www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 5 Issue 8 August 2016 Page No. 17673-17678

Adaptive control for MIMO nonlinear systems based on PID neural networks

Tamer A. Al Zohairy [1], Khaled S. Salem [2]

[1] Mathematics Department
Faculty of since AL Azhar University
Cairo-Egypt
tamer.alzohairy@gmail.com

[2]Mathematics Department
Faculty of since AL Azhar University
Cairo-Egypt
khaled.salem@msn.com

Abstract: ***In this paper a real time control technique for a nonlinear discrete time Multi-input Multi-output systems is presented. The proposed technique is based on the combination of PID controller properties and neural networks characteristic of self-learning and flexibility of function presentation. The standard back propagation (BP) algorithm is used to find parameters of the PID neural network controller. The suggested technique modifies the architecture used in SISO to fit with MIMO systems. The experimental results are presented toward the end of the paper to show the effectiveness of the proposed technique.***

Keywords: Adaptive Control, PID, neural networks, back propagation algorithm, Multi-input multi-output systems.

1. Introduction

In industrial production systems, the most leading challenge that faces the environments is the control problem. There are many artificial inelegance techniques are used to solve control problem this is because of their robustness, good dynamic response, fast rise time, precision dynamic and sufficient convergence speed; like fuzzy logics, genetic algorithm and neural networks [1-5].

One of the important control techniques is PID model which is widely used in solving the control of online systems. This is due to the characteristics of self-learning, self-organization and self-adaptation as it enhances the tracking performance of the system and automatically used to identify the parameters of the controlled system and adjust them according to system changes. Taking the gain of each element of Proportional (P), Integral (I) and Differential (D) will smooth function and find the optimal performance of the system outputs [6-8].

The use of neural networks in the control systems appears to offer new and promising developments toward better performance and can solve several problems in the control systems field due to fast adaptability and approximation capability [9]. The dynamic element embedded in neuron structure can create the dynamics of any control and adapt its changes and then the system changes [4].Also neural networks can compensate for the effects of any desired degree of nonlinearities and system uncertainties, so that the stability, convergence of the controlled system can be improved [10,11].

In [12] adding adaptive mechanism such as neural networks to the PID controller would result the correct parameters. Also in [13, 15] it had shown that parameters of the PID neural networks controller are self-adjusted in time to achieve the best performance, and suitable to be used in complex systems, nonlinear systems, systems with disturbance and dynamic systems. In [16-18] PID neural network shows great ability and accurate results in solving the Single-input Single-output problem by making use of a neuron's self learning ability, and automatically adjust the parameter of the PID and make them adapt to the changes in the structure and parameters of the controlled objects.

Most of the results reported in the above literatures of the adaptive control of nonlinear dynamical system are related to single-input single output systems. But most practical systems have multiple inputs multiple outputs (MIMO) .Then our interest in this paper is to study the problem controlling nonlinear multi input multi output systems using PID neural network controller.

In our paper we will extend the method used in [16] which solves the Single-input Single-output control problem to be suitable for multi variables system. The extension to the structure of the PID neural network and its learning algorithm is done to fit with the Multi-input Multi-output model systems.

This paper is organized as follows: In section 2 the statement of problem is given. The architecture of the PID neural network for multi input multi output is shown in section 3. Section 4 covers the control system model. Dynamics of the PID controller and the learning algorithm are given in section 5. Summary of the control algorithm of multi input multi output nonlinear systems is given in section 6. Section 7 covers a comparison study with other method. Simulation results through two examples are shown in section 7.

DOI: 10.18535/ijecs/v5i8.44

2. Statement of the problem

Consider the discrete time unknown nonlinear MIMO system with m inputs and m outputs, is described by the following difference equation:

$$\mathrm{y(k+1) = f[y(k),\ y(k-1), \dots,\ y(k-t+1),\ v(k),\ v(v-1),\ \dots,\ v(k-t+1)]} \quad (1)$$

where the output vector at time k is presented by $\mathrm{y(k) = [y_1(k),\ y_2(k), \dots,\ y_m(k)] \in R^m}$, and the input vector at time k is presented by $\mathrm{v(k) = [v_1(k),\ v_2(k), \dots,\ v_m(k)] \in R^m}$. The objective in this paper is to develop an adaptive control method to adapt $\mathrm{v_i(k)}$of discrete unknown multi-input multi-output nonlinear systems given by equation (1) such that system output $\mathrm{y_i(k)}$ follows a known and bounded trajectory $\mathrm{r_i(k+1)}$using PID neural Network, where $\mathrm{r(k) = [r_1(k), r_2(k), \dots,\ r_m(k)]}$.

3. Architecture of the PID neural network

As known in [16] the traditional PID contains 3 controllers merged together to present one controller. The three controllers proportional (P), integral (I), derivative (D) make a specific task in the main controller; by increasing the speed of reaching the target values, eliminating the steady-state error, and damping the dynamic response respectively. The PID neural network has the following Architecture shown in figure 1.

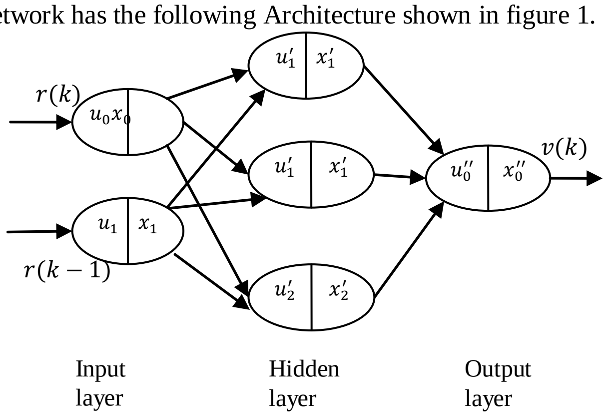

Figure 1: Architecture of PID neural network

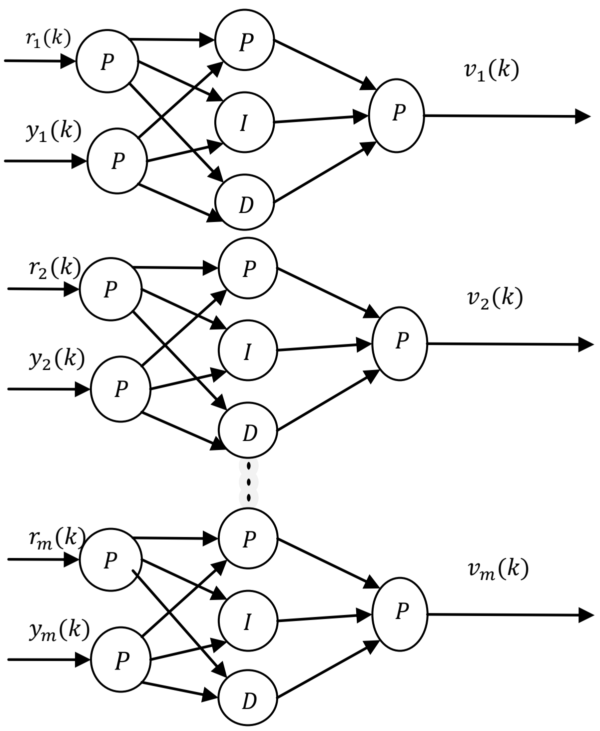

Figure 2: Architecture graph of PID neural network for multi-input multi-output system

In case of single input single output systems one PID neural network controller is used to solve the control problem. In this paper, to solve the multi input multi output systems control problem given in section 2, s PID neural network controllers are needed to find m control inputs. Each PID controller is considered as a sub-network. Each sub-network contains one hidden layer with 3 neurons one of each types of P-neuron (Proportional), I-neuron (Integral) and D-neuron (Derivative), one input layer with 2 P-neurons and one output layer with one P-neuron as shown in figure.2.

4. Control system model

The structure of PID neural networks controller for multi input multi output is shown in Figure.3. As mentioned in section 3 the PID neural network controller contains m sub-networks .Every sub-network receive its inputs from the reference model and the delayed value of the output system that is relevance to it. The control signal of each sub-system is considered as a new input of the multi input multi output system. The free parameters of each sub-PID neural network will be adapt the overall control signals causing that outputs of the system follows the reference model signals. In this paper, Backpropagation algorithm is used to find the parameters of each sub PID neural network for multi input multi output control.

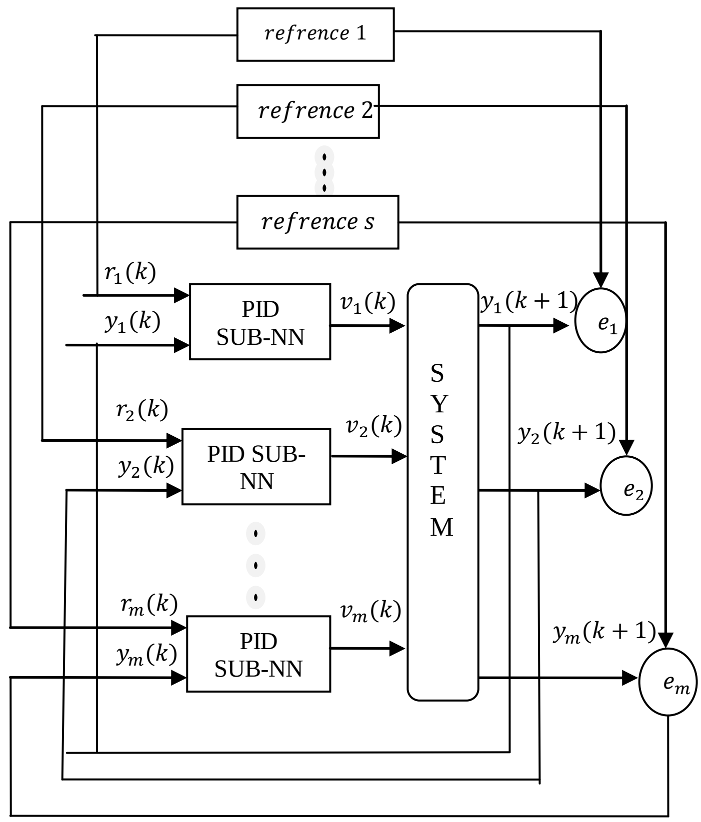

Figure 3: Structure of control system for Multi-input Multi-out using PID neural networks

5. PID Neural networks dynamics and learning algorithm.

In the controller represented by sub PID neural network, given in section 3; every sub network will have one of the three types: Proportional, Integral and Derivative. Following gives the detailed knowledge about the dynamics and learning mechanism for each type of neurons in m sub-networks:

5.1 PID Neural networks dynamics

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

$$u_{s1}(k) = u_s(k),\ u_{s2}(k) = y_s(k) \quad \text{where } s = 1,2,3,\dots,m \tag{2}$$

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

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

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

$$u'_{sj}(k) = \sum w_{sji}(k) x_{si}(k) \tag{4}$$

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

$$x'_{sj}(k) = \begin{cases} -1 & \text{if } u'_{sj}(k) < -1 \\ u'_{sj}(k) & \text{if } -1 < u'_{sj}(k) < 1 \\ 1 & \text{if } u'_{sj}(k) > 1 \end{cases} \tag{5}$$

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

$$x'_{sj}(k) = \begin{cases} -1 & \text{if } u'_{sj}(k) < -1 \\ x'_{sj}(k-1) + u'_{sj}(k) & \text{if } -1 < x'_{sj}(k-1) + u'_{sj}(k) < 1 \\ 1 & \text{if } u'_{sj}(k) > 1 \end{cases} \tag{6}$$

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

$$x'_j(k) = \begin{cases} -1 & \text{if } u'_{sj}(k) < -1 \\ u'_{sj}(k) - u'_{sj}(k-1) & \text{if } -1 < u'_{sj}(k) - u'_{sj}(k-1) < 1 \\ 1 & \text{if } u'_{sj}(k) > 1 \end{cases} \tag{7}$$

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

$$u''_{sl}(k) = \sum w_{slj}(k)\, x_{sj}(k) \tag{8}$$

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

$$x''_{sl}(k) = \begin{cases} \min() & \text{if } u''_{sl}(k) < \min() \\ u''_{sl}(k) & \text{if } -1 < u''_{sl}(k) < 1 \\ \max() & \text{if } u''_{sl}(k) > \max() \end{cases} \tag{9}$$

In activation function give by equation (9) the max and min values are put based on reference model related to the sub-network and the system to adapt the control signal for the inputs during learning process.

The output from the PID multi–networks $v_s(k)$, is given by the following equation:

$$v_s(k) = x''_{sl}(k). \tag{10}$$

5.2 PID Neural networks learning algorithm

In this paper the back propagation algorithm used to train parameters for each sub network. The objective of multi-input multi-output PID neural network is to minimize cost function given by:

$$J_s(k+1) = \frac{1}{2}\left(r_s(k+1) - y_s(k+1)\right)^2 \tag{11}$$

Such that the weight change in the network is calculated using gradient descent method. The following gives details about the learning process for each layer of the sub-network.

5.1.1 Learning rules for the parameters of the output layer neuron:

The PID neural network weights of output layer based on gradient descent method are given by:

$$\Delta w'_{slj}(k+1) = -\mu \frac{\partial J_s(k+1)}{\partial w'_{slj}(k+1)} \tag{12}$$

where

$$\frac{\partial J_s(k+1)}{\partial w'_{slj}(k)} = \frac{\partial J_s(k+1)}{\partial y_s(k+1)} \cdot \frac{\partial y_s(k+1)}{\partial v_s(k)} \cdot \frac{\partial v_s(k)}{\partial x''_{sl}(k)} \cdot \frac{\partial x''_{sl}(k)}{\partial u''_{sl}(k)} \cdot \frac{\partial u''_{sl}(k)}{\partial w'_{slj}(k)} \tag{13}$$

from equations (9), (10) and (11) we calculate the partial derivatives

$$\frac{\partial J_s(k+1)}{\partial y_s(k+1)} = -e_s(k+1) \quad ,$$

$$\frac{\partial v_s(k)}{\partial x''_{sl}(k)} \cdot \frac{\partial x''_{sl}(k)}{\partial u''_{sl}(k)} = 1,$$

$$\frac{\partial u''_{sl}(k)}{\partial w'_{slj}(k)} = x'_{sj}(k) .$$

Because the multi-input multi-output nonlinear system is unknown, the derivative of y_s with respect to v_s takes the following form as:

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

Then

$$\frac{\partial J_s(k+1)}{\partial w'_{slj}(k)} = -e_s(k+1) . sgn \frac{y_s(k+1) - y_s(k)}{v_s(k) - v_s(k-1)} . 1 . 1 . x'_{sj}(k) \tag{15}$$

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

$$\Delta w'_{slj}(k+1) = \mu . e_s(k+1) . sgn \frac{y_s(k+1) - y_s(k)}{v_s(k+1) - v_s(k)} . x'_{sj}(k) \tag{16}$$

5.1.2 Learning rules for the parameters of the hidden neurons:

The PID neural network weights of the hidden layer are based on gradient descent method given by:

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

where the derivative of J_s with respect to w_s is given by:

$$\frac{\partial J_s(k+1)}{\partial w_{sji}(k)} = \frac{\partial J_s(k+1)}{\partial y_s(k+1)} \cdot \frac{\partial y_s(k+1)}{\partial v_s(k)} \cdot \frac{\partial v_s(k)}{\partial x''_{sl}(k)} \cdot \frac{\partial x''_{sl}(k)}{\partial u''_{sl}(k)} \cdot \frac{\partial u''_{sl}(k)}{\partial x'_{sj}(k)} \cdot \frac{\partial x'_{sj}(k)}{\partial u'_{sj}(k)} \cdot \frac{\partial u'_{sj}(k)}{\partial w_{sji}(k)} \tag{18}$$

Form equations (8) and (4) we calculate the partial derivatives

$$\frac{\partial u''_{sl}(k)}{\partial x'_{sj}(k)} = w'_{slj}(k) \quad ,$$

$$\frac{\partial su'_j(k)}{\partial sw_{ji}(k)} = x_{si}(k).$$

Because of the activation function in each sub network is different for hidden layer neurons so the derivative of x'_{sj} with respect to u'_{sj} is calculated by:

$$\frac{\partial x'_{sj}(k)}{\partial u'_{sj}(k)} \approx sgn\frac{\Delta x'_{sj}(k)}{\Delta u'_{sj}(k)} = sgn\frac{x'_{sj}(k)-x'_{sj}(k-1)}{u'_{sj}(k)-u'_{sj}(k-1)} \quad (19)$$

Then

$$\frac{\partial J_s}{\partial w_{sji}} = -e_s(k+1)\,.\; sgn\frac{y_s(k+1)-y_s(k)}{v_s(k)-v_s(k-1)}\,.1\,.1\,.\; w'_{slj}(k).\; sgn\frac{x'_{sj}(k)-x'_{sj}(k-1)}{u'_{sj}(k)-u'_{sj}(k-1)}\,.\; x_{si}(k) \quad (20)$$

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

$$\Delta w_{sji} = \mu.\, e_s(k+1).\, sgn\frac{y_s(k+1)-y_s(k)}{v_s(k)-v_s(k-1)}\,.\, w'_{slj}(k).\; sgn\frac{x'_{sj}(k)-x'_{sj}(k-1)}{u'_{sj}(k)-u'_{sj}(k-1)}\,.\, x_{si}(k) \quad (21)$$

6. Algorithm Summary

The summary of the algorithm for MIMO system based on PID neural network is as following :

1.Initialize weights and parameters of all sub PID neural network.

Start loop1(Sample loop):

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

Start loop2(Sub-network loop):

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

End loop2(Sub-network loop).

End loop1(Sample loop).

7. Simulation results

The simulation is accomplished on two discrete time nonlinear multi-input multi-output systems to verify the proposed control strategy.

Example 1:

Consider the following multi-input multi-output system of order 2 described by the following equation [19]:

$$y_1(k+1) = \frac{1}{(1+y_1(k))^2}(0.8y_1(k) + v_1(k-1) + 0.2v_2(k-2))$$
$$y_2(k+1) = \frac{1}{(1+y_2(k))^2}(0.9y_2(k) + 0.3v_1(k-2) + 0.2v_2(k-1)) \quad (22)$$

and the reference model given by:

$$r_1(k) = 0.7$$
$$r_2(k) = 0.4 \quad (23)$$

The proposed PID multi input multi output algorithm is applied on the given system based on the architecture graph given in section 4. For the system given in example 1, two sub-networks are used. The weights of the sub-networks are initiated randomly with values between [0, 1], learning rate is set to 0.6.

After applying the suggested algorithm given in section 5 the actual outputs of the system will follow the corresponding references given in equation (23) after 55 steps as shown in figures 4 and 5 where the outputs of the system are in dotted line and references are continuous lines.

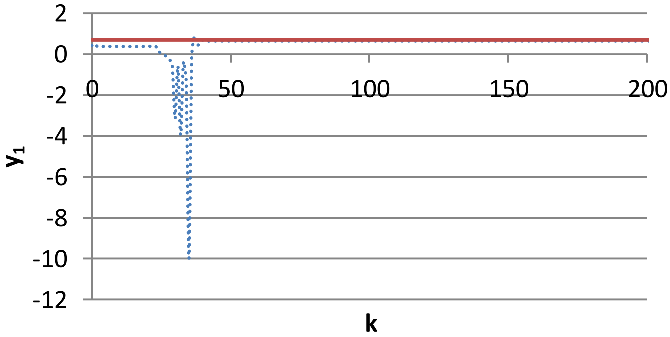

Figure 4: The system response of y_1 (example1).

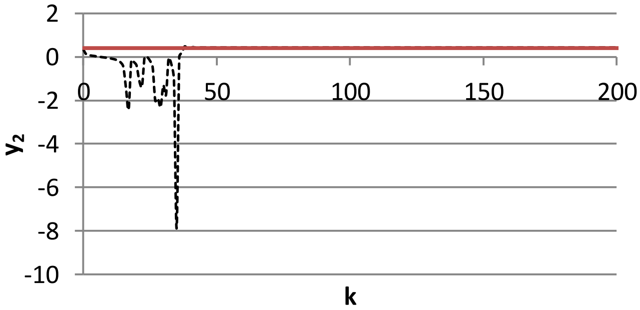

Figure 5: The system response of y_2(example 1).

Example 2:

Consider the following multi-input multi-output system of order 3 described by the following equations [19]:

$$y_1(k+1) = 0.4y_1(k) + \frac{v_1(k)}{1+v_1^2(k)} + 0.2v_1^3(k) + 0.5v_2(k) + 0.3y_2(k)$$
$$y_2(k+1) = 0.2y_2(k) + \frac{v_2(k)}{1+v_2^2(k)} + 0.4v_2^3(k) + 0.2v_1(k) + 0.3y_3(k)$$
$$y_3(k+1) = 0.2y_3(k) + \frac{v_3(k)}{1+v_3^2(k)} + 0.3v_3^3(k) + 0.3v_2(k) + 0.3y_1(k) \quad (24)$$

and the reference model given by:

$$r_1(k) = 0.7$$

$$r_2(k) = 0.5$$
$$r_2(k) = 0.6 \quad (25)$$

The proposed PID multi input multi output algorithm is applied on the given system based on the architecture graph given in section 4. For the system given in example 1, three sub-networks are used. The weights of the sub-networks are initiated randomly with values between [0, 1] and Learning rate is set to 0.3.

After applying the proposed algorithm given in section 5 the outputs of the system will follow the corresponding references given in (25) respectively after 88 steps as shown in figures 6, 7 and 8 where the outputs of the system are in dotted lines and references are continuous lines.

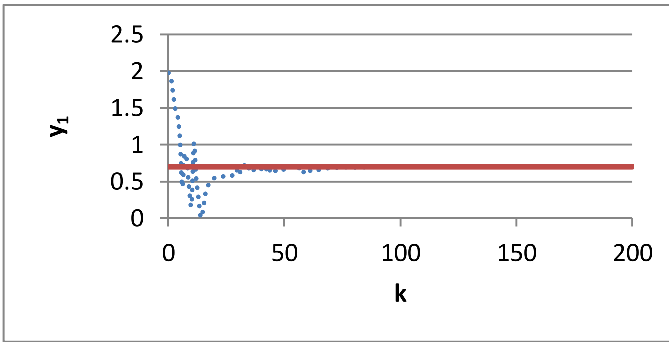

Figure. 6: The system response of y_1 of the system given in example 2.

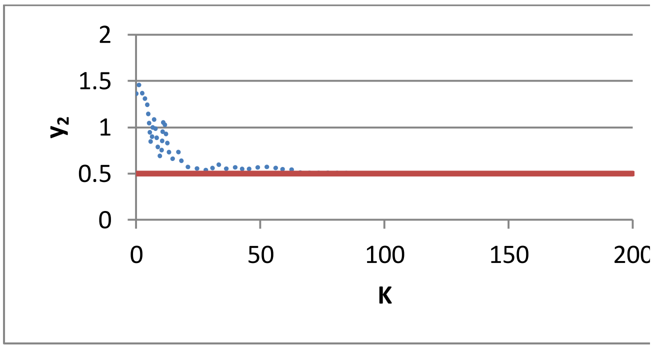

Figure. 7: The system response of y_2 of the system given in example 2.

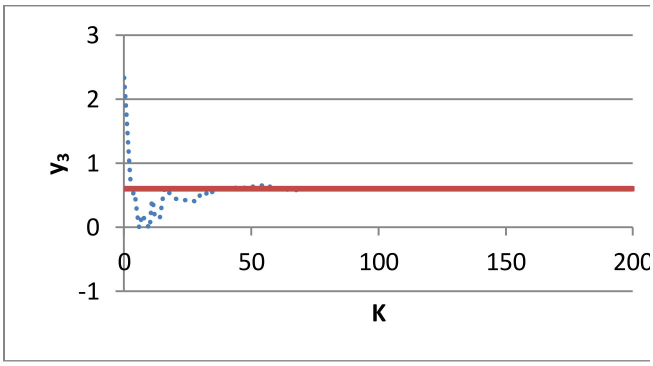

Figure. 8: The system response of y_3 of the system given in example 2.

8. Comparisons with other works

This section gives a comparison study between the proposed method in this paper and that is given in [19] to solve the same control problem.

From the results obtained in section 7, example 1 and that is given in [19] to solve the same control problem, it was found that the method in [19] reaches the reference model after 200 steps while the proposed method in this paper reaches the references after 55 steps.

Also from the results obtained in section 7, example 2 and that is given in [19] to solve the same control problem, it was found that the method in [19] reaches the reference model after 150 steps while the proposed method in this paper reaches the references after88 steps.

From the results and the comparison study it was found that the PID neural network used in [19] is quite slow compared with the proposed method used in this paper to solve the same problem which make the proposed method more suitable and reliable for the real time problems.

9. Conclusion

A direct adaptive PID neural network controller for the class of discrete time nonlinear multi-input multi output systems has been proposed in this paper. The PID neural network controller is divided into sub-networks based on the degree of the multi-input multi-output system used. Each sub-network is the same as that is used for single-input single-output systems. The PID controller parameters are updated on line. In each time step back propagation algorithm with the gradient descent method is used to update sub-network weights. Experimental results done on different multi-input multi output systems have shown that, using the proposed controller the system outputs can reach the to its desired references in less time and more efficiency. Comparison study with other work has shown that the proposed method gives faster and accurate results due to its simplicity, fast adaptation for the discrete nonlinear multi-input multi output systems.

10. REFERENCES

[1] J.G. Lai, H. Zhou, W.S. Hu , "A New Adaptive Fuzzy PID Controller Method And Its Application" in FCBRM , 2016.

[2] Ivan Sekaj, Martin Foltin ,Michal Gonos , "Genetic Algorithm Based Adaptive Control Of An Electromechanical MIMO System", 2016.

[3] O. P. Malik, "Amalgamation of adaptive control and AI techniques applications to generator excitation control", 2004.

[4] Alireza Yazdizadeh , Arash Mehrafrooz , Kasra Dastjani Farahani , Roohollah Barzaminix, "Adaptive PID controller design with application to nonlinear water level" in NEKA Power Plant, 2004.

[5] T. A. A. Alzohairy , "Neural Internal Model Control for tracking Unknown Non affine Nonlinear Discrete-time Systems under External Disturbances", 2012.

[6] B.G.D. Achintha Madhusank, W.R. de Mel, "ARTIFICIAL NEURAL NETWORK-BASED NONLINEAR DYNAMIC MODELLING OF THE TWIN-ROTOR MIMO SYSTEM", 2011.

[7] Jun Kang , Wenjun Meng, Ajith Abraham, Hongbo Liu, "An adaptive PID neural network for complex nonlinear system control", 2014.

[8] Upendra Kumar; Sachin Mahajan; Shilpesh Sinha; Manjeet Kaur, “Adaptive PID Control System-A Review”, 2016.
[9] Xiong Luo ; Jennie Si, “Stability of direct heuristic dynamic programming for nonlinear tracking control using PID neural network” ,2013.
[10] Chih-Min Lin ;Chun-Fei Hsu, “ Neural-Network Hybrid Control for Antilock Braking Systems”, 2003.
[11] Shuzhi Sam Ge, , Jin Zhang, Tong Heng Lee, “Adaptive Neural Network Control for a Class of MIMO Nonlinear Systems With Disturbances in Discrete-Time”, 2004.
[12] Hao Zhang, P.R. Ukrainetz, R.T. Burton, P.N. Nikiforuk, “Implementation of Neural Network Approach In MIMO Electro hydraulic Servosystem Control” ,1996.
[13] Wei SU, “A Model Reference-Based Adaptive PID Controller for Robot Motion Control of Not Explicitly Known Systems” , 2007.
[14] AgusTrisanto , “PID Controllers Using neural network For Multi-Input Multi-Output Magnetic Levitation System” , 2007.
[15] Yan Zhang, Yanbo Li, Liping Yang, and Peng Yang,2011,Nonlinear PID-Predictive Control for Multivariable Nonlinear System
[16] Qiushuang LIU, Xiaoli XU , “PID neural network control of a membrane structure inflation system” , 2010.
[17] Huailin Shu , Youguo Pi ,2000,PID neural networks for time-delay systems.
[18] Tamer A. Al-zohairy, Khaled S. Salem, “Model Following Control of SISO Nonlinear Systems using PID Neural Networks” in IJSA, 2016.
[19] Tarun Varshney , “NN-PID Based Control of MIMO Systems”, 2015.

Author Profile

Tamer A. Alzohairy received the B. Sc. Degree in computer science from faculty of Science, Ain Shams University, Cairo, Egypt in 1989. M.Sc degree in Computer science from Math. Dept., Faculty of science, Al-Menoufia University, Egypt, in 1997 and Ph.D in computer science from math. Dept. Faculty of science, Suez Canal University, Egypt in 2003. He is working currently in Computer science Dept., Science college, Al Azhar University, Cairo, Egypt..

Khaled S. Raslan received the B. Sc. Degree in computer science from faculty of Science, AL Azhar University, Cairo, Egypt in 2007. M.Sc student in Computer science from Math. Dept., Faculty of science, Al Azhar university, Egypt, from 2013 till now, He is working currently in Computer science Dept., Science college, Al Azhar University, Cairo, Egypt.