Journal of Theoretical and Applied Information Technology
31st January 2019. Vol.97. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ADAPTIVE PID CONTROLLER BASED ON NEURAL NETWORKS FOR MIMO NONLINEAR SYSTEMS

[1]SABRINE SLAMA, [1]AYACHI ERRACHDI, [1]MOHAMED BENREJEB

[1]Tunis El Manar University, National Engineering School of Tunis, Department of Electrical Engineering,

Le Belvédère B.P. 37, Tunis, 1002, Tunisia

E-mail: slema.sabrine@gmail.com

ABSTRACT

This paper proposes an adaptive tuning of Proportional-Integrate-Derivative (PID) controller. This approach is developed to address a class of Multi-Input Multi-Output (MIMO) nonlinear systems. The adaptive PID controller is built based on neural networks combining the PID control and explicit neural structure. The strategy of training consists of on-line tuning of the neural controller weights using the back-propagation (BP) algorithm to select the suitable combination of PID gains such that the error between the reference signal and the actual system output converges to zero. The control scheme is based on a neural network model, using a variable learning rate, of the system that is adapted by gradient descent (GD) method to learn system dynamic. The results of simulation show that improved and stable tracking is achieved with the proposed adaptive PID controller.

Keywords: *Adaptive PID Controller; Neural Network; Multivariable Nonlinear Systems.*

1. INTRODUCTION

In recent years the interest of researchers in the field of system control has seen significant growth. Among the control methods, the model reference adaptive control [1, 2, 3, and 4], the indirect control [5], the proportional-integrate-derivative (PID) controller [1, 2, 6-19] and other.

The PID controller is one of the popular methods used for systems control for its simplicity. Although this advantage, it is difficult to regulate linear or nonlinear systems using the linear PID controller based on a constant gains. For nonlinear systems, PID performance diminishes under a great deal of tuning effort of constant gains to ensure local stability. To overcome this problem, lot of efforts has been dedicated to synthesize experimental and analytical techniques to tune the PID parameters.

For instance, in [1], the adaptive PID controller and the model reference adaptive control are proposed for improving the response time and tracking performance of the hydraulic actuator control system. In [6], an adaptive PID controller is proposed using the recursive least square (RLS) algorithm and is applied on SISO stable and unstable systems considering the presence of changes in the systems parameters.

In [7], based on adaptive wavelet neural networks, a PID discrete control scheme for induction motor drives is presented. An adaptive PID control based on radial basis function neural network for quadrotor is presented in [8]. The author, in [2], proposes a model reference adaptive control and PID control to suppress the effect of nonlinear and time-varying mass unbalance torque disturbance on the dynamic performances of an aerial inertially stabilized platform. In [10], to solve the problems of low loading precision, slow response speed, and poor adaptive ability of a mobile dynamometer in a tractor traction test, a PID control strategy based on a radial basis function neural network with self-learning and adaptive ability is proposed. An on-line tuning of a neural PID controller based on plant hybrid modeling is proposed in [11] and is applied for a nonlinear model describing the response of Saccharomyces cerevisiae. In [12], a PID-type fuzzy logic controller tuning strategy is proposed using a particle swarm optimization approach and is applied to an electrical DC drive benchmark.

Therefore, there is continuous interest for researchers to develop the control methods with higher accuracy and stability by various disturbances rejection. Moreover, there is a great demand to study the general multi-input multi-output (MIMO) systems as the design of multivariable control system is highly applicable in industry since it can handle more real processes. Motivated by the above discussion, an on-line adaptive PID controller is proposed in this paper to control a MIMO nonlinear system based on on-line identification algorithm via neural networks using a variable learning rate to get fast response time and good tracking performance. Results show better performance those with better identification properties, because using tuning depends on the quality of the identification system.

This paper is organized as follows. Section 2 briefly introduces the problem under consideration. In section 3, the adaptive auto-tuning of the PID is detailed. Indeed, in this section the neural network model is demonstrated and a multivariable adaptive PID controller and its adaptation mechanism using GD method are introduced. In section 4, the proposed algorithm is proposed. In section 5, an example of a nonlinear system is presented to illustrate the proposed efficiency of the methods. At last, concluding remarks are in Section 6.

2. PROBLEM DESCRIPTION

Consider the above discrete nonlinear multivariable system with n inputs and n outputs expressed in terms of its difference equation in the following form [3]

$$Y(k+1)=f\left[Y(k),\ldots,Y(k-n_Y),U(k),\ldots,U(k-n_U)\right] \quad (1)$$

where

$Y(k)=\left[y_i(k)\right]^T_{i=1,\ldots,n}$ and $U(k)=\left[u_i(k)\right]^T_{i=1,\ldots,n}$

($Y(k)\in\mathbf{R}^n$, $U(k)\in\mathbf{R}^n$), are respectively the input and output vectors, n_Y and n_U are the number of past system input and output respectively, $f=\left[f_i\right]_{i=1,\ldots,n}$ is the nonlinear function mapping specified by the model and k is the discrete time index.

The problem is to find a controller $U(k)$ to ensure that the system output $Y(k)$ tracks as possible the desired reference $R(k)$ as $R(k)=\left[r_i(k)\right]^T_{i=1,\ldots,n}$.

To overcome this control problem, we propose an adaptive PID controller output which can be expressed as follows:

$$\begin{aligned}U(k)=&U(k-1)+K_P\left[E_c(k)-E_c(k-1)\right]+K_IE_c(k)\\&+K_D\left[E_c(k)-2E_c(k-1)+E_c(k-2)\right]\end{aligned} \quad (2)$$

where $K_P\in\mathbf{R}^n, K_I\in\mathbf{R}^n$ and $K_D\in\mathbf{R}^n$ are the proportional, integral and derivative gains vectors of the PID controllers and $E_c(k)=\left[e_{c_i}(k)\right]^T_{i=1,\ldots,n}=\left[r_i(k)-y_i(k)\right]^T_{i=1,\ldots,n}$ is the tracking error vector. Therefore, the adaptive mechanism is used for self-adjustment of the PID gains to achieve the best tracking performance.

3. ADAPTIVE AUTO-TUNING OF PID CONTROLLER

The control structure of the PID auto-tuner embedded into the control loop will be presented in this section. This structure is applied to general MIMO control system defined in equation (1). This structure, shown in Figure 1, includes the controlled system, the PID controller and the neural network tuner that is used to adjust adaptively the controller parameters where the bold arrows indicate vector-valued.

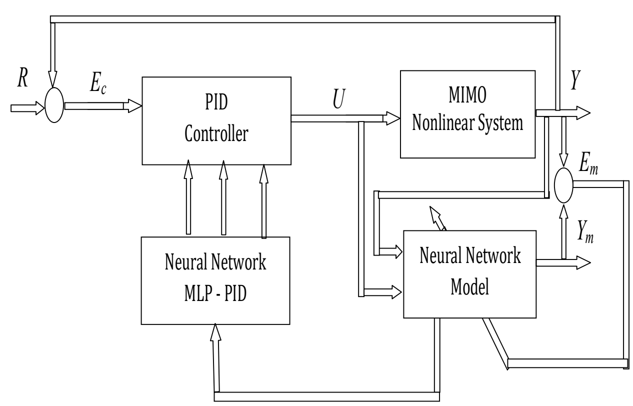

Figure 1. The PID Auto-tuning Structure for MIMO Nonlinear System

The update of the PID controller parameters are based on the efficiency of the neural network model and the neural network tuner.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.1 The Neural Network Model

The multi-layer perceptron (MLP) is used in the model network and in the NN tuner. Each block consists of three layers with the sigmoid activation function for all neurons. The identification model consists of n network outputs given by

$$y_{m_j}(k+1) = \lambda_j f\left(\sum_{k=1}^{n_2} w_{kj}^1 f\left(\sum_{i=1}^{n_1} w_{ik}^0 x_i(k)\right)\right) = \lambda_j f\left(\sum_{k=1}^{n_2} w_{kj}^1 f(h_k^0)\right) \quad ; j = 1,..,n \tag{3}$$

and can be rewritten in the following compact vector-valued form

$$Y_m(k+1) = \lambda f\left(w^{1T} F(Wx)\right) = \lambda f(net) \tag{4}$$

with $h_k^0 = f\left(\sum_{i=1}^{n_1} w_{ik}^0 x_i(k)\right)$, $w^0 = \left[w_{ik}^0\right]_{\substack{i=1,\dots,n_1\\k=1,\dots,n_2}}$, $w^1 = \left[w_{kj}^1\right]^T_{\substack{k=1,\dots,n_2\\j=1,\dots,n}}$, $x = \left[x_i\right]^T_{i=1,\dots,n_1}$, $F(Wx) = \left[f(h_k^0)\right]^T_{k=1,\dots,n_2}$, $\lambda = \left[\lambda_j\right]_{j=1,\dots,n}$ is the scaling coefficients vector, n_1 is the neuron number of input layer, n_2 is the neuron number of hidden layer and n is the neuron number of output layer and $F'(Wx) = diag\left[f'(h_1^0),\dots,f'(h_{n_2}^0)\right]$ is the Jacobian matrix of $F(Wx)$.

In the neural network structure, weights are tuned to minimize the identification error vector $E_m(k) = \left[e_{m_j}(k)\right]_{j=1,..,n}$ via the GD method.

To apply the GD method, the squared error function is defined as follows

$$J(k) = \tfrac{1}{2}\left(e_{m_1}^2(k) + \dots + e_{m_j}^2(k) + \dots + e_{m_{n_3}}^2(k)\right) \tag{5}$$

The update formula of the output weights and the hidden weights of the network are given as

$$w^1(k+1) = w^1(k) + \eta\lambda E_m(k) f'(net) F(Wx) \tag{6}$$

$$w^0(k+1) = w^0(k) + \eta\lambda E_m(k) f'(net) F'(Wx) w^1 x^T \tag{7}$$

where the derivative of the sigmoid function is [5]

$$f'(net) \approx \frac{1}{4} \tag{8}$$

and the variable learning rate is given as follows

$$\eta = 1/\left(\lambda^2 f'^2(net)\left[F^T(Wx)F(Wx) + w^{1T}F'(Wx)F'(Wx)w^1 x^T x\right]\right) \tag{9}$$

The neural network MLP model will be used to adjust the parameters of the PID controller.

3.2 The PID-MLP neural network controller

To find the PID controller parameters k_P, k_I and k_D; the used control error $e_{c_i}(k)$ can be written as:

$$e_{c_i}(k) = r_i(k) - y_i(k) \tag{10}$$

where $r_i(k)$ is the reference command.

The efficiency of this adaptive controller is estimated by a performance function defined as the squared error:

$$J_{c_i}(k) = \tfrac{1}{2}(e_{c_i}^2(k)) \tag{11}$$

The adaptive controller output can be represented in the updating algorithm as:

$$u_i(k) = u_i(k-1) + k_{P_i}(k)\left[e_{c_i}(k) - e_{c_i}(k-1)\right] + k_{I_i}(k)e_{c_i}(k) + k_{D_i}(k)\left[e_{c_i}(k) - 2e_{c_i}(k-1) + e_{c_i}(k-2)\right] \tag{12}$$

where k_{P_i}, k_{I_i} and k_{D_i} are the i^{th} proportional gain, integral gain and derivative gain respectively.

Let propose the proportional error as

$$e_{P_i}(k) = e_{c_i}(k) - e_{c_i}(k-1) \tag{13}$$

the integral error as

$$e_{D_i}(k) = e_{c_i}(k) - 2e_{c_i}(k-1) + e_{c_i}(k-2) \tag{14}$$

and the derivative error as

$$e_{I_i}(k) = e_{c_i}(k) \tag{15}$$

then, the adaptive controller output $u_i(k)$ can be represented in the updating algorithm as:

$$u_i(k) = u_i(k-1) + k_{P_i}(k)e_{P_i}(k) + k_{I_i}(k)e_{I_i}(k) + k_{D_i}(k)e_{D_i}(k) \tag{16}$$

Journal of Theoretical and Applied Information Technology
31st January 2019. Vol.97. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The PID-MLP structure is presented in Figure 2, where the network parameters are given as follows :

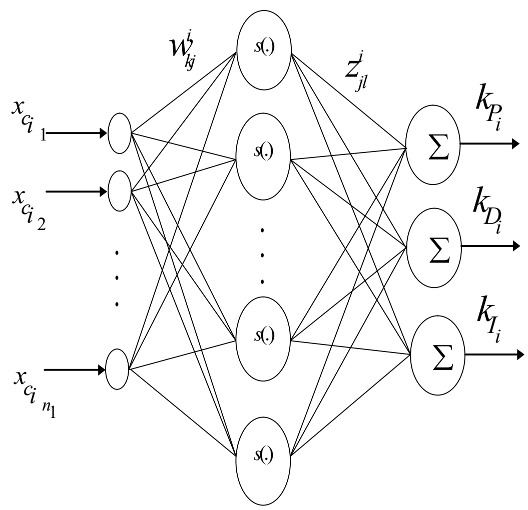

Figure 2. The PID-MLP Auto-tuning Neural Network Structure for MIMO System

$$O_i(k) = \left[O_1^i(k)\ O_2^i(k)\ O_3^i(k)\right]^T = \left[k_{P_i}(k)\ k_{I_i}(k)\ k_{D_i}(k)\right]^T \quad (17)$$

with

$$O^i(k) = \lambda_{c_i} s\left(\sum_{j=1}^{n_4} z^i_{jl} s\left(\sum_{k=1}^{n_3} w^i_{kj} x_{c_{ik}}(k)\right)\right) = \lambda_{c_i} s\left(\sum_{j=1}^{n_4} z^i_{jl} s(h_j^1)\right) \quad (18)$$

and can be rewritten in the following compact vector-valued form

$$O^i(k) = \lambda_c s\left(z^{i^T} S(Wx_{ci})\right) = \lambda_c s(net_{PID}) \quad (19)$$

with $S(Wx_c) = \left[s(h_j^1)\right]^T_{j=1,\ldots,n_4}$, $w^i = \left[w^i_{kj}\right]_{k=1,\ldots,n_3;\ j=1,\ldots,n_4}$,

$z^i = \left[z^i_{jl}\right]^T_{j=1,\ldots,n_4;\ l=1,\ldots,3}$ and λ_{c_i} is the scaling coefficients.

The weights update of the PID-MLP is obtained by minimizing the cost function as follows:

$$\frac{\partial J_{c_i}(k)}{\partial \gamma^i(k)} = -e_{c_i}(k)\frac{\partial y_i(k)}{\partial u_i(k)}\frac{\partial u_i(k)}{\partial O_i(k)}\frac{\partial O^i(k)}{\partial net_{PID_i}(k)}\frac{\partial net_{PID_i}(k)}{\partial \gamma^i(k)} \quad (20)$$

where γ^i can take w^i or z^i .

Finally, the parameters update of the PID-MLP is obtained as follows:

$$z^i(k+1) = z^i(k) + \eta_{c_i}\lambda_{c_i}\lambda e_{c_i}(k) f'(net) z^0 F'^T(Wx) w_1^0 J_{u,O} s'(net_{PID}) S(Wx_c) \quad (21)$$

$$w^i(k+1) = w^i(k) + \eta_{c_i}\lambda_{c_i}\lambda e_{c_i}(k) f'(net) z^0 F(Wx) w_1^0 J_{u,O} s'(net_{PID}) z_i S'(Wx_c) x_c \quad (22)$$

with $J_{u,O} = \dfrac{\partial u_i(k)}{\partial O^i(k)} = \left[e_{P_i}(k)\ e_{I_i}(k)\ e_{D_i}(k)\right]^T$

and $\eta_{c_i} = \left[\eta_{P_i}\ \eta_{I_i}\ \eta_{D_i}\right]^T$.

3.3 The proposed algorithm

In this section, a summary of the proposed algorithm of the adaptive PID controller for MIMO nonlinear system is presented.

Offline phase:

- Initialization of neural network parameters of the neural network model w^0 and w^1 using M observations, ($M << N$).

Online phase:

At time instant $(k+1)$, we have a new data $r_i(k+1)$, using the obtained input vector x_{c_i} , if the condition $e_{m_i}(k+1) \le \varepsilon_1$, where $\varepsilon_1 > 0$ is a given small constant, is satisfied then the neural network model, given by the equation (5), approaches sufficiently the behavior of the system,

- If the condition $e_{c_i}(k+1) \le \varepsilon_2$, where $\varepsilon_2 > 0$ is a given small constant, is satisfied then the neural network PID controller provides sufficiently the control law $u(k+1)$,
- If $e_{m_i}(k+1) \le \varepsilon_1$, is not satisfied the update of the synaptic weights of the neural network model is necessary, using the equation (8) and (9),

- If $e_{c_i}(k+1) \le \varepsilon_2$, is not satisfied the update of the synaptic weights of the neural network PID controller is necessary, using the equation (21) and (22),
- End.

4. SIMULATION RESULTS

In order to evaluate the performance of the proposed method, simulations of a nonlinear system [3] are carried out.

$$y_1(k+1) = \frac{y_1(k)}{1+y_2^2(k)} + u_1(k) \tag{23}$$

$$y_2(k+1) = \frac{y_1(k)y_2(k)}{1+y_2^2(k)} + u_2(k) \tag{24}$$

where $u_i(k)$ and $y_i(k)$, $i = 1,2$, are respectively the inputs and the outputs of the system.

Both neural network model and neural network tuner consist of an input layer, a single hidden layer with 20 nodes, and an output layer, identically. The used scaling coefficients are $\lambda_{ci} = \lambda_i = 1$.

4.1 System without disturbances

In this section, we examine the effectiveness of the proposed control system for the nonlinear multi-input multi-output system without disturbances. Indeed, the parameters of the adaptive PID controller need an online identification model for controlling the system. Once the neural network model is identified, the obtained model is used to produce as the neural network model of the control architecture, as shown in Figure 1. The MLP neural network model is based on a variable learning rate, given by the expression (9), and a derivative of the sigmoid function, given by the expression (8), guarantees faster convergence and better identification performance.

Indeed, in offline phase, using a reduced number of observations ($M = 4$) to find, either, the parameters initialization of the neural network model (w^0, w^1).

In online phase, at instant ($k+1$), we use the input vector of the neural network MLP-PID controller

$$x_{c_1} = \left[e_{c_1}(k) \quad r_1(k) \quad u_1(k-1) \right]^T \tag{25}$$

for the first output and

$$x_{c_2} = \left[e_{c_2}(k) \quad r_2(k) \quad u_2(k-1) \right]^T \tag{26}$$

for the second output.

The tracking control objective for this system is to follow as possible the reference signal. In this simulation, the reference signal vector, $R(k)$, is the desired value which is defined as

$$\begin{cases} r_1(k) = \sin(\frac{2k\pi}{100}) \\ r_2(k) = \sin(\frac{2k\pi}{250}) \end{cases} \tag{27}$$

The mean squared tracking error (MSE), given by the expression (28), as a way of evaluating the tracking error between the system output and the desired value, is used to perform a PID controller.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(r_i(k) - y_i(k))^2 \tag{28}$$

In neural network MLP-PID, initial weight value was set to a random value ranging from -0.5 to 0.5, the learning rates η_1 and η_2 were set to 0.46 and 0.35 respectively.
Figure 3 shows the response of the system $y_1(k)$ and $y_2(k)$ toward the reference inputs $r_1(k)$ and $r_2(k)$ with the tracking errors $e_{c_1}(k)$ and $e_{c_2}(k)$. The evolution of the control laws are presented in Figure 4. The results show that the tracking of adaptive PID controller has some oscillations (overshot) at the beginning that might be caused by the randomly set neural network weights. As time goes on, the system outputs track the reference inputs satisfactorily.
A concordance between both the system outputs and the desired values with MSE (e_{c1}) equal to 0.0029 and MSE (e_{c2}) equal to 0.0027 are noticed.
The PID parameters $k_{P_1}, k_{P_2}, k_{D_1}, k_{D_2}, k_{I_1}$ and k_{I_2} are adjusted by self-learning NN until the tracking errors approach zero asymptotically.

Figure 5 corresponds to the behavior presented by the PID controller gains during 500 iterations.

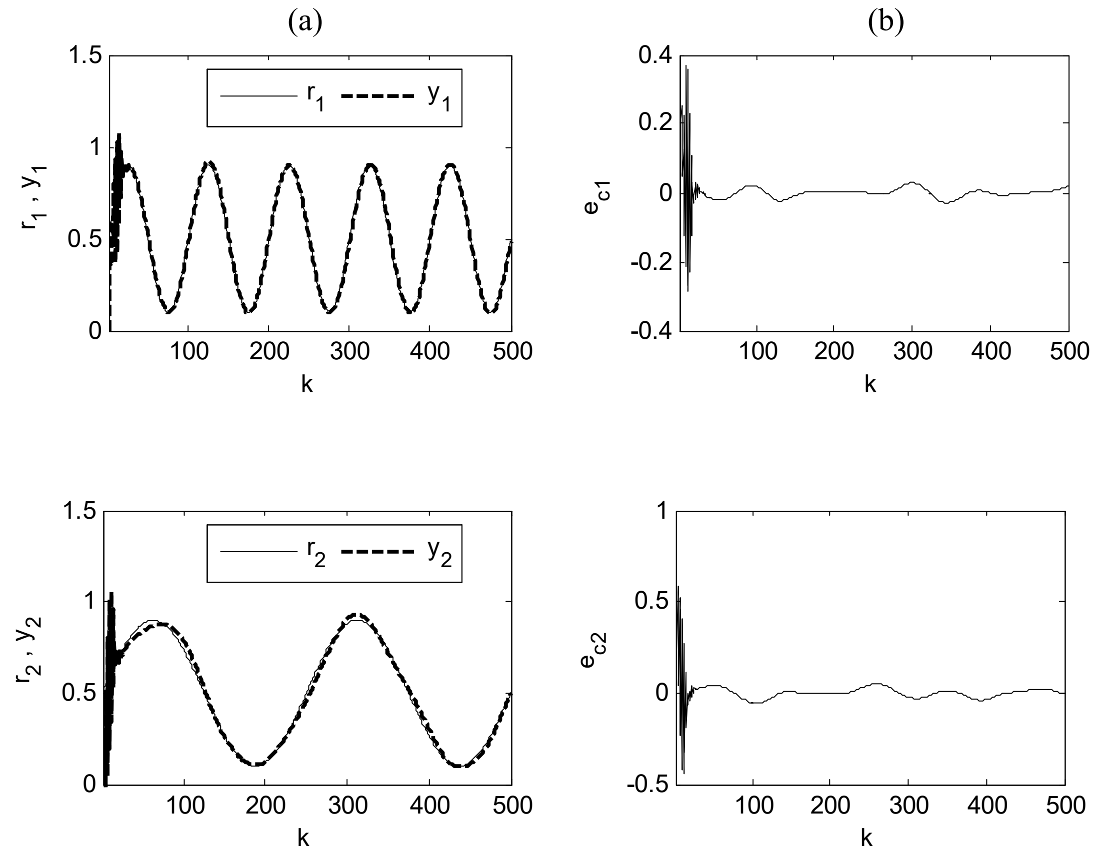

Figure 3. Response of the NN adaptive PID scheme :(a) Reference signals and Plant responses; (b) Tracking errors.

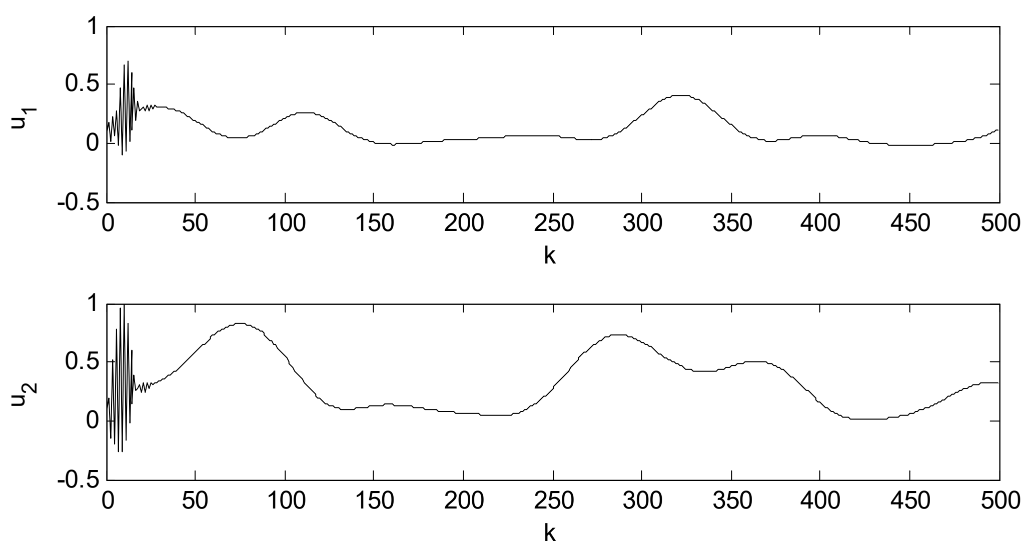

Figure 4. Evolution of Control Signals

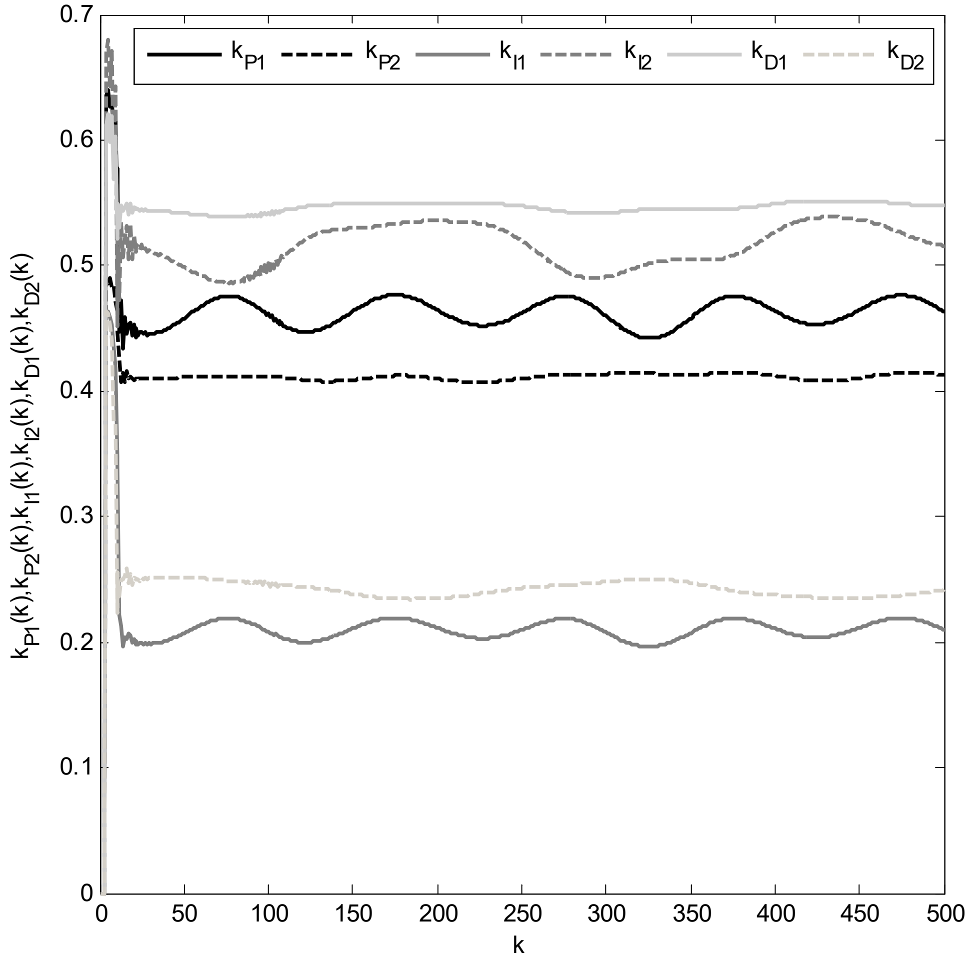

Figure 5. Evolution of PID Gains

As it has been presented, the BP neural network technique is capable of providing suitable parameters for MLP-PID controller and the targeted system output can be achieved.

4.2 System with disturbances

To examine the ability of the proposed PID controller system from disturbances, a random signal is added to the outputs of the nonlinear system.

The performance of the on-line adaptive tuning based on MLP-PID algorithm in terms of MSE is proved. Analysis shows that the response speed, stability, small system error and adaptability of the BPNN based PID control system have been guaranteed (can quickly response to such situation and calculate the suitable $k_{P_1}, k_{P_2}, k_{D_1}, k_{D_2}, k_{I_1}$ and k_{I_2} for the PID controller so that the system can be kept stable).

Figure 6 shows the response of the system $y_1(k)$ and $y_2(k)$ toward the reference inputs $r_1(k)$ and $r_2(k)$ with the tracking errors $e_{c_1}(k)$ and $e_{c_1}(k)$. The evolution of the control laws are presented in Figure. 7. The results show that the tracking of adaptive PID controller has some oscillations (overshot) at the beginning that might be caused by the randomly set neural network weights. As time goes on, the system outputs track the reference inputs satisfactorily.

A concordance between both the system outputs and the desired values with MSE (e_{c1}) equal to 0.0036 and MSE (e_{c2}) equal to 0.0047 are noticed. The PID parameters $k_{P_1}, k_{P_2}, k_{D_1}, k_{D_2}, k_{I_1}$ and k_{I_2} are adjusted by self-learning NN until the tracking errors approach zero asymptotically.

Figure 8 corresponds to the behavior presented by the PID controller gains during 500 iterations.

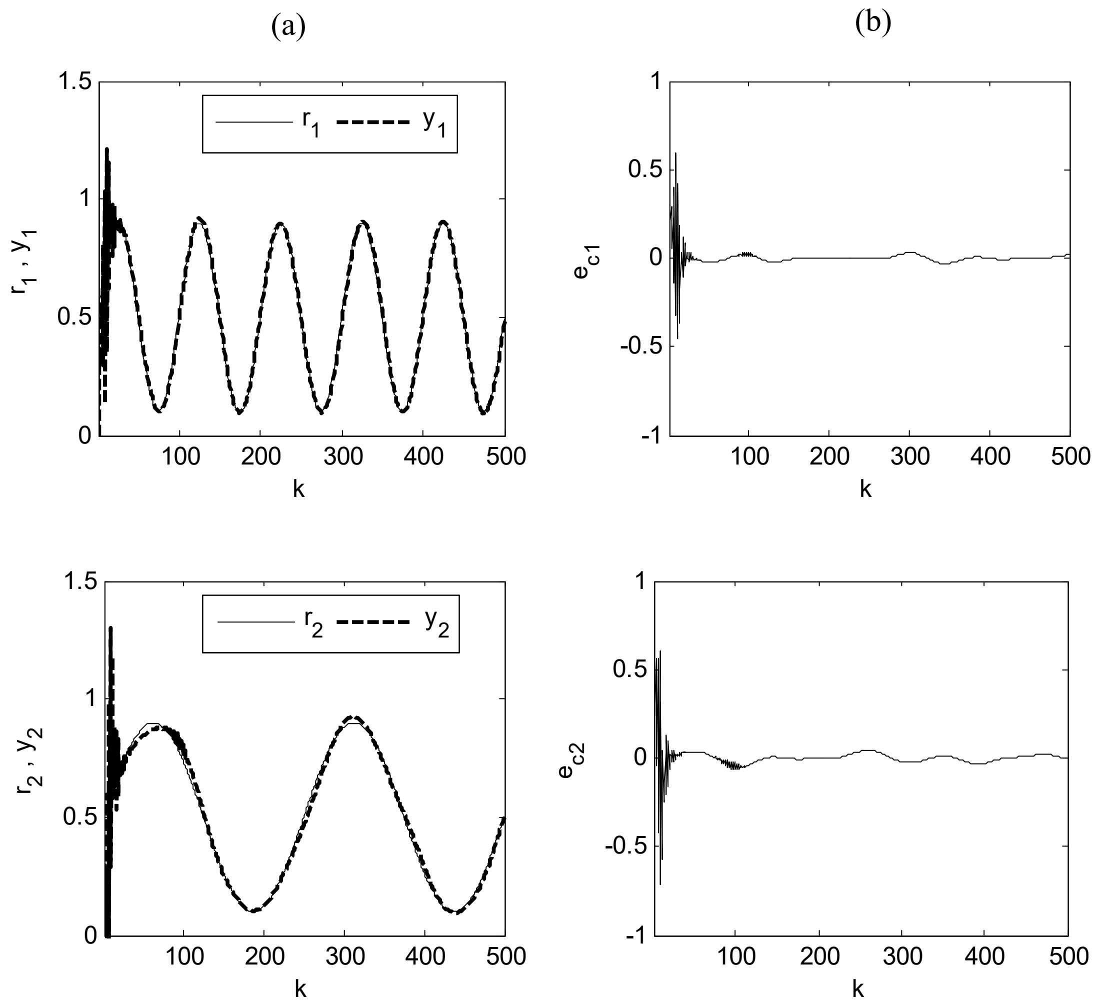

Figure 6. Response of the NN adaptive PID scheme in presence of noise :(a) Reference signals and Plant responses; (b) Tracking errors

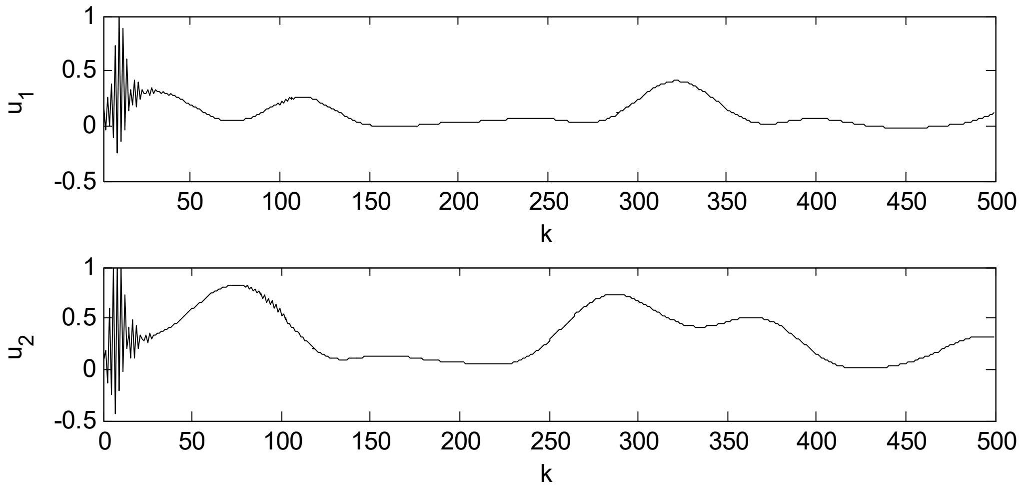

Figure 7. Evolution of Control Signals in presence of noise

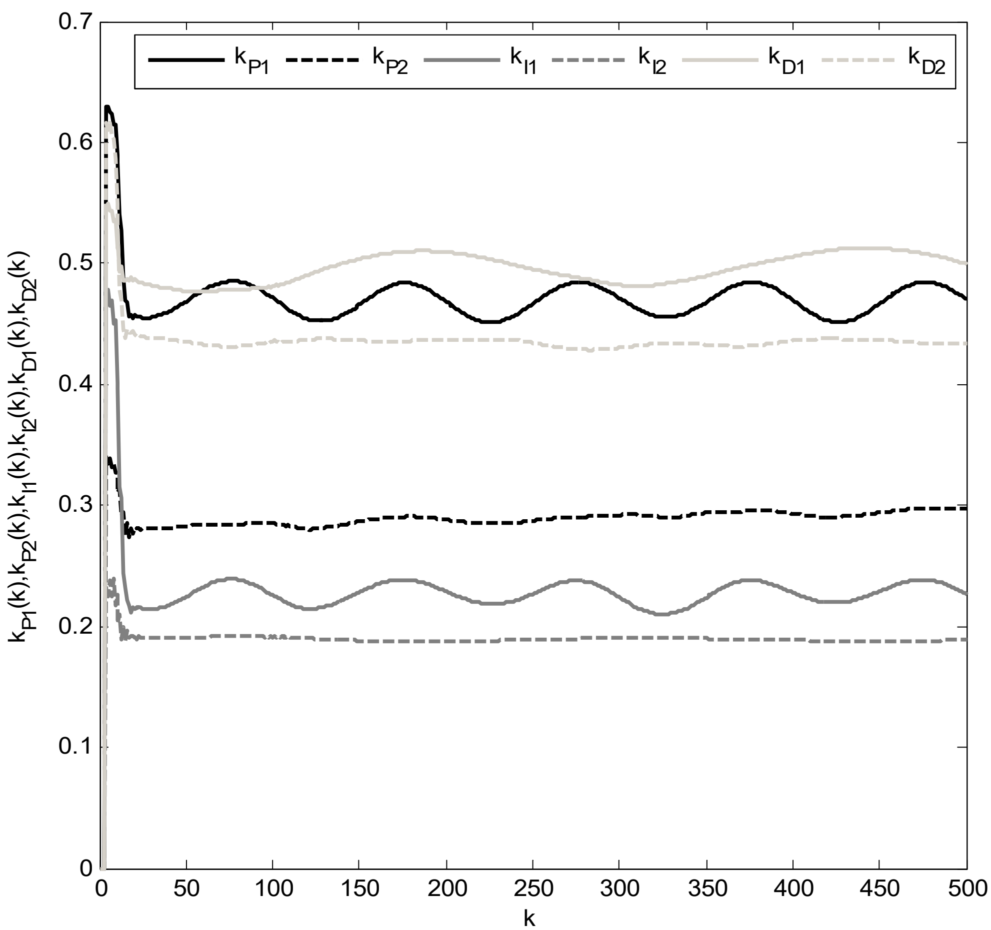

Figure 8. Evolution of PID Gains in presence of noise

5. CONCLUSION

In this paper, an adaptive tuning method for proportional-integrate-derivative controller is proposed to control a MIMO nonlinear system. The neural network tuner is trained online in order to make the real system close as possible to the reference signal. In each discrete time step, back propagation algorithm with the gradient descent method is used to update sub-networks weights. This requires the use of a model network, which is trained online, to obtain an approximation to the plant Jacobian. The control strategy used to define the adaptation law is based on the tracking error between the plant output and the reference. The use of neural networks allows nonlinearities in the controlled system to be considered for tuning purposes.

Experimental results done on multi-input multi output nonlinear system have shown that using the proposed controller, the system outputs can track asymptotically the desired references in less time and more efficiency. Also, the results show that the adaptive PID scheme is good at the disturbance rejection.

Comparison study with other works has shown that the proposed method gives faster and accurate results due to its simplicity, fast adaptation mechanism for the discrete nonlinear MIMO systems.

REFERENCES

[1] Xin Zuo, Jian-wei Liu, Xin Wang, and Hua-qing Liang, "Adaptive PID and model reference adaptive control switch controller for nonlinear Hydraulic Actuator", *Mathematical Problems in Engineering*, 2017, pp. 1-15.

[2] Xiangyang Zhou, Chao Yang, and Tongtong Cai, "A model reference adaptive control PID compound scheme on disturbance rejection for an aerial inertially stabilized platform", *Journal of Sensors*, 2016, pp. 1-11.

[3] Sabrine Slama, Ayachi Errachdi, Mohamed Benrejeb , "Model reference adaptive control for MIMO nonlinear systems using RBF neural networks", *2nd International Conference on Advanced Systems and Electrical Technologies (IC_ASET),* Hammamet, March 22-25,2018.

[4] R. Prakash, R. Anita, Neuro- PI controller based model reference adaptive control for nonlinear systems, International Journal of Engineering, Science and Technology Vol. 3, No. 6, 2011, pp. 44-60.

[5] Ayachi Errachdi and Mohamed Benrejeb, "Performance comparison of neural network training approaches in indirect adaptive control", *International Journal of Control, Automation and Systems*, Vol. 16, No. 3, 2018, pp. 1448-1458.

[6] R.A. Fahmy, R.I. Badr, and F. A. Rahman, "Adaptive PID controller using RLS for SISO stable and unstable systems", *Advances in Power Electronics*, Vol. 2014, 2014, pp. 1-5.

[7] O. A. Dominguez-Ramirez, L. E. Ramos-Velasco, and V. Parra-Vega, "Online identification for Auto-tuning PID based on wavelet neural networks: an experimental validation on an AC motor", *European Control Conference (ECC)*, Zürich, Switzerland, July 17-19, 2013.

[8] H. Hasanpour, M. Heydari Beni, M. Askari, "Adaptive PID control based on RBF NN for quadrotor", *International Research Journal of Applied and Basic Sciences*, Vol. 11 (2), 2017, 177-186.

[9] R.A. Fahmy, R.I. Badr, and F.A. Rahman, "Alternative approach to use RLS algorithm in multivariable online adaptive PID controllers for MIMO systems", *IETE Journal of Research*, Vol. 64, No. 1, 2018, pp. 27–35.

[10] Xianghai Yan, Liyou Xu, and Yuan Wang, "The loading control strategy of the mobile dynamometer vehicle based on neural network PID", *Mathematical Problems in Engineering*, 2017, pp. 1-7.

[11] A. Andrášik, A. Mészáros, S.F. de Azevedo, "On-line tuning of a neural PID controller based on plant hybrid modeling", *Computers and Chemical Engineering*,Vol. 28, 2004, pp. 1499–1509.

[12] S. Bouallègue, J. Haggège, M. Ayadi, M. Benrejeb, "PID-type fuzzy logic controller tuning based on particle swarm optimization", *Engineering Applications of Artificial Intelligence*, Vol. 25, 2012, pp. 484–49.3

[13] Meric Cetin, SerdarIplikci, "A novel auto-tuning PID control mechanism for nonlinear

systems", *ISA Transactions*, Vol. 58, 2015, pp. 292–308.

[14] Wei Lu, Jianhua Yang and Xiaodong Liu, "The PID controller based on the artificial neural network and the differential evolution algorithm", *Journal of Computers*, Vol. 7, No. 10, OCTOBER 2012, pp. 2368--2375.

[15] Rajesh Kumar, Smriti Srivastava and J.R.P. Gupta, "Artificial neural network based PID controller for online control of dynamical systems", *1st IEEE International Conference on Power Electronics. Intelligent Control and Energy Systems (ICPEICES),*2016.

[16] M. K. O. Ayomoh, M. T. Ajala, "Neural network modeling of a tuned PID controller", *European Journal of Scientific Research*, Vol. 71, No. 2, 2012, pp. 283-297.

[17] Junghui Chen, Tien-Chih Huang, "Applying neural networks to on-line updated PID controllers for nonlinear process control", *Journal of Process Control* , Vol. 14, 2004, pp. 211–230.

[18] Jianhua Zhang and Junghui Chen, "Neural PID control strategy for networked process control", *Mathematical Problems in Engineering*, 2013, pp. 1-11.

[19] Cheng Cheng, Min-Sen Chiu, "Robust PID controller design for nonlinear processes using JITL technique", *Chemical Engineering Science*, Vol. 63, 2008, pp. 5141 -- 5148.