Model Following Control of SISO Nonlinear Systems using PID Neural Networks

Tamer A. Al-zohairy
AL Azhar University
Cairo-Eygpt

Khaled S. Salem
AL Azhar University
Cairo-Eygpt

ABSTRACT

In this paper we propose a direct adaptive neural network strategy for a class of unknown nonlinear single-input single-output systems. The adaptive controller is based on PID neural network. The PID neural network defines three neurons with the function of proportional (P), integral (I) and differential (D), into a neural network. PID neural network parameters are obtained using back propagation learning algorithm. Simulation results have been presented here in illustrate the effectiveness and accuracy of the proposed control strategy for tracking unknown single-input single-output (SISO) nonlinear discrete-time systems with and without long time delay.

General Terms

Adaptive control, PID Neural network control, SISO systems, time delayed systems.

Keywords

PID Controller, Neural network, Back Propagation algorithm, time delay systems, direct adaptive control.

1. INTRODUCTION

Because of the fast evolution in the control industry, the development of control problem becomes more complex in nonlinear systems, and difficult to be implemented, so the solution of this problem becomes one of the most focused research points [1-5].

PID controller is one of the most commonly controllers in the industrial closed loop control systems for its simple algorithm, good robustness and stability. But conventional PID controller has its disadvantage that it is not suitable for the control of long time-delay and nonlinear systems, in which the P, I and D parameters will have no changes after completion, resulting in the parameter variations of controlled objects cannot be tracked in real time environment and cannot meet increasing requirements of control quality in the production process [6-9].

Artificial neural networks (ANNs) have shown an excellent ability to model any nonlinear function to a desired degree of accuracy. Because of this property, they are suitable for the identification and control of nonlinear plants. From the different classes of networks, feedforward neural networks and particularly multi-layer perceptrons (MLPs) are the most frequently used for nonlinear control [1], [3] and [10].

Researches in control theory have shown that merging the properties of the PID with the neural networks will give the most correct presentation of the controller parameters, where three neurons in hidden layer for proportional (P), integral (I) and derivative (D) constitute parameters of the PID [1] ,[7], [11] and [12].

This paper will focus on the using of PID neural network for tracking unknown single-input single-output nonlinear discrete-time systems with and without long time delay. The aim of the PID neural network is to find the uppermost values for Proportional (P), Integral (I) and Differential (D) parameters using back propagation algorithm. Where a special activation functions are used for input, hidden and output layers of the network. As we will see, based on the systems under study, adding some modifications in the learning rules and activation functions are important in the learning process.

This paper is organized as follows: In section 2 the general form of the single-input single-output time delay systems is given. The architecture of the PID neural network is shown in section 3. Section 4 covers the control system model structure. Activation functions of network to move the knowledge between layers are given in section 5. Section 6 shows the learning rules of the network based on back propagation algorithm. Summary of the control algorithm for single-input single-output nonlinear systems is given in section 7. Simulation results through three examples are shown in section 8.

2. STATEMENT OF THE PROBLEM

Consider the unknown discrete time nonlinear SISO system be described by the difference equations:

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n+1), v(k), v(k-1), \dots, v(k-n+1)] \tag{1}$$

where the output $y(k+1)$ depends on the past values of the input and output at time $k = 0,1,2,\dots,n$.

Our objective here is to develop a method to design a model following adaptive control $v(k)$ for discrete time single-input single-output nonlinear systems given by equation (1) using PID neural Network such that system output $y(k+1)$ follows a known and bounded trajectory $r(k+1)$.

3. PID NEURAL NETWORK ARCHITECTURE

The traditional PID control combines 3 types of controllers into one controller, each one of these controllers will do special task in the control system. The proportional (P) feedback control signal to increase the speed of reaching the target values but still allows a non-zero steady-state error for the system, the integral (I) controller is used to eliminate the steady-state error, and the derivative (D) controller used to damp the dynamic response.

The PID neural network is constituted by integrating PID control law into the neural network. It is a multilayer neural network contains one hidden layer with 3 neurons, one input with 2 P-neurons and one output layer with one P-neuron as shown in figure.1. Each neuron in the hidden layer of the PID neural network is one of the three types of P-neuron (Proportional), I-neuron (Integral) and D-neuron (Derivative). Each neuron in the hidden layer has its own behavior and rules, as will be seen in section 5.

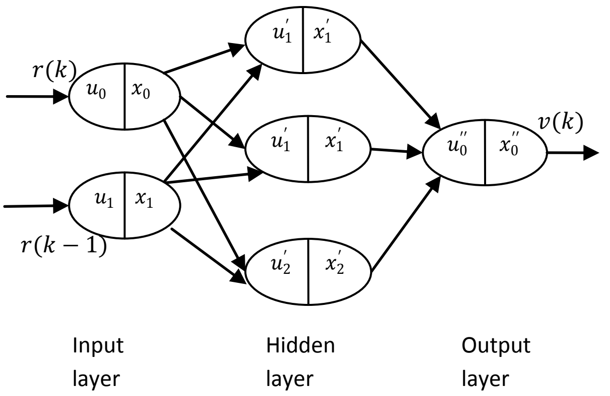

Figure.1: Architecture graph of PID neural network.

4. STRUCTURE OF THE CONTROL SYSTEM BASED ON PID NEURAL NETWORKS

The structure the controller based on PID neural networks is shown in Figure.2. The PID neural network controller receives its inputs from the reference model and the delayed value of the output system. The free parameters of PID neural network have to be adapted such that the control signal cause the output of the system follows the reference model signal. In this research, the backpropagation training algorithm is used to find PID neural network controller parameters.

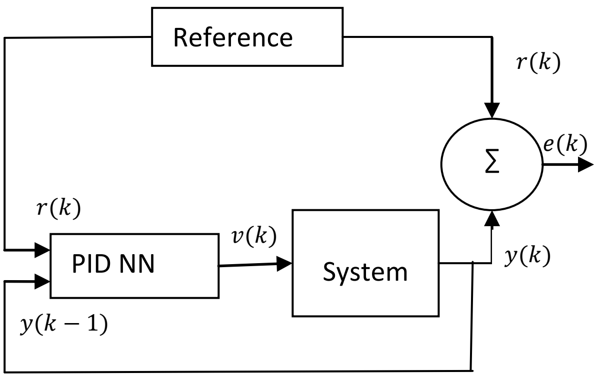

Figure.2. Structure of control system using PID neural networks.

5. PID NEURAL NETWORK DYNAMICS

As mentioned in section (3), the neurons of PID neural network are one of the three types: Proportional, Integral and Derivative. Following gives the detailed knowledge about the dynamics of each neuron:

1. The input layer which consists of 2 P-neurons receives its inputs through a linear transfer function given by:

$$u_1(k) = r(k),\ u_2(k) = y(k) \qquad (2)$$

and the following activation function is used to find the response of the input layer neurons :

$$x_i(k) = \begin{cases} -1 & if\ u_i(k) < -1 \\ u_i(k) & if -1 < u_i(k) < 1 \\ 1 & if\ u_i(k) > 1 \end{cases} \qquad (3)$$

2. The hidden layer which consists of 3 neurons P, I and D receives its input through a transfer function as follows:

$$u_j^{'}(k) = \sum w_{ji}\ (k) x_i(k) \qquad (4)$$

The following activation function is used to find the response of the P-neuron:

$$x_j^{'}(k) = \begin{cases} -1 & if\ \ u_j^{'}(k) < -1 \\ u_j^{'}(k) & if -1 < u_j^{'}(k) < 1 \\ 1 & if\ u_j^{'}(k) > 1 \end{cases} \qquad (5)$$

the following activation function is used to find the response of the I-neuron:

$$x_j^{'}(k) = \begin{cases} -1 & if\ u_j^{'}(k) < -1 \\ x_j^{'}(k-1) +\ u_j^{'}(k) & \\ & if -1 < x_j^{'}(k-1) +\ u_j^{'}(k) < 1 \\ 1 & if\ u_j^{'}(k) > 1 \end{cases}$$

(6)

and the following activation function is used to find the response of the D-neuron:

$$x_j^{'}(k) = \begin{cases} -1 & if\ u_j^{'}(k) < -1 \\ u_j^{'}(k) - u_j^{'}(k-1) & \\ & if -1 < u_j^{'}(k) - u_j^{'}(k-1) < 1 \\ 1 & if\ u_j^{'}(k) > 1 \end{cases}$$

(7)

3. The output layer which consists of one P-neuron receives its input through a transfer function as follows:

$$u^{''}_l(k) = \sum w_{lj}\ (k)\ x_j(k) \qquad (8)$$

and the following activation function is used to find the response of the output layer neurons :

$$x^{''}_l(k) = \begin{cases} \min() & if\ u^{''}_l(k) < \min() \\ u^{''}_l(k) & if\ \ -1 < u^{''}_l(k) < 1 \\ \max() & if\ u^{''}_l(k) > \max() \end{cases}$$

(9)

In activation function give by equation (9) the max and min values are put based on reference model and the system study to limit the control signal during the learning process.

The output from the PID neural network, $v(k)$, is given by the following equation:

$$v(k) = x_l^{''}(k). \qquad (10)$$

6. ADAPTATION RULES OF PID NEURAL NETWORK CONTROLLER

The objective of the learning process is to minimize the cost function given by:

$$J(k+1) = \frac{1}{2}(r(k+1) - y(k+1))^2 \qquad (11)$$

where the weight change in the network is calculated using gradient descent method. The following subsections give more details about the learning process.

6.1 Learning rules of the output neuron

In this case, the change in the output layer weights based on gradient descent method is given by:

$$\Delta w_{lj}^{'}(k+1) = -\mu \frac{\partial J(k+1)}{\partial w_{lj}^{'}(k+1)} \tag{12}$$

where

$$\frac{\partial J(k+1)}{\partial w_{lj}^{'}(k)} = \frac{\partial J(k+1)}{\partial y(k+1)} \cdot \frac{\partial y(k+1)}{\partial v(k)} \cdot \frac{\partial v(k)}{\partial x_l^{''}(k)} \cdot \frac{\partial x_l^{''}(k)}{\partial u_l^{''}(k)} \cdot \frac{\partial u_l^{''}(k)}{\partial w_{lj}^{'}(k)} \tag{13}$$

Form equations (9), (10) and (11) we get

$$\frac{\partial J(k+1)}{\partial y(k+1)} = -e(k+1) \quad ,$$

$$\frac{\partial v(k)}{\partial x_l^{''}(k)} \cdot \frac{\partial x_l^{''}(k)}{\partial u_l^{''}(k)} = 1,$$

$$\frac{\partial u_l^{''}(k)}{\partial w_{lj}^{'}(k)} = x_j^{'}(k) \ .$$

Because the nonlinear system with a single input single output is unknown, the derivative of y with respect to v takes the following form as given by [1]:

$$\frac{\partial y(k+1)}{\partial v(k)} \approx sgn \ \frac{\Delta y(k+1)}{\Delta v(k)} = sgn \frac{y(k+1)-y(k)}{v(v)-v(k-1)} \tag{14}$$

Then

$$\frac{\partial J(k+1)}{\partial w_{lj}^{'}(k)} = -e(k+1) . sgn \frac{y(k+1)-y(k)}{v(k)-v(k-1)} \ . 1 \ . 1 \ . x_j^{'}(k) \tag{15}$$

Substituting from equation (15) into equation (12) the weight changes of the output layer is given by:

$$\Delta w_{lj}^{'}(k+1) = \mu \ . e(k+1) \ . sgn \frac{y(k+1)-y(k)}{v(k+1)-v(k)} \ . x_j^{'}(k) \tag{16}$$

6.2 Learning rules of the hidden neurons

In this case, the change in the hidden layer weights based on gradient descent method is given by:

$$\Delta w_{ji}(k+1) = -\mu \frac{\partial J(k+1)}{\partial w_{ji}(k)} \tag{17}$$

where the derivative of J with respect to w is given by:

$$\frac{\partial J(k+1)}{\partial w_{ji}(k)} = \frac{\partial J(k+1)}{\partial y(k+1)} \cdot \frac{\partial y(k+1)}{\partial v(k)} \cdot \frac{\partial v(k)}{\partial x_l^{''}(k)} \cdot \frac{\partial x_l^{''}(k)}{\partial u_l^{''}(k)} \cdot$$

$$\frac{\partial u_l^{''}(k)}{\partial x_j^{'}(k)} \cdot \frac{\partial x_j^{'}(k)}{\partial u_j^{'}(k)} \cdot \frac{\partial u_j^{'}(k)}{\partial w_{ji}(k)} \tag{18}$$

Form equations (8) and (4) we get

$$\frac{\partial u_l^{''}(k)}{\partial x_j^{'}(k)} = w_{lj}^{'}(k) \quad ,$$

$$\frac{\partial u_j^{'}(k)}{\partial w_{ji}(k)} = x_i(k).$$

Because of the activation functions are different for hidden layer neurons so the derivative of $x_j^{'}$ with respect to $u_j^{'}$ is calculated as follows [1]:

$$\frac{\partial x_j^{'}(k)}{\partial u_j^{'}(k)} \approx sgn \frac{\Delta x_j^{'}(k)}{\Delta u_j^{'}(k)} = sgn \frac{x_j^{'}(k)-x_j^{'}(k-1)}{u_j^{'}(k)-u_j^{'}(k-1)} \tag{19}$$

Then

$$\frac{\partial J}{\partial w_{ji}} = -e(k+1) \ . \ sgn \frac{y(k+1)-y(k)}{v(k)-v(k-1)} \ . 1 \ . 1 \ . \ w_{lj}^{'}(k) \ . \ sgn \frac{x_j^{'}(k)-x_j^{'}(k-1)}{u_j^{'}(k)-u_j^{'}(k-1)} \ . \ x_i(k) \tag{20}$$

Substitute from equation (20) into equation (17) the learning rule for the hidden layer weights is given by:

$$\Delta w_{ji} = \mu . e(k+1) . sgn \frac{y(k+1)-y(k)}{v(k)-v(k-1)} \ . w_{lj}^{'}(k). \ sgn \frac{x_j^{'}(k)-x_j^{'}(k-1)}{u_j^{'}(k)-u_j^{'}(k-1)} \ . x_i(k) \tag{21}$$

7. PID NEURAL NETWORK ALGORITHM SUMMARY

The control for unknown SISO nonlinear systems using PID neural network is as follows:

- Initialize weights and parameters of the PID neural network.

  Start loop1:

- The input layer neurons of the PID neural network are set by equation (2).

  Start loop2:

- Calculate output of the input layer of the PID neural network using equation (3).
- The hidden layer inputs of the PID neural network are set by equation (4).
- Calculate output of the hidden layer neurons P, I, D of the neural network using equations (5), (6) and (7) respectively.
- The output layer inputs are set by equation (8).
- Calculate output of the PID neural network using equation (10).
- The error between system output and reference model is calculated using equation (11).
- Weights of the PID neural network are adapted using the learning rules given by equations (16) and (21).

  End loop2.
  End loop1.

8. EXPERIMENTAL RESULTS

Our experiments are done with 3 single-input single-output time delayed systems. The applications of the proposed PID neural network algorithm on the three systems are as follows:

Example 1:

Consider a SISO nonlinear given by:

$$y(k+1) = 0.468 \ y(k) + 0.532 \ v(k) \tag{22}$$

and assume that reference model is given by:

$$r(k) = 1 \tag{23}$$

The initial weights of the PID network are set randomly between [0, 1] and the learning rate is set to 0.006.

After studying the system (22) and according to the reference model, equation (9) is changed and takes the form:

$$x''_l(k) = \begin{cases} -1 & if\ u''_l(k) < -1 \\ u''_l(k) & if\ -1 < u''_l(k) \\ 1 & if\ u''_l(k) > 1 \end{cases}$$

After applying the proposed algorithm given in section 7 it was found that the output of system follow the reference given by equation (23) after 20 time steps as shown in the figure.3 where outputs of the system and reference model are by dotted line and continuous line respectively .

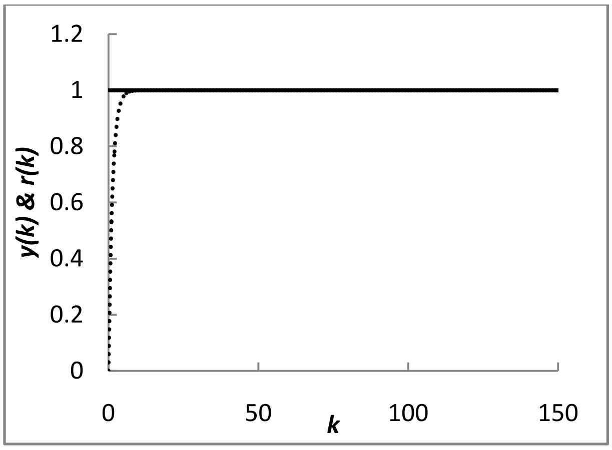

Figure.3. The reference signal $r(k)$and the system output $y(k)$ during the learning process based on the PID neural network.

Figure.4 shows that the error between the reference output and system output will become zero after 20 time steps through the learning process.

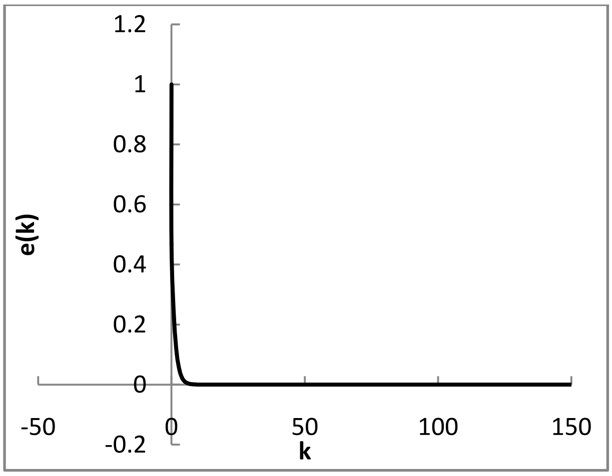

Figure.4. Error signal between the reference model $r(k)$ and system $y(k)$ during the learning process based on the PID neural network.

The control signal from the PID neural network during learning process is shown in figure 5.

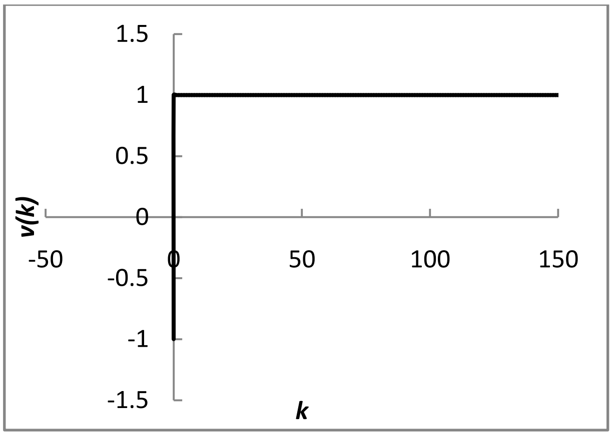

Figure.5. The control signal from the PID neural network during learning process.

The weights of the PID neural network after training are given by:

$w_{00} = -58095$, $w_{01} = 0.563585$,

$w_{10} = 0.193304$, $w_{11} = -102665$,

$w_{20} = 0.585009$, $w_{21} = 0.479873$.

for the hidden layer weights, and

$w'_{00} = 33.7421$,

$w'_{01} = 34.7199$,

$w'_{02} = 0.81976$.

for the output layer weights.

Example 2:

Consider a SISO nonlinear system given by:

$$y(k+1) = 0.868y(k) - 0.1746\, y(k-4) + 0.3066v(k-5) \quad (24)$$

and assume that the reference model is given by:

$$r(k) = 0.8 \quad (25)$$

The initial weights of the PID network are set randomly between [0, 1] and the learning rate is set to 0.5.

After studying the system (24) and according to the reference model, equation (9) is changed and takes the form:

$$x''_l\,(k = \begin{cases} -0.8 & if\ u''_l(k) < -0.8 \\ u''_l(k) & if\ -1 < u''_l(k) \\ 0.8 & if\ u''_l(k) > 0.8 \end{cases}$$

After applying the proposed algorithm given in section 7 it was found that the output of system follow the reference given by equation (25) after 90 time steps as shown in the figure.6, where outputs of the system and reference model are by dotted line and continuous line respectively .

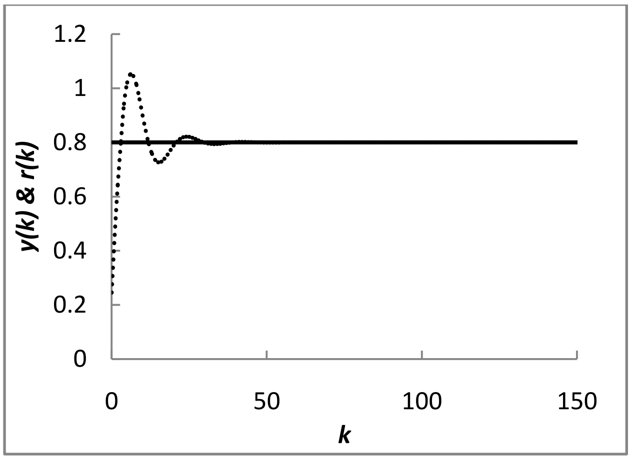

Figure.6. The reference signal $r(k)$and the system output $y(k)$ during the learning process based on the PID neural network.

Figure.7 shows that the error between the reference output and system output will become zero after 90 time steps through the learning process

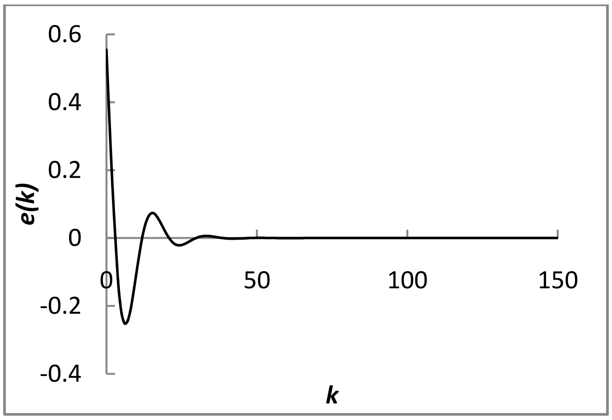

Figure.7. Error signal between the reference model $r(k)$ and system $y(k)$ during the learning process based on the PID neural network.

The control signal of the PID neural network during learning process is shown in figure.8.

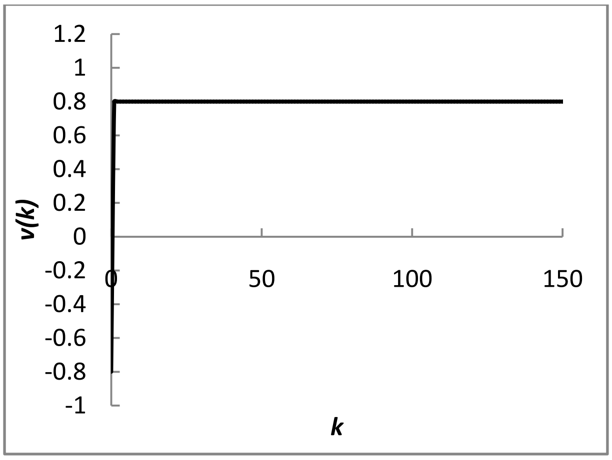

Figure.8. The control signal from the PID neural network during learning process.

The weights of the PID neural network after training are given by:

$w_{00} = 2285.58$, $w_{01} = 0.563585$,

$w_{10} = 0.193304$, $w_{11} = -339030$,

$w_{20} = 0.585009$, $w_{21} = 0.479873$.

for the hidden layer weights, and

$w'_{00} = 6687.09$,

$w'_{01} = 7060.19$,

$w'_{02} = 0.80177$.

for the output layer weights.

Example 3:

In this case the system is described by equation

$$y(k+1) = \frac{y(k)}{1+y^2(k)} + v^3(k) \qquad (26)$$

And the reference model is given by:

$$r(k+1) = 0.6 * r(k) + \sin\left(\frac{2\pi k}{25}\right) + \sin\left(\frac{2\pi k}{10}\right) \qquad (27)$$

The initial weights of the PID network are set randomly between [0, 1] and the learning rate is set to 0.007.

After studying the system (26) and according to the reference model, equation (9) is changed and takes the form:

$$x''_l(k) = \begin{cases} -4 & if\ u''_l(k) < -4 \\ u''_l(k) & if\ -1 < u''_l(k) \\ 4 & if\ u''_l(k) > 4 \end{cases}$$

After applying the proposed algorithm given in section 7 it was found that the output of system follow the reference given by equation (27) after 25 time steps as shown in the figure.9, where outputs of the system and reference model are by dotted line and continuous line respectively .

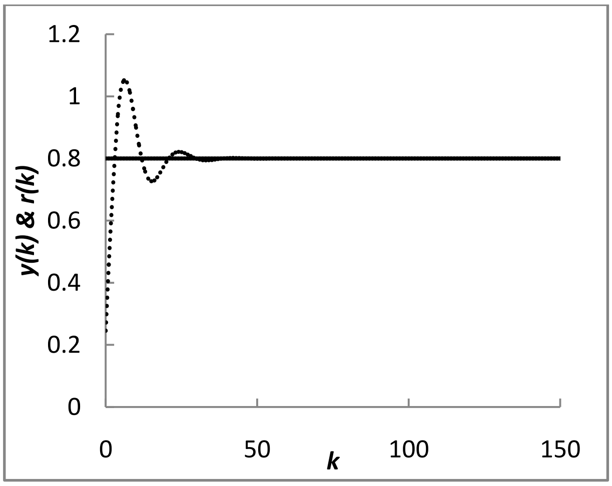

Figure.9. The reference signal $r(k)$and the system output $y(k)$ during the learning process based on the PID neural network.

Figure.10 shows that the error between the reference output and system output will become zero after 25 time steps through the learning process.

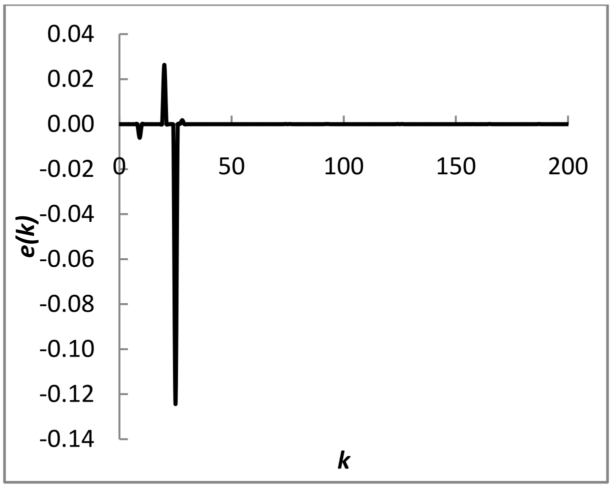

Figure.10. Error signal between the reference model $r(k)$ and system $y(k)$ during the learning process based on the PID neural network.

The control signal of the PID neural network during learning process is shown in figure.11.

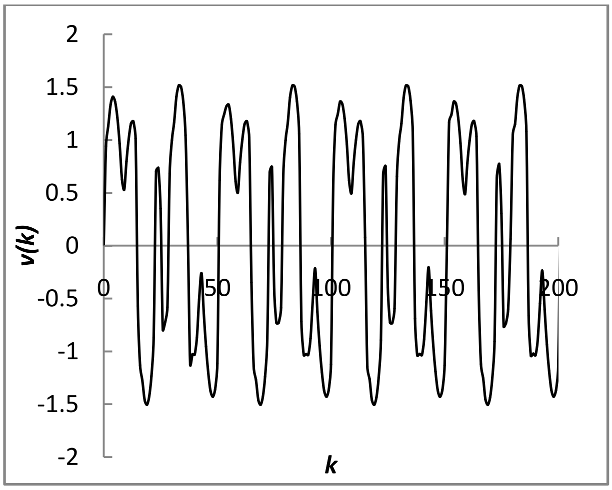

Figure.11. The control signal from the PID neural network during learning process.

9. CONCLUSION

In this paper, we have proposed a model following adaptive controller for a class of discrete SISO nonlinear systems. in the proposed method PID neural networks are used for online control of unknown nonlinear SISO system with a long time delay. Simulation results on 2 different SISO systems have shown that the PID neural networks can drive the system outputs to the desired reference with a satisfactory perfo-rmance and in a short time steps. The general conclusion therefore, is that PID neural network should be preferred for online control problems because: 1) simple structure, 2) minimum convergence time and 3) suitable for the control of long time-delay nonlinear systems.

10. FUTURE WORK

The application of the PID neural network on different types of multi-input multi-output systems is the main interest in the next paper.

11. REFERENCES

[1] HuailinShu.,Yoyguo Pi.2000. PID neural networks for time delayed systems.

[2] Fang Ding, Yun Zhang. 2011. The GA-BP Algorithm Based controller of PID Neural Network and Its Application.

[3] Yang Zhi-gang, Qian Jun-lei. 2013. PID Neural Network Adaptive Predictive Control for Long Time Delay System.

[4] QiushuangLIU,Xiaoli XU. 2010. PID neural network control of a membrane structure inflation system.

[5] AsheshaPanbude, Manish Sharma.2015. Implementation of Neural Network for PID Controller.

[6] Woo-yongHan,Jin-WookHan,Chang-goo Lee. 1999. Development of a Self-tuning PID controller based on Neural Network for Nonlinear Systems.

[7] Liu Luoren, LuoJinling. 2011. Research of PID Control Algorithm Based on Neural Network.

[8] B. Guo , H. Liu , Z. Luo , F. Wang. 2009. Adaptive PID Controller Based on BP Neural Network.

[9] Wei Lu, JianhuaYang ,Xiaodong Liu.2012.The PID controller based on the artificial neural network and the differential evolution algorithm.

[10] T. A. A. Alzohairy. 2012. Neural Internal Model Control for tracking Unknown Nonaffine Nonlinear Discrete-time Systems under External Disturbances.

[11] Qun Yu, HuaShu,HuailinShu. 2008. Development of PID neural network control system with virtual instrument.

[12] DilsadEngin, Mustafa Engin. 2013. Auto-tuning of PID parameters with programmable logic controller.