Neural Network Controller Based on PID Controller for Two links- Robotic Manipulator Control

Leila Fallah Araghi, M. Habibnejad Korayem, Amin Nikoobin, Farbod Setoudeh

Abstract— **This paper a new method for the control of Two link-robotic manipulator systems using Neural Network has been presented, The first method is based on Proportional-Derivative controller, The second method is based on Proportional-Integral-Derivative controller, the third method is based on artificial Neural Network by PD controller, and the forth method is based on artificial Neural Network by PID controller for control of Two link- robot.**

***Index Terms* — Two link- robotic manipulator systems, Neural Network, PD controller, PID controller**.

I. INTRODUCTION

In the recent years using intelligece control such as fuzzy control, Neural Network, Neuro Fuzzy and because that they can control nonlinear systems that would be difficult or impossible to model mathematically. In the recently years In Dynamic control of robot have been utilized in many researchers work in this area. Such as Lianfang Tian et al use a neural network approach for the motion control of constrained flexible manipulators robots [1].

Yi , et al have investigated the robustness and stability of a fuzzy logic controller applied to a robotic manipulator with uncertainties such as friction, unmodeled dynamics, and external disturbance etc [2], Kumbla et al have implemented hierarchical control on robotic manipulator using fuzzy logic [3].

Bannerjee, et al have used a Fuzzy Logic Controller to achieve position control of a two-link manipulator [4]. Adams, et al [5] has used GA to optimize the membership functions and rule bases of a multi-stage fuzzy PID controller with a fuzzy switch for robot control. Brudka, et al presented an intelligent robot control system which employs ultrasonic distance measurements, and for

Consecutive stages of data processing they used to neural networks applications [6], Adamiv, et al

used neural networks application for mobile robot control on predetermined trajectory of the road [7], Ya-Chen , et al used an Fuzzy neural adaptive controller to multiple-link robot control [8], Devendra P, et al used the proportional plus derivative (PD) control with the PD controller gain parameters optimized via Genetic Algorithm (GA) And Fuzzy Logic for control of Two link- robot [9], Z.G. Zhang, et al report on the design and stability analysis of a simple quadruped running controller that can autonomously generate steady running with good energy efficiency and suppress.

Dongbing Gu, et al presented a new path-tracking scheme for a car-like mobile robot based on neural

predictive control, they employed A multi-layer back-propagation neural network to model non-linear kinematics of the robot instead of a linear regression estimator in order to adapt the robot to a large operating range [10], Mathew L, et al studied on the implementation of several Intelligent control techniques as applied to the balancing of the inverted wedge problem. These included a basic four-input direct fuzzy controller (including the use of the nonlinear input term) and an adaptive fuzzy control technique known as the FMRLC [11].

II. DYNAMIC EQUATION OF TWO LINK- ROBOTIC MANIPULATOR SYSTEMS [12]

In this section we derive the equations of motion for an individual link based on the direct method has been derived, i.e. Newton-Euler Formulation. The motion of a rigid body can be decomposed into the translational motion with respect to an arbitrary point fixed to the rigid body, and the rotational motion of the rigid body about that point. The dynamic equations of a rigid body can also be represented by two equations: one describes the translational motion of the centroid (or center of mass), while the other describes the rotational motion about the centroid. The former is Newton's equation of motion for a mass particle, and the latter is called Euler's equation of motion.

Figure 1 shows all the forces and moments acting on link *i*.

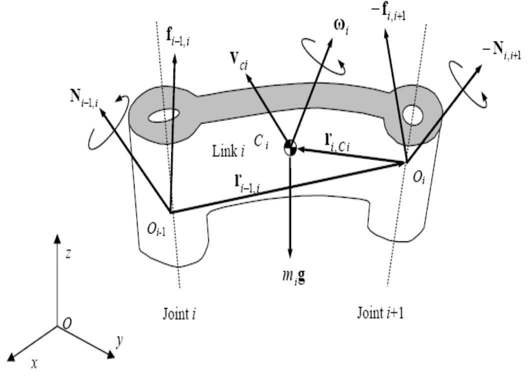

Fig. 1: Free body diagram of link *i* in motion

Let be V_{ci} the linear velocity of the centroid of link *i* with reference to the base coordinate frame *O-xyz,* which is an inertial reference frame. The inertial force is then given by $-m_i \dot{V}_{ci}$, where m_i is the mass of the link and $\dot{V}_{ci}$ is the time derivative of V_{ci}.

Based on D'Alembert's principle, the equation of motion is then obtained by adding the inertial force to the static balance of forces in eq.(1) so that

$$f_{i-1,i} - f_{i,i+1} + m_i g - m_i \dot{V}_{ci} = 0 \ , i = 0,1,...,n \tag{1}$$

$f_{i-1,i}$ and $-f_{i,i+1}$ are the coupling forces applied to link *i* by links *i*-1 and *i*+1, respectively, and **g** is the acceleration of gravity.

Adding these terms to the original balance of moments we have:

$$N_{i-1,i} - N_{i,i+1} - (r_{i-1,i} - r_{i,ci}) \times f_{i-1,i} + (-r_{i,ci}) \times f_{i,i+1} - I_i \dot{\omega}_i - \omega_i \times (I_i \omega_i) = 0 \tag{2}$$

$N_{i-1,i}$ and $-N_{i,i+1}$ are the moment applied to link *i* by links *i*-1 and *i*+1, respectively.

If we consider two individual links robot, Let us obtain the Newton-Euler equations of motion for the two individual links.

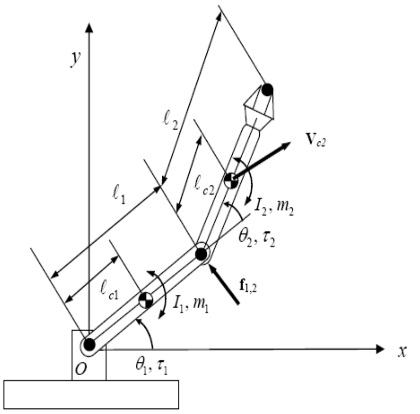

Fig.2: Mass properties of two planar robots

From eq. (1) and (2), the Newton-Euler equations for link 1 are given by:

$$f_{0,1} - f_{1,2} + m_1 g - m_1 \dot{V}_{c1} = 0 \tag{3}$$

$$N_{0,1} - N_{1,2} - (r_{0,1}) \times f_{0,1} + (r_{1,c1}) \times f_{1,2} - I_1 \dot{\omega}_1 = 0 \tag{4}$$

Note that all vectors are 2 x 1, so that moment N *i-1,i* and the other vector products are scalar quantities. Similarly, for link 2,

$$f_{1,2} + m_2 g - m_2 \dot{V}_{c2} = 0 \tag{5}$$

$$N_{1,2} - (r_{1,c2}) \times f_{1,2} - I_2 \dot{\omega}_2 = 0 \tag{6}$$

To obtain closed-form dynamic equations, we first eliminate the constraint forces and separate them from the joint torques, so as to explicitly involve the joint torques in the dynamic equations. For the planar manipulator, the joint torques τ*1* and τ*2* are equal to the coupling moments:

$$N_{i-1,i} = \tau_i \tag{7}$$

Substituting eq.(7) into eq.(6) :

$$\tau_2 - (r_{1,c2}) \times (m_2 \dot{V}_{c2} - m_2 g) - I_2 \dot{\omega}_2 = 0 \tag{8}$$

Similarly, eliminating $f_{0,1}$ yields,

$$\tau_1 - \tau_2 - (r_{0,c1}) \times m_1 \dot{V}_{c1} + (r_{0,1}) \times m_2 \dot{V}_{c2} + (r_{0,c1}) \times m_1 g + (r_{0,1}) \times m_2 g - I_1 \dot{\omega}_1 = 0 \tag{9}$$

Next, we rewrite V_{ci}, ω_{ci} and $r_{i-1,i}$ using joint displacements θ_1, θ_2 which are independent variables. Note that ω_2 is the angular velocity relative to the base coordinate frame, while θ_2 is measured relative to link 1. Then, we have

$$\omega_1 = \dot{\theta}_1, \omega_2 = \dot{\theta}_1 + \dot{\theta}_2 \tag{10}$$

The linear velocities can be written as

$$V_{c1} = \begin{bmatrix} -l_{c1} \dot{\theta}_1 \sin(\theta_1) \\ l_{c1} \dot{\theta}_1 \cos(\theta_1) \end{bmatrix} \tag{11}$$

$$V_{c2} = \begin{bmatrix} -\{l_1 \sin(\theta_1) + l_{c2} \sin(\theta_1 + \theta_2)\} \dot{\theta}_1 - l_{c2} \sin(\theta_1 + \theta_2) \dot{\theta}_2 \\ \{l_1 \cos(\theta_1) + l_{c2} \cos(\theta_1 + \theta_2)\} \dot{\theta}_1 - l_{c2} \cos(\theta_1 + \theta_2) \dot{\theta}_2 \end{bmatrix}$$

(12)

$$\tau_1 = H_{11} \ddot{\theta}_1 + H_{12} \theta_2 - h \dot{\theta}_2 - 2h \dot{\theta}_1 \dot{\theta}_2 + G_1 \tag{13}$$

$$\tau_2 = H_{22}\,\ddot{\theta}_2 + H_{21}\,\ddot{\theta}_1 - h\,\dot{\theta}_1 + G_2$$

That:

$$H_{11} = m_1\,l_{c1}{}^2 + I_1 + m_1(l_1{}^2 + l_{c2}{}^2 + 2l_1l_{c2}\cos(\theta_2)) + I_2$$

$$H_{22} = m_1\,l_{c2}{}^2 + I_2 \qquad (14)$$

$$G_1 = m_1\,l_{c1}\;g\cos(\theta_1) + m_2 g\{l_{c2}\cos(\theta_1+\theta_2) + l_1\cos(\theta_1)\}$$

$$H_{11} = m_2(l_{c2}{}^2 + l_1l_{c2}\cos(\theta_2)) + I_2$$

$$h\; = m_2l_1l_{c2}\sin(\theta_2) \qquad (15)$$

$$G_2 = m_2 g\,l_{c2}\cos(\theta_1+\theta_2) \qquad (16)$$

III. NEURAL NETWORK CONTROLLER BASED ON PID AND PD CONTROLLER [13, 14, 15, 16]

This paper presents two strategies for achieving the control in two link- robotic manipulator systems. First one is based on Proportional plus Derivative (PD) controller with its control by use Neural Network controller based on PD controller.

The performance of a PD controller depends upon the proportional gain (Kp) and derivative gain (Kd). Dependent upon the values of Kp and Kd are the stability, settling time, maximum overshoot and many other system performance indicators. The transform function of PD controller that used in this paper is [16]:

$$G(S) = K_P + K_D S \qquad (17)$$

In this paper a PD controller is used in control of Two link- robotic manipulator systems, The block diagram of a PD controllers is shown in Figure 3.

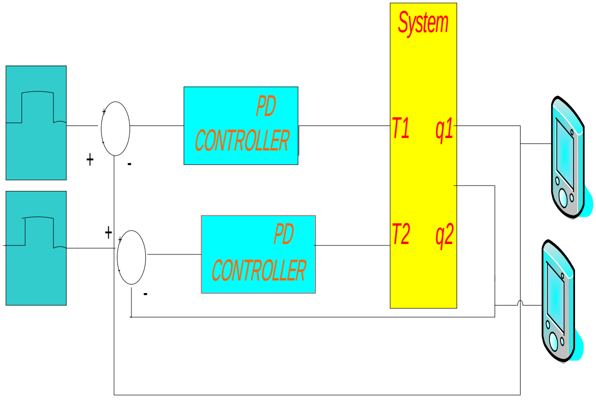

Fig. 3: the schematic of PD Controller

Artificial neural network can be applied to various problems such as function approximation, pattern recognition, signal processing, classification, etc. Appendix I shows the more details about general view on artificial neural network. There are typically two steps involved when using neural networks for control:

1-System identification
2-Control design

In the system identification stage, you develop a neural network model of the plant that you want to control. In the control design stage, you use the neural network plant model to design (or train) the controller. In each of the three control architectures described in this chapter, the system identification stage is identical.

The model predictive control method is based on the receding horizon technique. The neural network model predicts the plant response over a specified time horizon [14, 16];

The Neural Network controller based on PD controller has been used for control of Two link- robotic manipulator systems, The block diagram of a Neural Network controllers based on PD controllers is shown in Figure 4.

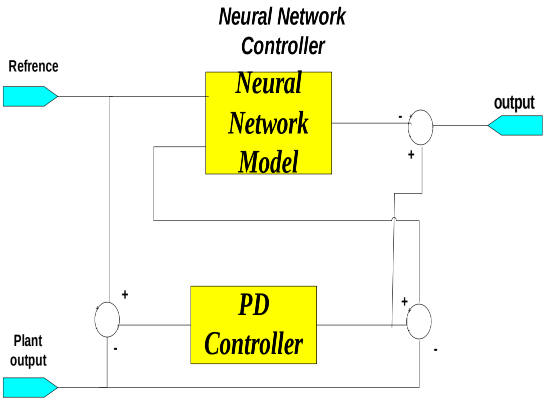

Fig. 4: the structure of the Neural Network controller

The second strategy is based on Proportional-Integral-Derivative (PID) with its control by use Neural Network controller based on PID controller both of these strategies are briefly described below:

PID stands for Proportional-Integral-Derivative. This is a type of feedback controller whose output, a control variable (CV), is generally based on the error (e) between some user-defined set point (SP) and some measured process variable (PV). Each element of the PID controller refers to a particular action taken on the error:

Proportional: error multiplied by a gain, K_p. This is an adjustable amplifier. In many systems K_p is responsible for process stability: too low and the PV can drift away; too high and the PV can oscillate. Integral: the integral of error multiplied by a gain, K_i. In many systems K_i is responsible for driving error to zero, but to set K_i too high is to invite oscillation or instability or integrator windup or actuator saturation Derivative: the rate of change of error multiplied by a gain, K_d. In many systems K_d is responsible for system response: too high and the PV will oscillate; too low and the PV will respond sluggishly. The designer should also note that derivative action amplifies any noise in the error signal.

The transform function of PID controller that used in this paper is:

$$G(S) = K_P + K_I S + \frac{K_D S}{NS + 1} \tag{18}$$

In this strategy PID controller is used in control of Two link-robotic manipulator systems, The block diagram of a PID controllers is shown in Figure 5.

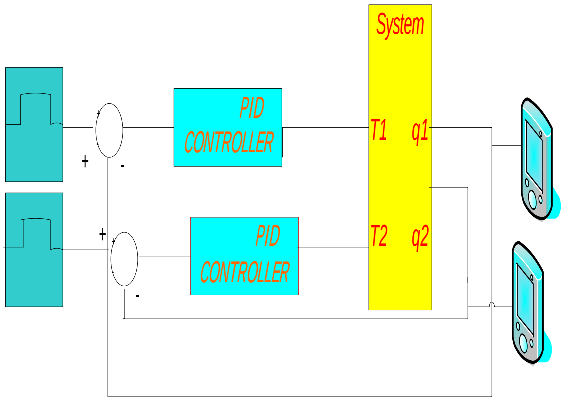

Fig. 5: the schematic of PID Controller

The Neural Network controller based on PID controller has been used for control of Two link- robotic manipulator *systems*, The block diagram of a Neural Network controllers based on PID controllers is shown in Figure 6.

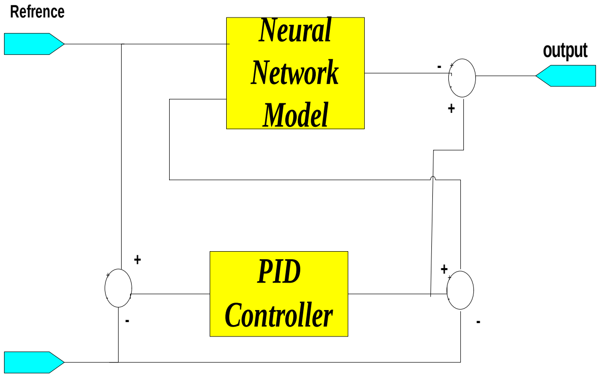

Fig. 6: the structure of the Neural Network controller based on PID controllers

The structure of Neural Network controller based on PID controller and PD controller that used in control of Two link-robotic manipulator systems, is shown in figure 7.

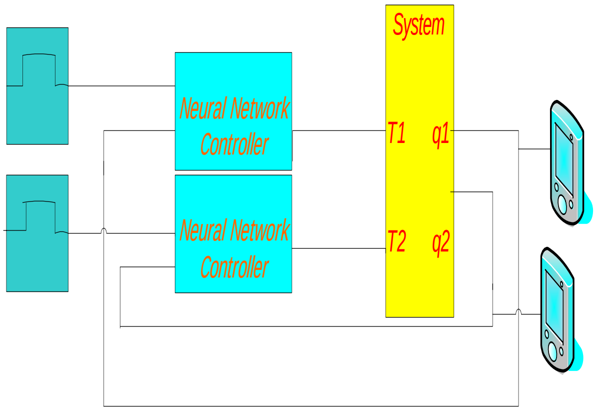

Fig. 7: the schematic of Neural Network controller

In Neural Network controller as learning rules Modified Levenberg-Marquardt has been used [13,15].

IV. SIMULATION RESULT

In the first the dynamic model of two link- robotic manipulator systems has been simulated using Matlab software 2007 as you see the system is unstable. Figure 8 shows the schematic of Two link- robotic manipulator systems and figure 9 shows the simulation result for step input:

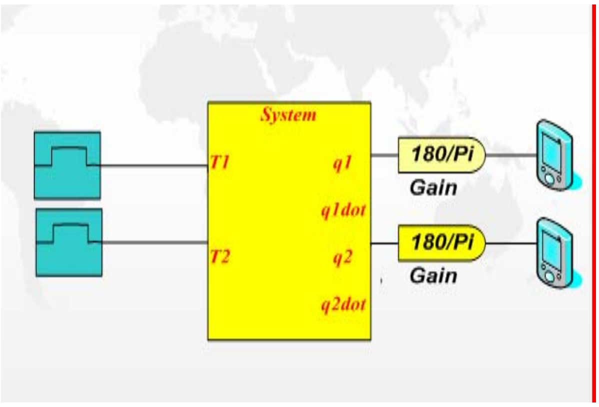

Fig. 8: the schematic of the Two link- robotic manipulator systems

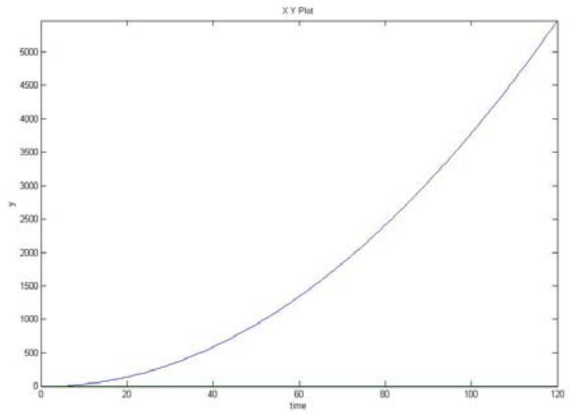

Fig. 9: the output of two link- robotic manipulator systems

As the first method for closed loop control of two link robot the PD controller has been used, figure 10 shows the simulation result

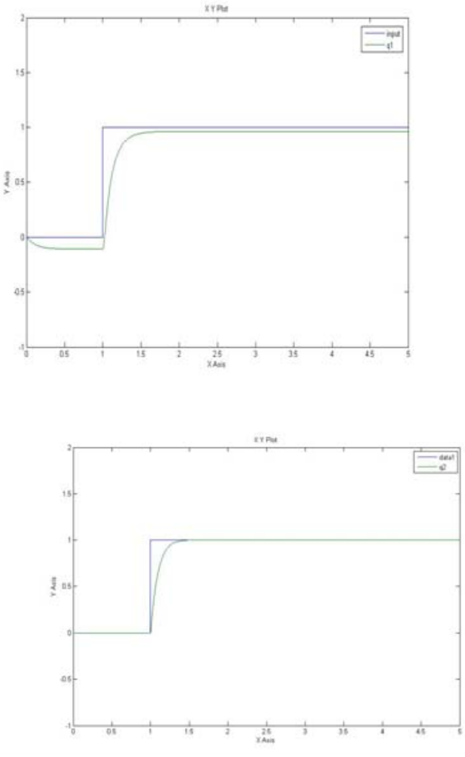

Fig. 10: the output of two link- robotic manipulator systems using PD controller

As the second method closed loop control of robotic manipulator systems the Neural Network controller based on PD controller has been used, figure 11 shows the simulation result:

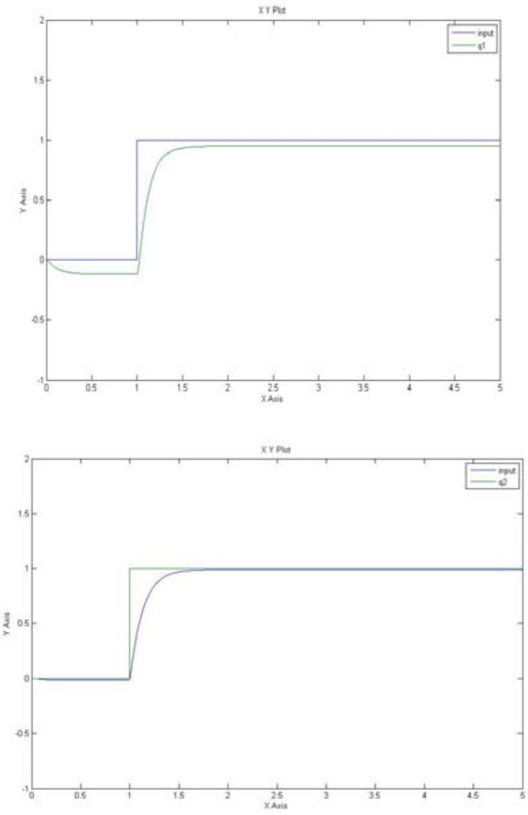

Fig. 11: the output of two link- robotic manipulator systems using Neural Network controller based on PD controller

As the third method for closed loop control of two link robotic manipulator systems the PID controller has been used, figure 12 shows the simulation result:

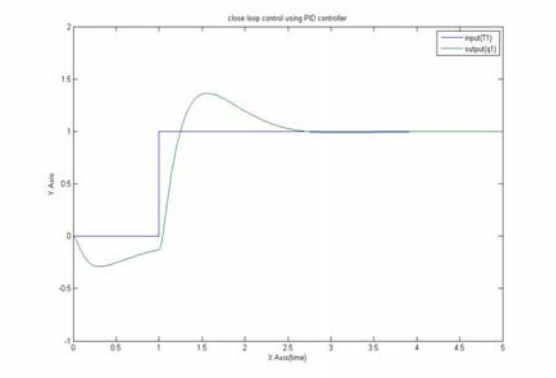

Fig. 12: the output of two link- robotic manipulator systems using PID controller

As the forth method closed loop control of two link robot the Neural Network controller based on PID controller has been used**,** figure 13 shows the simulation result:

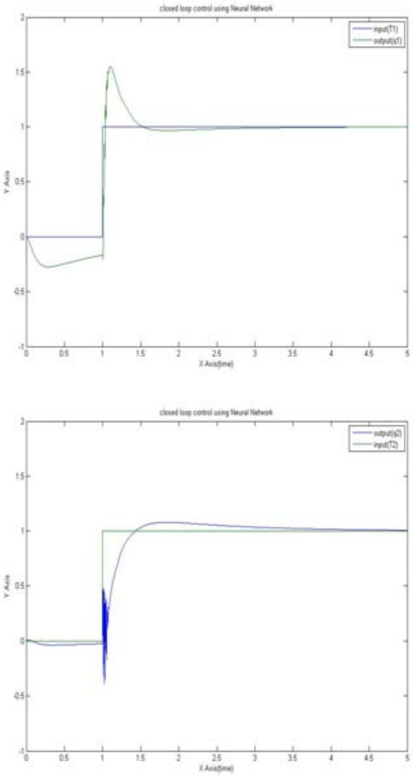

Fig. 13: the output of two link- robotic manipulator systems using Neural Network controller based on PID controller

The following table shows the comparison between these methods**:**

Table I: comparison between these methods for Link 1

Q1	Rise Time	Settling Time	Overshoot	Undershoot	Peak Time	Steady state error
PD	0.2505	1.4787	0	11.0565	5	0.0373
PID	0.1579	2.5076	36.2654	28.9523	1.5632	7.4895e-006
Neural Network& PID	0.0236	2.1768	55.1602	27.4314	1.1056	0.0016
Neural Network & PD	0.3995	2.9504	0.0028	12.6383	4.1000	0.0513
Feedback compensator by pid	0.5280	4.8189	0.3823	0	4.9493	5.9849e-007

Table II: comparison between these methods for Link 2

Q2	Rise Time	Settling Time	Overshoot	Undershoot	Peak Time	Steady state error
PD	0.1963	1.3412	0.0022	0.1013	1.9256	1.5413e-004
PID	0.3231	3.1904	16.9354	0.6991	1.8029	2.1544e-004
Neural Network & PID	0.3107	3.0221	7.2733	38.4734	1.8361	0.0057
Neural Network & PD	0.4776	2.9797	0.0036	1.3076	3.8699	0.0123

V. CONCLUSION

This paper presents 4 method for the control of Two link-robotic manipulator systems, The first method is based on Proportional-Derivative controller, The second method is based on Proportional-Integral-Derivative controller, and the thirth method used artificial Neural Network by PD controller, and the forth method used artificial Neural Network by PID controller for control of Two link- robotic manipulator systems.

REFERENCES

[1] Lianfang Tian, Jun Wang, and Zongyuan Mao, "Constrained Motion Control of Flexible RobotManipulators Based on Recurrent Neural Networks" *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 34, NO. 3, JUNE 2004*

[2] Yi, S.Y. and Chung, M., J., 1997, "A Robust Fuzzy Logic Controller for Robot Manipulators with Uncertainties", *IEEE Transactions on Systems, Man and Cybernetics*, 27, No. 4, pp. 706 – 713.

[3] umbla, K.K. and Jamshidi, M., 1994, "Control of Robotic Manipulator Using Fuzzy Logic", *Proceedings of the Third IEEE Conference on Fuzzy Systems*, Vol. 1, pp. 518-523.

[4] Bannerjee, S. and Woo, P.Y., 1993, "Fuzzy Logic Control of Robot Manipulator", *IEEE Conference on Control Applications*, pp. 87-88.

[5] Adams, J.M. and Rattan, K.S., 2001, "Genetic Multi-Stage Fuzzy PID Controller with Fuzzy Switch", *Proceedings of the IEEE International Conference on Control Applications*, pp. 323-327.

[6] Brudka, M. Pacut, A., "Intelligent robot control using ultrasonic measurements" Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference, 1999. IMTC/99.

[7] Adamiv, O. Koval, V. Turchenko, I., "Predetermined movement of mobile robot using neural networks" *Proceedings of the Second IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2003.*

[8] Petropoulakis, L., " Intelligent control using agents and fuzzy behavioural structures" : *Proceedings of the 2000 IEEE International Symposium on Intelligent Control, 2000.*

[9] Ya-Chen Hsu Guanrong Chen Malki, H.A., " Fuzzy neural adaptive controller design: with application to multiple-link robot control" *IEEE International Conference on Neural Networks, 1997.*

[10] Dongbing Gu and Huosheng Hu., "Neural Predictive Control for a Car-like Mobile Robot" *International Journal of Robotics and Autonomous Systems, Vol. 39, No. 2-3, May, 2002.*

[11] Mathew L. Moore, John T Musacchio1, Kevin M. Passino, "Genetic adaptive control for an inverted wedge: experiments and comparative analyses" *Engineering Applications of Artificial Intelligence 2001.*

[12] H. Harry Asada, *Introduction to Robotics*. Massachusetts Institute of Technology.

[13] Hagan, M.T., Menhaj, M.B., "Training feed forward networks with the Marquardt algorithm", *IEEE Transactions on Neural Networks. 5:6, page(s): 989-993, 1994.*

[14] Hagan, M.T., Demuth**,** H.B., Beale, M., *Neural networks design*. Boston, 1996.

[15] Suratgar, A.A., Tavakoli, M.B., Hosseinabadi, A., "Modified levenberg-marquardt neural network training", *Enformatika, Page(s):46-48, 2005.*

[16] Neural Network toolbox, Mtlab 2008.