorithm described in [7]. This formulation is different from back propagation algorithms, in that the learning parameters are adjusted to force the error between the actual and desired outputs to satisfy a stable difference error equation, rather than to minimize an error function. Figure 3 shows the schematic representation of the neural network depicted in figure 2. Notice that $W1(k)$, $W2(k)$, and $WI(k)$ are the values at time k, of the weight matrices,$X = [x_1, x_2, ..., x_n]^T$ is the input to the neural network, Y_d is the desired output vector, and the operator $\Gamma(.)$ satifies

$$\Gamma(-x) = -\Gamma(x). \tag{2}$$

With regard to figure 1, the weight vector $W(k)$ can be viewed as a controlled difference equation with the error $e(k)$, between the actual and desired outputs, as an output signal defined by a mapping g in the following terms

$$W(k+1) \quad = \quad h(W(k), U(k))$$

$$e(k) = g(W(k)), \tag{3}$$

where $W(k) \in \Re^n$, $U(k) \in \Re^n$, $e(k) \in \Re$. The output map can be used to define a switching plane of the form

$$S(0) = \{W \in \Re^n \mid e = g(W) = 0\} \tag{4}$$

A quasi-sliding mode is said to exist on $S(0)$ if there exists a control law $U(k)$ such that the behaviour of 3 satisfies the relation

$$\mid e(k+1) \mid < \mid e(k) \mid, \tag{5}$$

for $e(k) \neq 0$ [8]. Notice that the controllers U(k) in 3 are the correction terms of the present value of the weight vector. In order to derive an expression for the controllers, note that from figure 1

$$e(k) = y_d - y_n(k). \tag{6}$$

THEOREM 1.

If the weight vector W(k) of the adaptive neuron, shown in figure 1, is adjusted according to the rule:

$$W(k+1) = W(k) + \frac{\alpha e(k)\Gamma(x)}{x^T\Gamma(x)}, \tag{7}$$

with $0 < \alpha < 2$, then the error $e(k)$ will tend asymptotically to zero with the rate of convergence $(1 - \alpha)$, and a quasi-sliding mode on the zero switching plane $S(0)$ is guaranteed to exist [7].

PROOF.

From figure 1, note that

$$\begin{aligned} y_n(k) &= \sum_{i=1}^{n} w_i(k)x_i \\ &= W^T(k)X. \end{aligned} \tag{8}$$

Observe that

$$\begin{aligned} e(k+1) - e(k) &= y_d - y_n(k+1) - (y_d - y_n(k)) \\ &= y_n(k) - y_n(k+1) \\ &= \sum_{i=1}^{n} w_i(k)x_i - \sum_{i=1}^{n} w_i(k+1)x_i \\ &= -\sum_{i=1}^{n}(w_i(k+1) - w_i(k))x_i \\ &= -[W(k+1) - W(k)]^T X \end{aligned} \tag{9}$$

If $W(k+1) - W(k)$ is selected according to 7 then

$$\begin{aligned} e(k+1) - e(k) &= -X^T \frac{\alpha e(k)\Gamma(X)}{X^T\Gamma(X)} \\ &= -\alpha e(k) \qquad X \neq 0. \end{aligned} \tag{10}$$

Hence $e(k+1) = (1-\alpha)e(k)$ and if $0 < \alpha < 2$ then

$$\lim_{k\to\infty} e(k) = 0. \tag{11}$$

On the other hand, observe that

$$\mid e(k+1) \mid = \mid (1-\alpha)e(k) \mid < \mid e(k) \mid, \tag{12}$$

and therefore the quasi-sliding condition is satisfied. In order to extend the results of the previous theorem to three layers feedforward neural networks, of the type depicted in figure 2, observe that from the schematic representation of figure 3 we have that the vector $Y_n(k)$ of the output components $y_{nk}(k)$ can be represented as

$$Y_n(k) = [W1(k)]^T Z1(k), \tag{13}$$

where $W1(k) \in \Re^{n1xno}$, and $Z1(k) = \Gamma(Y1(k))$. Similarly, the vector Y1(k) of the components $y1_n(k)$ is

$$Y1(k) = [W2(k)]^T Z2(k), \tag{14}$$

with $W2(k) \in \Re^{n2xn1}$, and $Z2(k) = \Gamma(Y2(k))$. Finally the vector Y2(k) of the components $y2_m(k)$ is

$$Y2(k) = [WI(k)]^T X, \tag{15}$$

where $WI(k) \in \Re^{nixn2}$, and $X = [x_1x_2...x_{ni}]^T$. Note that the nonlinear operator Γ does not have to be a diagonal one. The error vector E(k) at time k is given by

$$E(k) = [e_1(k)e_2(k)...e_{no}(k)]^T = Y_d - Y_n(k), \tag{16}$$

Y_d being the desired output vector.

The weight updates are represented by the following equations:

$$W1(k+1) = W1(k) + U1(k) \tag{17}$$

$$W2(k+1) = W2(k) + U2(k) \tag{18}$$

$$WI(k+1) = WI(k) + UI(k) \tag{19}$$

THEOREM 2.

If the weight update matrices UI(k), U2(k), and U1(k) are respectively chosen as

$$UI(k) = -\frac{2\Gamma(X)[Y2(k)]^T}{X^T\Gamma(X)}; \qquad X^T\Gamma(X) \neq 0, \tag{20}$$

$$U2(k) = -\frac{2\Gamma(Z2(k))[Y1(k)]^T}{[Z2(k)]^T\Gamma(Z2(k))}; \quad [Z2(k)]^T\Gamma(Z2(k)) \neq 0, \tag{21}$$

$$U1(k) = \frac{\Gamma(Z1(k))[AE(k)]^T}{[Z1(k)]^T\Gamma(Z1(k))}; \quad [Z1(k)]^T\Gamma(Z1(k)) \neq 0, \tag{22}$$

then the error vector E(k) satisfies the following asymptotically stable difference equation

$$E(k+1) = (I - A)E(k), \tag{23}$$

where I is the identity matrix, and $(I - A)$ is an $n_o x n_o$ matrix with eigenvalues in the open unit circle of the complex plane.

PROOF.

Observe that

$$E(k+1) - E(k) = Y_n(k) - Y_n(k+1), \tag{24}$$

provided that $Y_d(k)$ is constant. Substituting equations 13, 14, 15, 17, 18, and 19 into equation 24 yields

$$\begin{aligned} E(k+1) - E(k) &= [W1(k)]^T \{\Gamma(Y1(k)) - \Gamma\{[W2(k) \\ &+ U2(k)]^T\Gamma[Y2(k) + [UI(k)]^T X]\}\} \\ &- [U1(k)]^T\Gamma\left\{[W2(k) + U2(k)]^T\Gamma[Y2(k)\right. \\ &+ \left.[UI(k)]^T X\right\} \end{aligned} \tag{25}$$

Now replacing the transposes of 20, 21, and 22 into equation 25 yields equation 23. Observe that the learning algorithm, represented by equation 7 for single layer networks, was derived assuming a constant teaching signal y_d. Next, we shall consider non-constant teaching signals in the adaptation algorithm.

THEOREM 3.

If the weight vector W(k) of the adaptive neuron of figure A, is adjusted according to the rule

$$W(k+1) = W(k) + \frac{\alpha(e(k) + y_d(k+1) - y_d(k))\Gamma(X)}{X^T\Gamma(X)}, \tag{26}$$

with $0 < \alpha < 2$, then the error e(k) will tend asymptotically to zero with the rate of convergence $(1 - \alpha)$, and a quasi-sliding mode on the zero switching plane S(0) is guaranteed to exist.

PROOF.

The proof is identical to the one of theorem 1.

NOTE.

It is worth mentioning that in our dynamic system control application, the desired output $y_d(k+1)$, for the neural network is not known in advance, -(y_d being the actual system output)-, however by using the present and the previous values of the system's outputs as values for the teaching signal, it is still possible to guarantee an asymptotically stable difference error equation, and the existence of a quasi-sliding mode on the zero switching plane S(0). This is:

$$\begin{aligned} e(k) - e(k-1) &= y_d(k) - y_n(k) - (y_d(k-1) - y_n(k-1)) \\ &= -(y_n(k) - y_n(k-1)) + y_d(k) - y_d(k-1) \\ &= -[W(k) - W(k-1)]^T X + y_d(k) - y_d(k-1). \end{aligned} \quad (27)$$

By selecting

$$W(k) - W(k-1) = \frac{\alpha(e(k) + y_d(k) - y_d(k-1))\Gamma(X)}{X^T\Gamma(X)}, \quad (28)$$

and substituting it into 27, yields

$$\begin{aligned} e(k) - e(k-1) &= -\alpha e(k) \\ &= (1+\alpha)^{-1} e(k-1). \end{aligned} \quad (29)$$

If $\alpha > 0$, then

$$\lim_{k \to \infty} e(k) = 0. \quad (30)$$

Furthermore

$$\mid e(k) \mid = \mid (1+\alpha)^{-1} e(k-1) \mid < \mid e(k-1) \mid, \quad (31)$$

for $\alpha > 0$, and therefore the quasi-sliding mode is guaranteed to exist on the switching plane $S(0) = e(k) = 0$. The extension of this result to a three layers network is straightforward.

3 NEURAL NETWORK-BASED CONTROL SCHEME

In a model reference control system environment the parameters of the controller are dynamically adjusted in such a way the output of the system under scrutiny follows the output of a defined model that exhibits desirable specifications [9].

In this section we shall present a model reference control scheme where the adjustable control parameters are directly supplied by a neural network [10]. It will be assumed that the plant is noise free, with a known structure, although its parameters will be considered unknown. The proposed scheme is shown in figure 4. Mathematically speaking the results can be illustrated with the following example. Let the nonlinear plant be described by

$$y_p(k+1) = f(y_p(k), y_p(k-1)) + u(k), \tag{32}$$

with the unknown function [2]

$$f(y_p(k), y_p(k-1)) = \frac{y_p(k)y_p(k-1)(y_p(k)+2.5)}{1+y_p^2(k)+y_p^2(k-1)}. \tag{33}$$

Let the reference model be described by

$$y_m(k+1) = 0.6y_m(k) + 0.2y_m(k-1) + r(k), \tag{34}$$

where $r(k)$ is a bounded reference input.

If the function $f(y_p(k), y_p(k-1))$ were known, it would be possible to control the plant adaptively by selecting

$$u(k) = -f(y_p(k), y_p(k-1)) + 0.6y_p(k) + 0.2y_p(k-1) + r(k). \tag{35}$$

In this case the output error between the plant and the reference model would be asymptotically stable [2].

In practice, the function $f(y_p(k), y_p(k-1))$ is not known, and therefore in order to be able to implement 35, it is necessary to estimate the value of f(..) on line. Let $n(y_p(k), y_p(k-1))$ be the estimate of $f(y_p(k), y_p(k-1))$ given by a neural network, and let us define the teaching signal

$$z(k) = y_p(k) - u(k-1). \tag{36}$$

Notice that $z(k)$ equals $f(y_P(k-1), y_p(k-2))$. It is required that as time increases, the error signal

$$e_n(k) = z(k) - y_n(k), \tag{37}$$

with $y_n(k)$ the output of the neural network, goes to zero asymptotically. It has been showed that by using adaptive rules of the type described by 7 or 26, the error signal equation 37 can be driven to zero asymptotically. Under this circunstances, the network output $y_n(k)$ will track the teaching signal z(k). The value $y_n(k+1)$, also given by the neural network, will be a one step ahead prediction of the teaching signal. This is

$$y_n(k+1) = N(y_p(k), y_p(k-1)) \rightarrow z(k+1) = f(y_p(k), y_p(k-1)). \tag{38}$$

Now the control action represented by equation 38 can be rewritten as

$$u(k) = -N(y_p(k), y_p(k-1)) + 0.6y_p(k) + 0.2y_p(k-1) + r(k). \quad (39)$$

Next we shall present some results obtained from digitally simulating the proposed control scheme using a single layer, and a three layer neural networks. Both the original algorithm, as presented in [7], and the modified version showed in this report, will be used to update the neural network weights.

For the sake of comparing the performance of the proposed control scheme when different sets of input signal to the neural network are used, we shall consider a delayed control sequence of u(k), and unitary step functions as input to the network.

Figures 5, 6, 7, and 8 show the results obtained using a single layer neural network with u(k-1), u(k-2), and u(k-3) as its input values. In this case it was assumed a constant teaching signal. Note the magnitude of the tracking error between the plant and the reference model outputs sketched in figure 6. Figures 9, 10, 11, and 12 represent the results obtained using the same single layer neural network of the previous example, but assuming unitary step functions at its its input, and considering that the teaching signal was not a constant. Here the magnitude of the tracking error -(figure 10)- is smaller than in the previous case. The results sketched in figures 13, 14, 15, and 16 were obtained using a three layer neural network with delayed values of the control sequence at its input, and assuming that the teaching signal was a constant. There were 4 neurons in the first hidden layer, and 2 neurons in the second hidden layer. Finally, figures 17, 18, 19, and 20 correspond to the results when the previous three layer neural network was used with unitary step functions at its input, and considering a non-constant teaching signal.

In all studied cases the reference model output was the sine function $Sin(5T) + 2$, and the sampling time for the neural network was 0.01 seconds. It is important to mention that when the input vector X to the neural network was a delayed control sequence of u(.), it was necessary to initiate the controller with a value different from zero in order to avoid division by zero in the neural network operation. This difficulty is solved by using step functions as input to the neural network.

4 CONCLUSIONS

An adaptive neural network-based control scheme was proposed and its performance was digitally simulated using a unknown plant which belong to a certain class of nonlinear systems. The role of the multilayer neural network considered here was to make an on-line estimation of the unknown input-output behavior of the plant. This estimation represents an one step ahead predictor, and was used to implement the nonlinear adaptive control action.

The updating algorithm for the weights of the neural network operates by making the error between the desired and the actual outputs satisfy an asymptotically stable difference equation. In the proposed algorithm, the teaching signal can be a varying one. Also the input to the neural network can be selected as any strictly positive or negative valued function.

The best results were obtained using a three layer neural network to estimate the unknown input-output behavior of the plant, and assuming that the teaching signal was not a constant one. In this case, the inputs to the neural network were three unitary step functions.

References

[1] Rumelhart D.E.; McClelland J.L. "Parallel Distributed Processing: Explorations In The Macrostructure Of Cognition".
Chapter 8, A Bradford Book, The MIT Press, Cambridge 1986.

[2] Narendra K.; Parthasarathy K. "Identification And Control Of Dynamic Systems Using Neural Networks".
IEEE Trans. Neural Networks, Vol. 1, Number 1, pp 4-27, March 1990.

[3] Fu-Chuang C. "Back Propagation Neural Networks For Nonlinear Self-Tuning Adaptive Control".
IEEE Control Systems Magazine, April 1990

[4] Sanner R. M.; Akin D. L. "Neuromorphic Pitch Attitude Regulator Of An Underwater Telerobot".
IEEE Control Systems Magazine, pp 62-67, April 1990.

[5] Pham D. T.; Liu X. "Neural Networks For Discrete Dynamic System Identification".
Journal Of Systems Engineering, Number 1, pp 51-60, 1991.

[6] Davalo E.; Naim P. "Neural Networks".
MacMillan Computer Science Series
Editions Eyrolles, Paris 1991.

[7] Sira-Ramirez H.; Zak S. "The Adaptation Of Perceptrons With Applications To Inverse Dynamics Identification Of Unknown Dynamic Systems".

IEEE Trans. Systems, Man, And Cybernetics, Vol. 21, Number 3,pp 634-643, May-June 1991.

[8] Spurgeon S. K. "On Conditions For The Development Of Stable Discrete Time Sliding Mode Control Systems".
Math. Rep. Number A124, Dept. Math. Sci., Univ. Technol. Loughborough, Leics., April 1990.

[9] Astrom K. J.; Wittenmark B. "Adaptive Control".
Addison-Wesley, Reading Ma. 1989.

[10] Colina-Morles E.; Mort N. "Neural Network Based Adaptive Control Design".
Journal of Systems Engineering, Springer-Verlag London Ltd. -(To be published)-

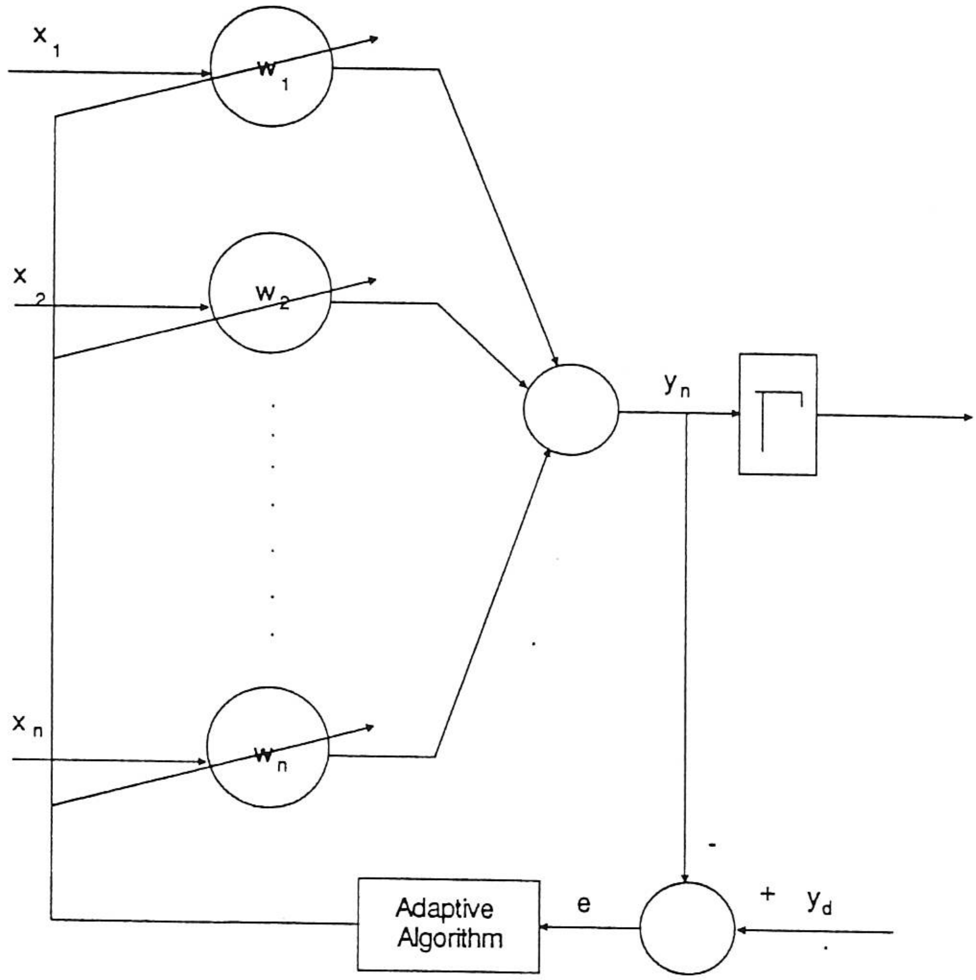


Figure 1: Single layer neuron as an adaptive element.

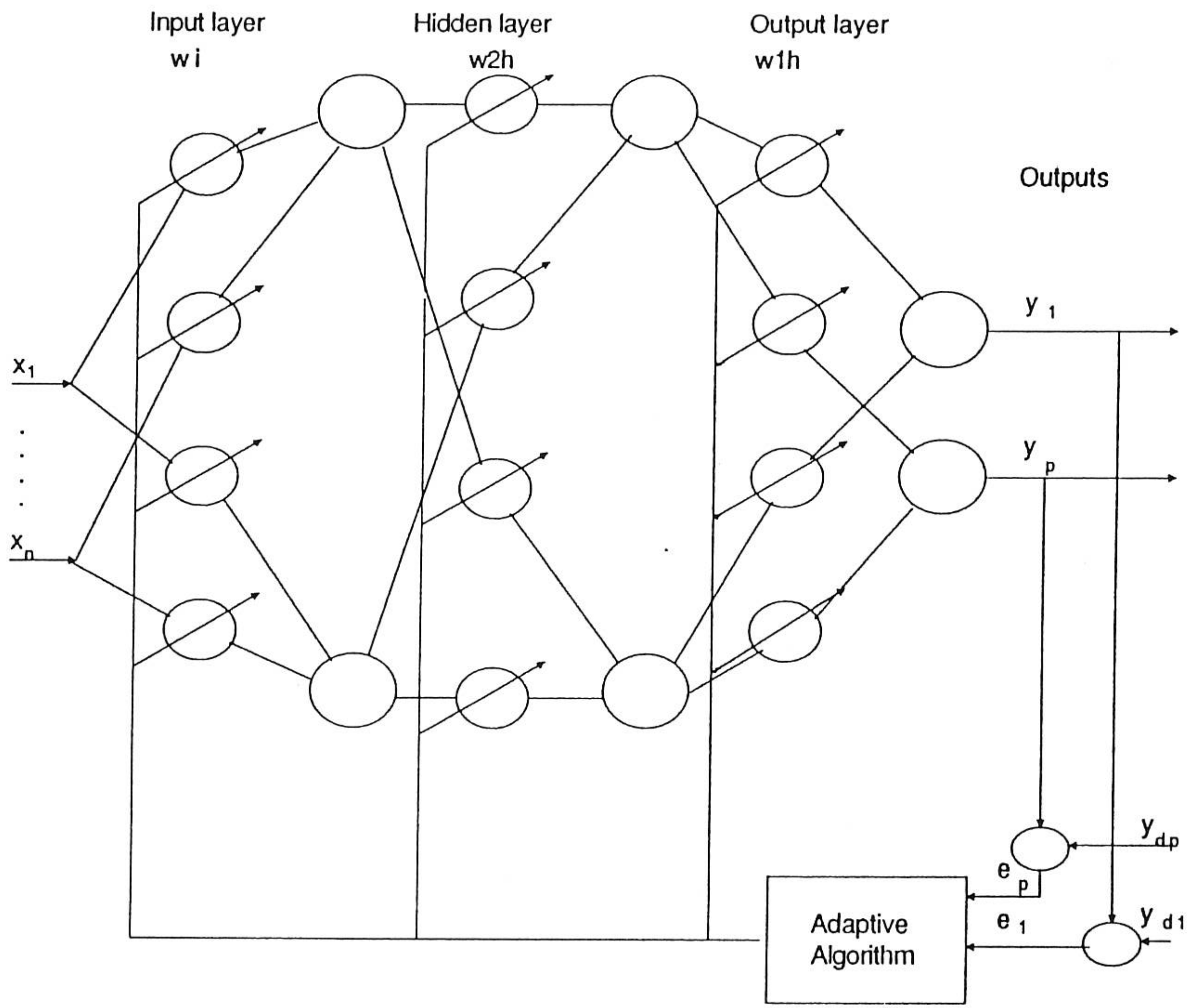


Figure 2: Three layer neural network.

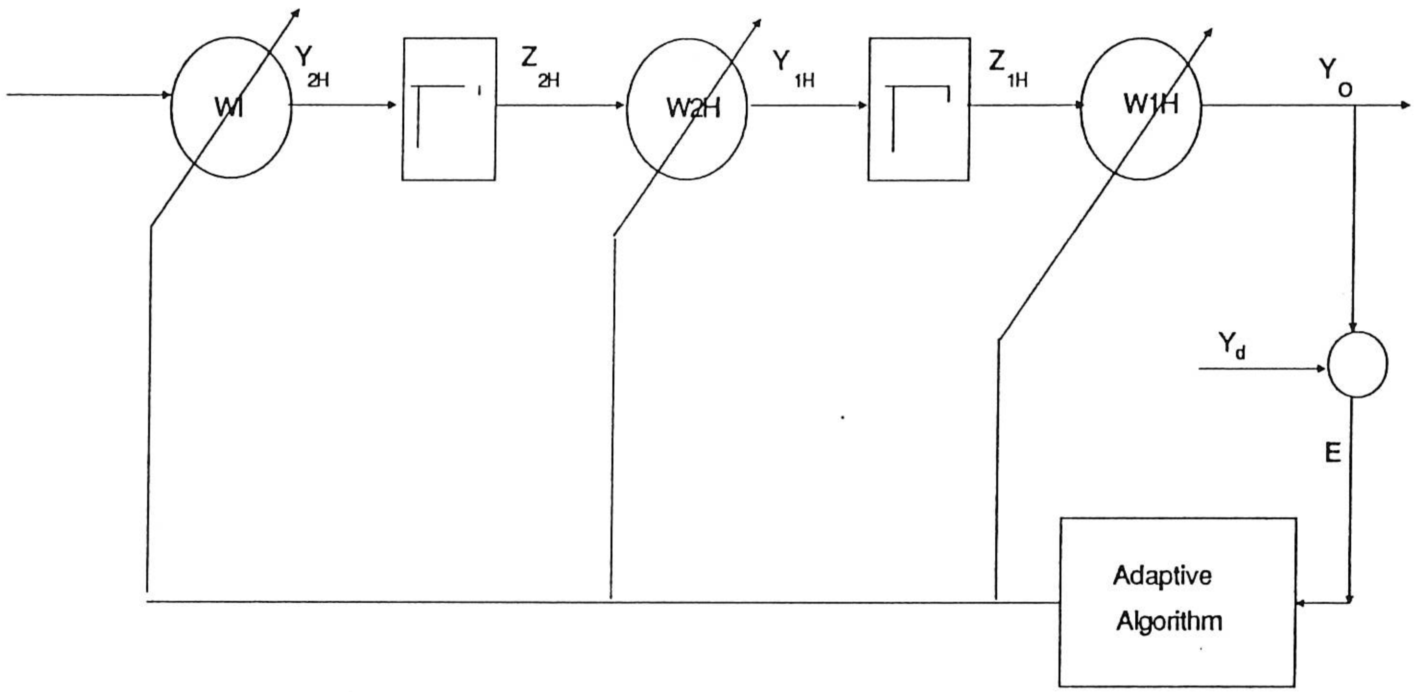


Figure 3: Schematic representation of the three layer neural network.

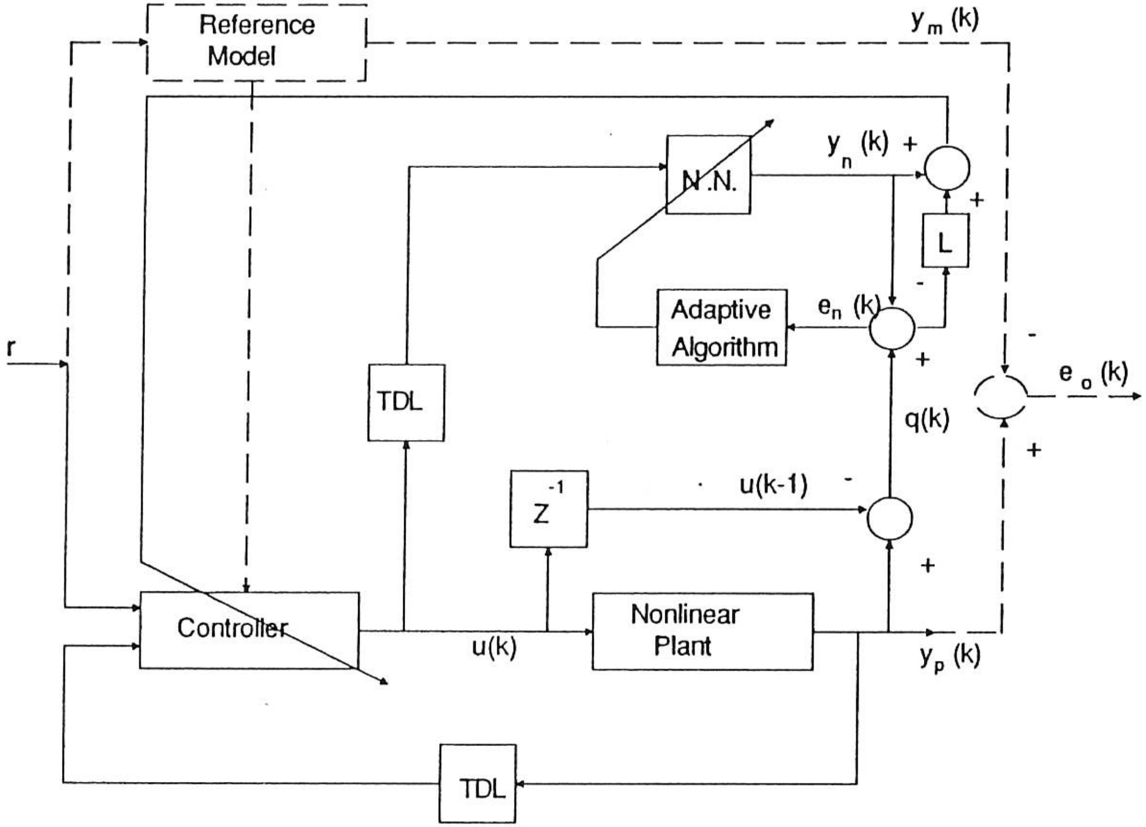


Figure 4: Model reference neural network-based control scheme.

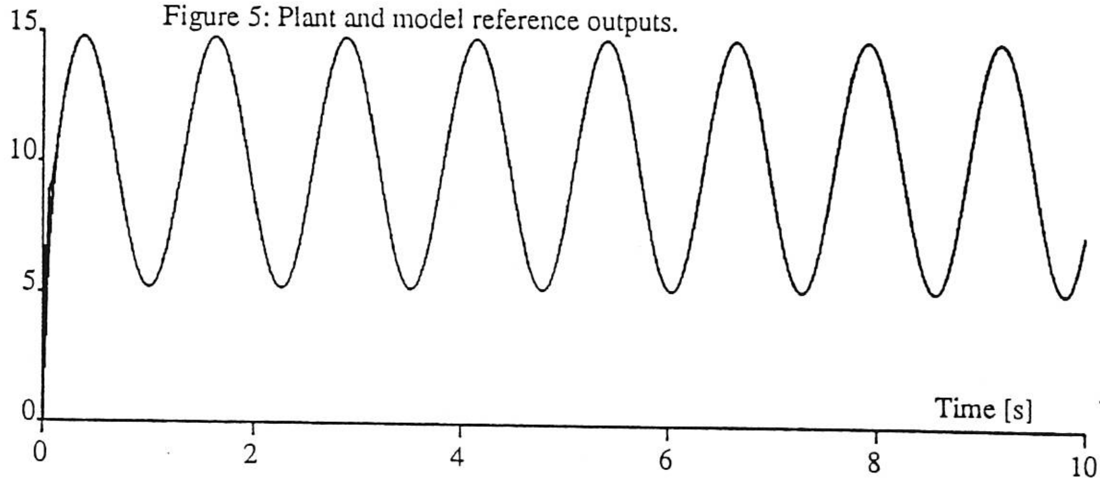


Figure 5: Plant and model reference outputs.

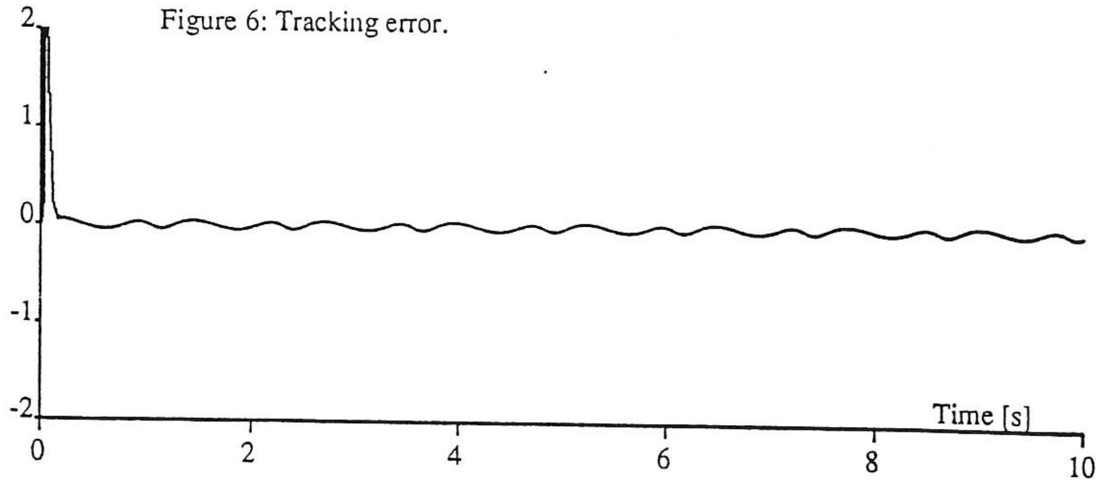


Figure 6: Tracking error.

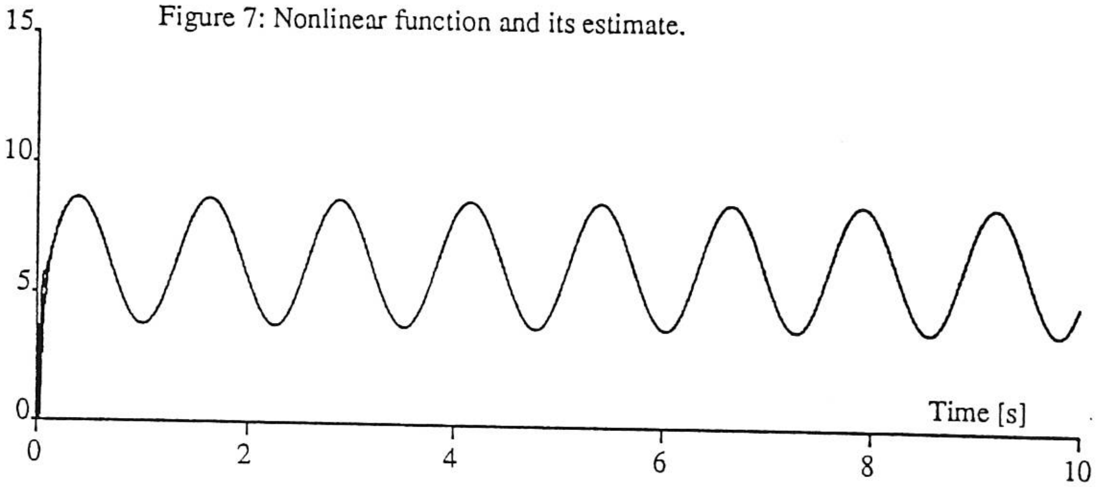


Figure 7: Nonlinear function and its estimate.

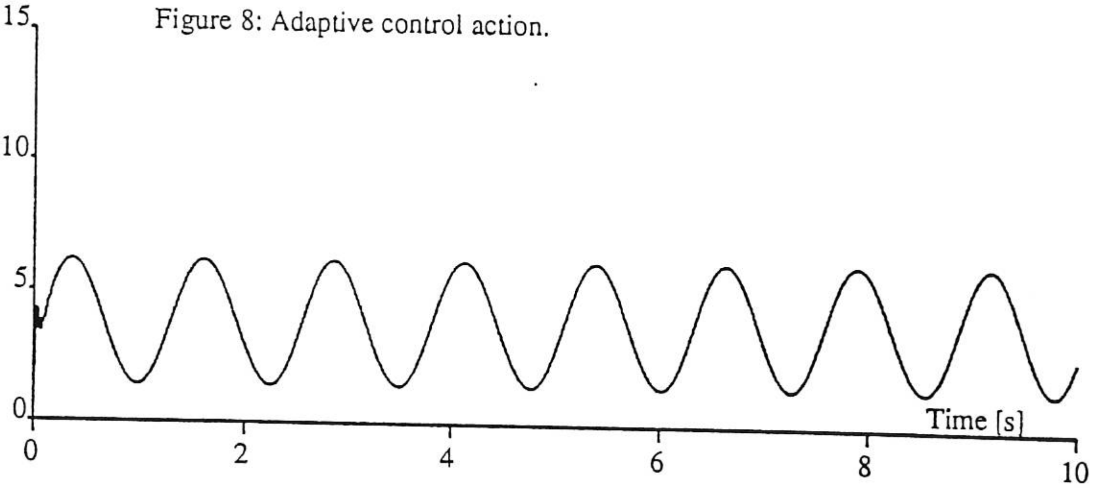


Figure 8: Adaptive control action.

Figure 9: Plant and model reference outputs.

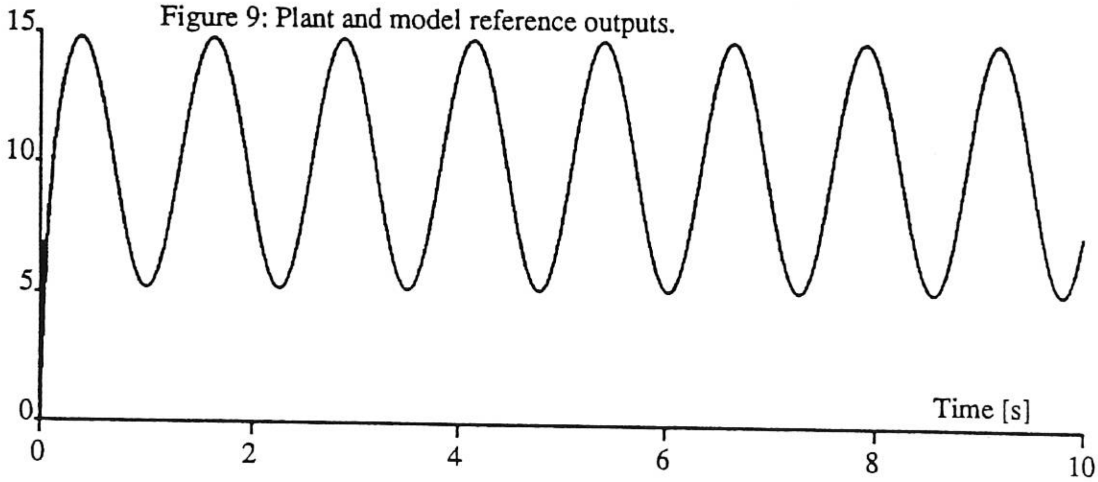
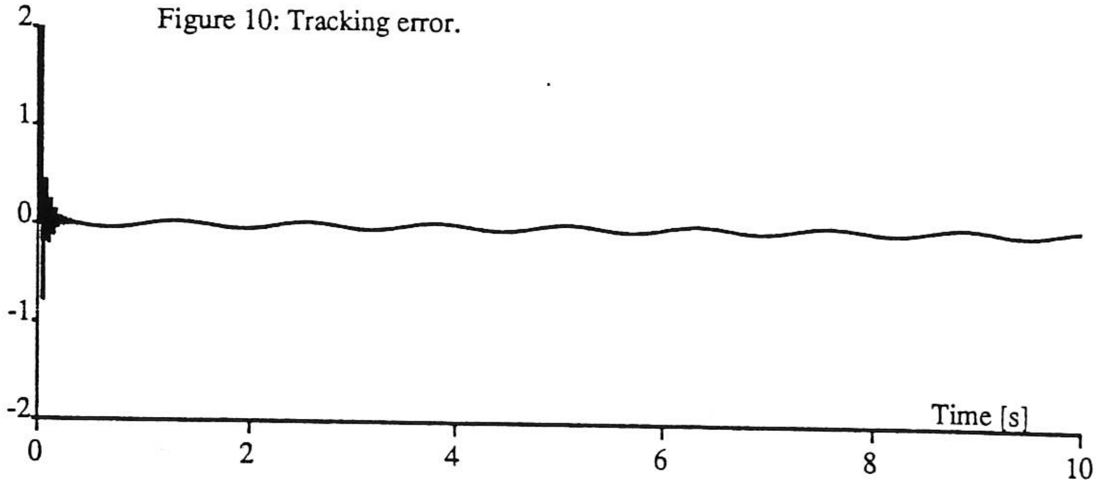


Figure 10: Tracking error.



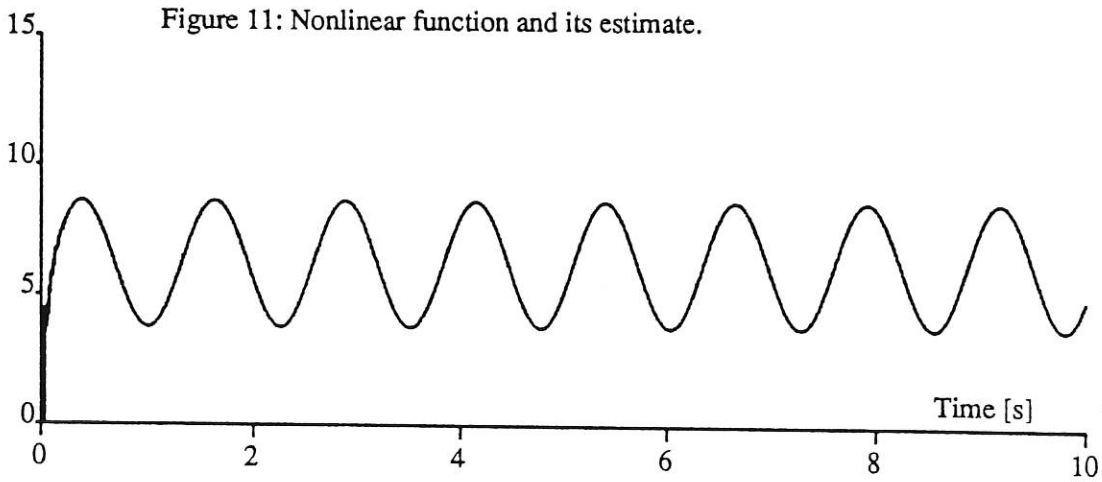


Figure 11: Nonlinear function and its estimate.

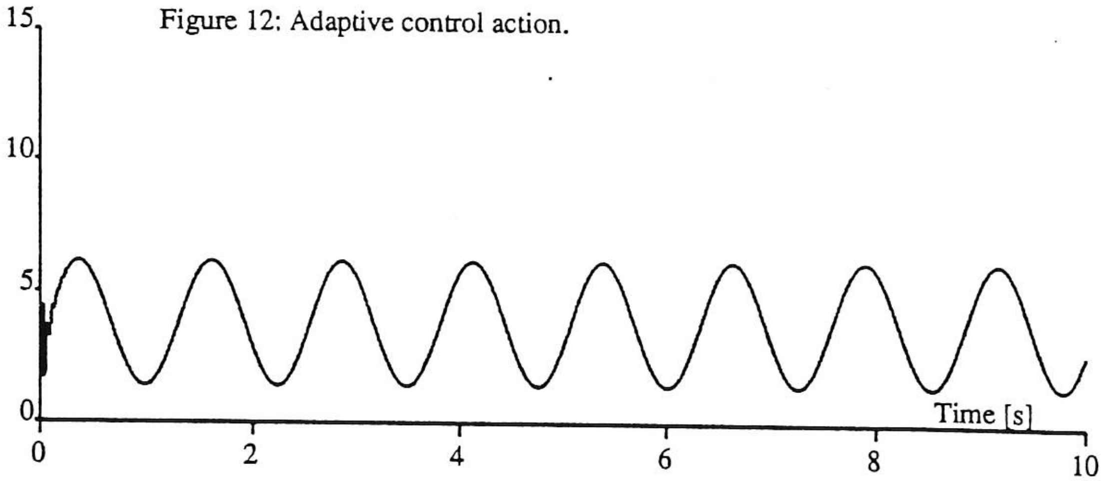


Figure 12: Adaptive control action.

Figure 13: Plant and model reference outputs.

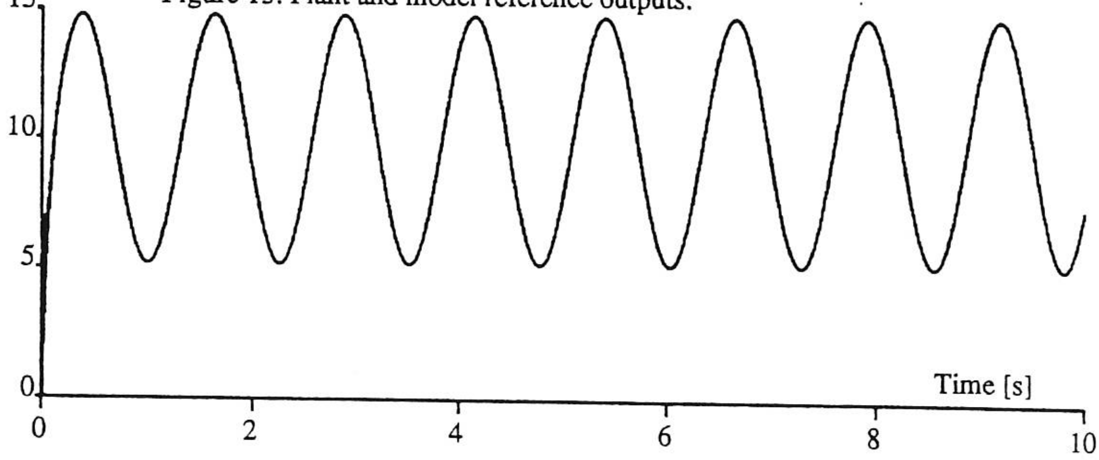
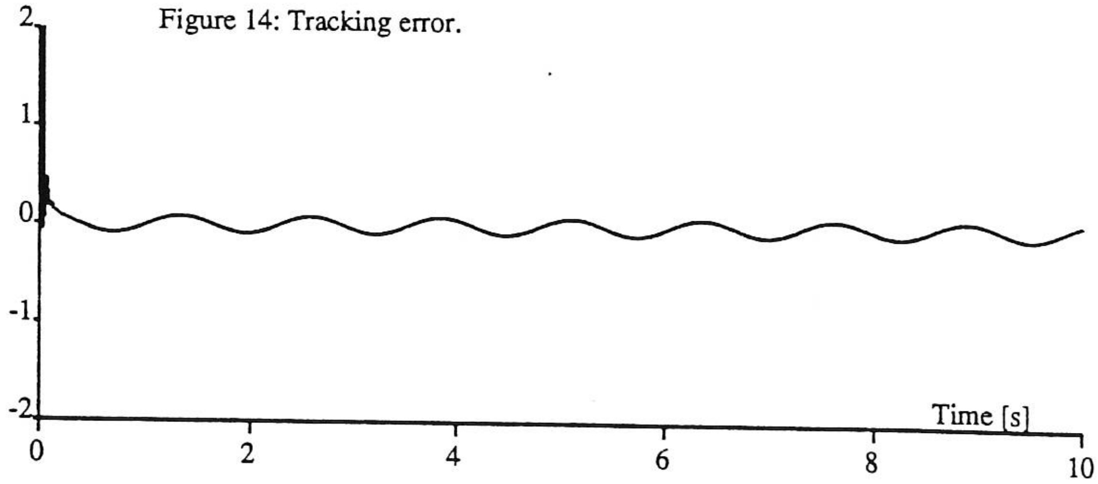


Figure 14: Tracking error.



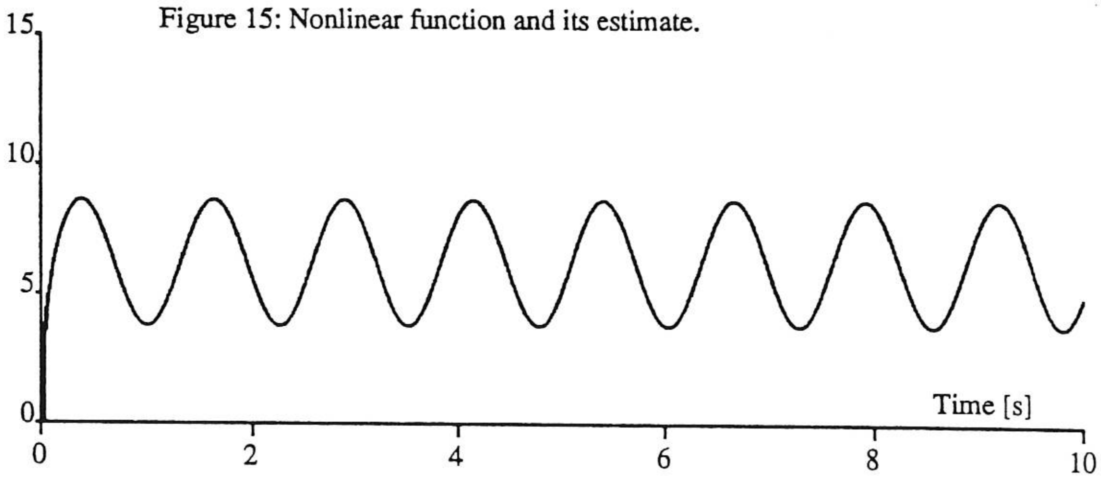


Figure 15: Nonlinear function and its estimate.

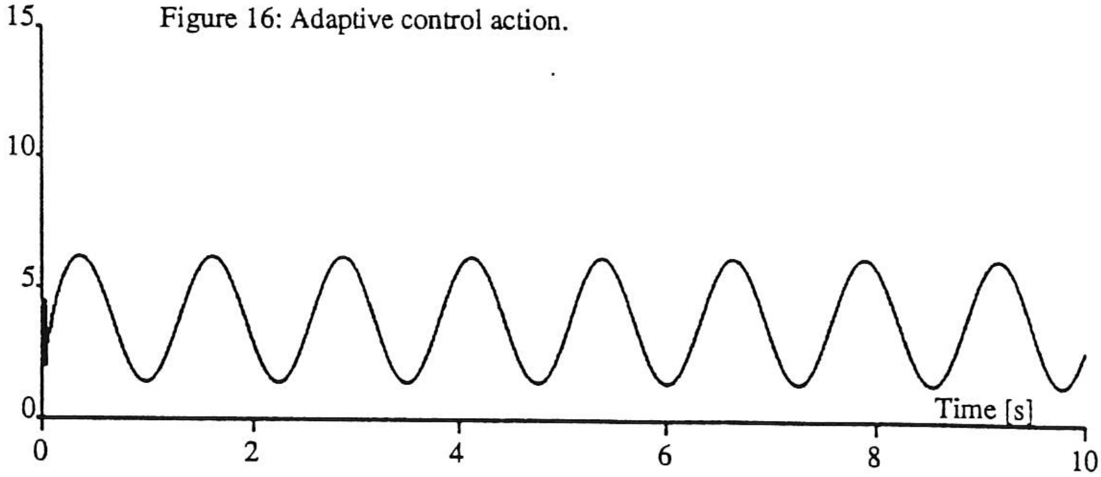


Figure 16: Adaptive control action.

Figure 17: Plant and model reference outputs.

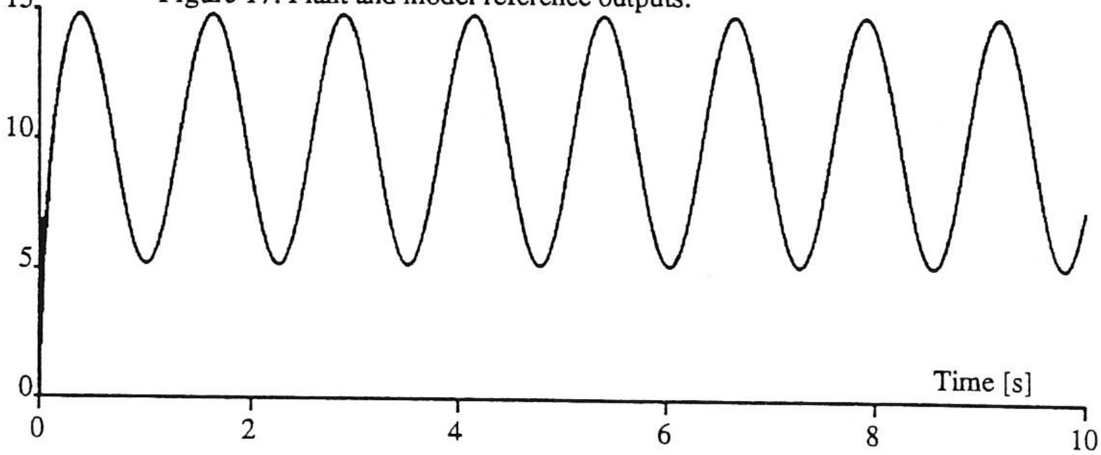
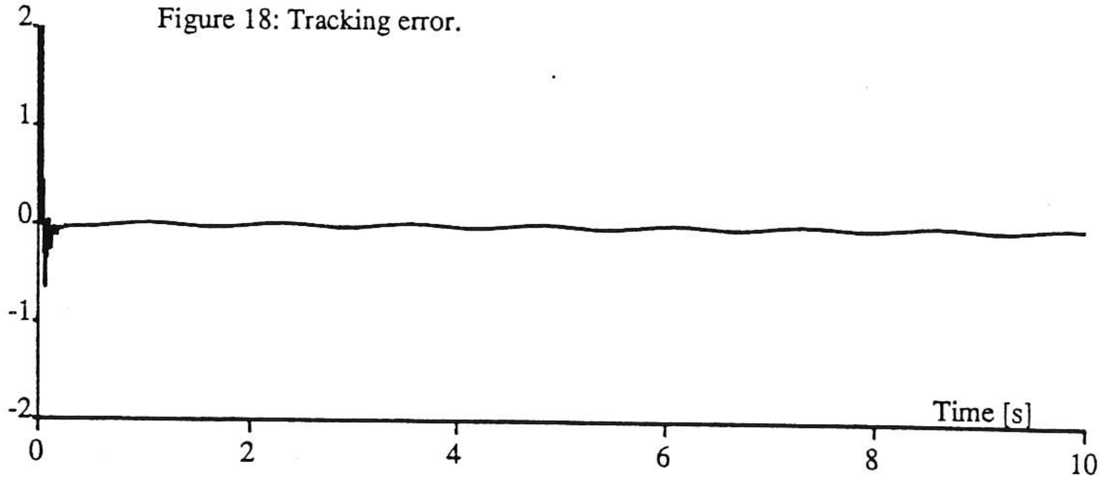


Figure 18: Tracking error.



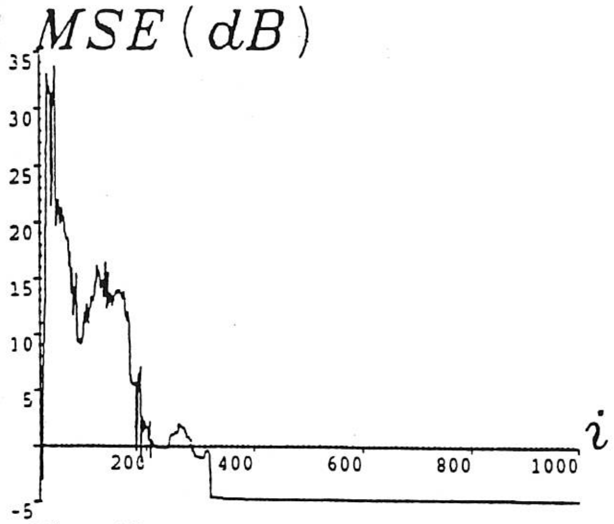


Fig. 12.

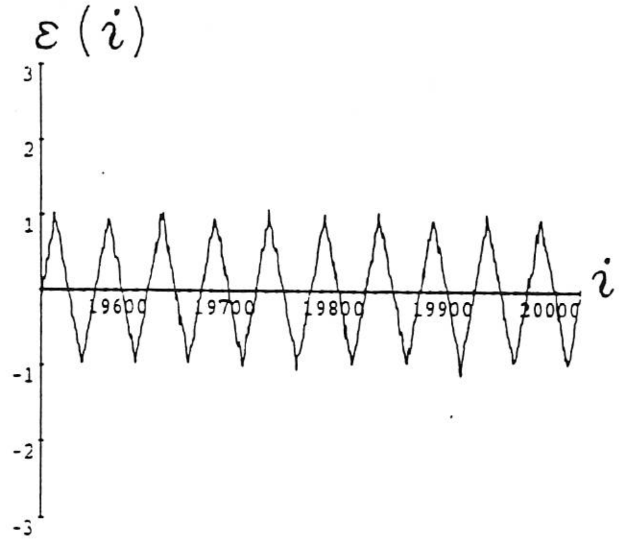


Fig. 13.

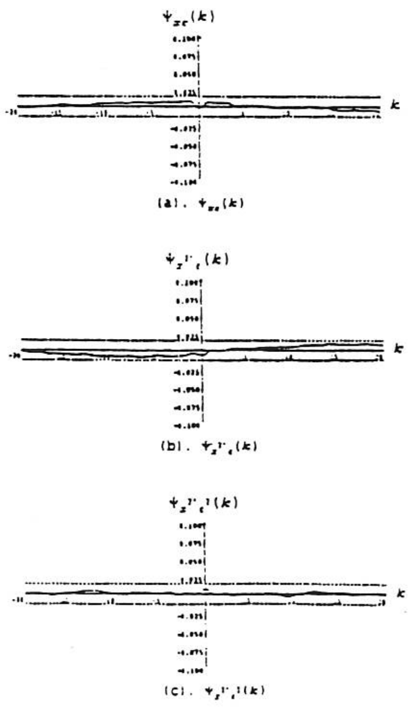


Fig. 14.

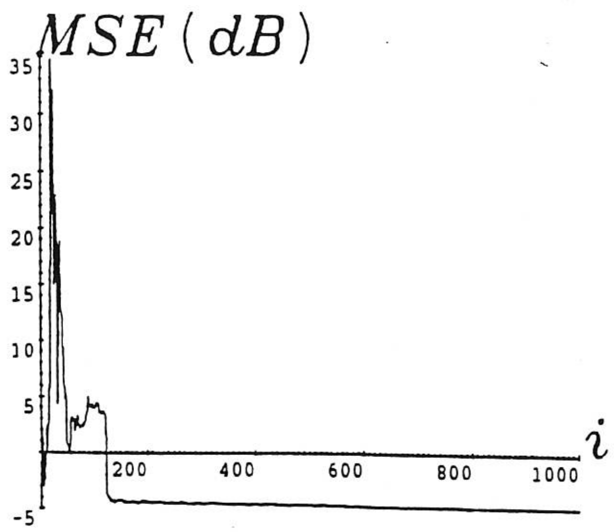


Fig. 15.

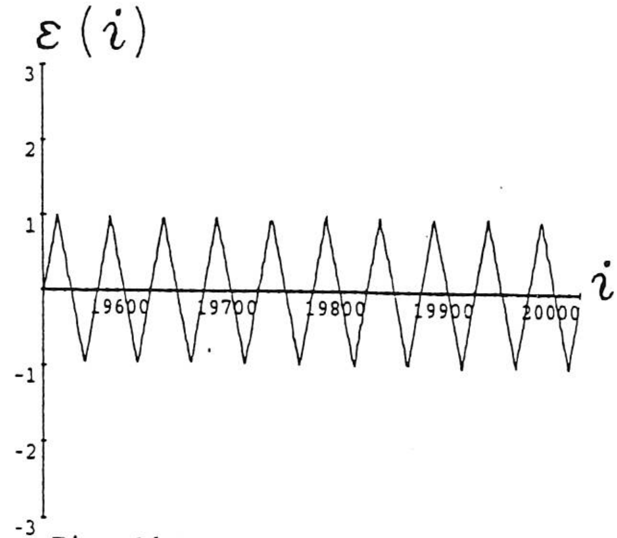


Fig. 16.

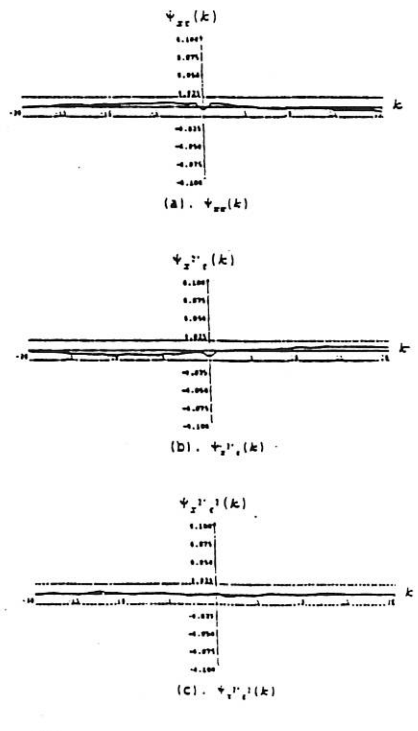


Fig. 17.