Proceedings of the 2008 IEEE
International Conference on Robotics and Biomimetics
Bangkok, Thailand, February 21 - 26, 2009

A novel hybrid Fuzzy-PID controller for tracking control of robot manipulators

A. R. Norouzzadeh Ravari
Department of Research and planning
K.R.E.C Company
Kerman, abbaspour Blvd. 76175-188, IRAN
a.norouzzadeh@gmail.com

H.D. Taghirad *Member, IEEE*
Department of Electrical Engineering
K. N. Toosi University of Technology
P.O. Box 16315-1355, Tehran, Iran
taghirad@kntu.ac.ir

Abstract - **In this paper, a novel hybrid fuzzy proportional-integral-derivative (PID) controller based on learning automata for optimal tracking of robot systems including motor dynamics is presented. Learning automata is used at the supervisory level for adjustment of the parameters of hybrid Fuzzy-PID controller during the system operation. The proposed method has better convergence rate in comparison with standard back-propagation algorithms, less computational requirements than adaptive network based fuzzy inference systems (ANFIS) or neural based controllers and having the ability of working in uncertain environments without any previous knowledge of environments' parameters. The proposed controller has been successfully applied in simulation to control a 6-DOF Puma 560 manipulator using robotic toolbox, and has satisfactory results. In this simulation also, external disturbance and noise are addressed. The result of simulation has also shown that the rate of convergence and robustness of the designed controller guarantees practical stability.**

Index Terms - Learning automata, hybrid controller, robot control, Fuzzy-PID controller.

I. INTRODUCTION

The tracking control of robot manipulators, has received a lot of attention in the past decades [1], [2]. Tracking control is needed to make the joints track a desired trajectory leading to the end-effector tracking the desired path [5], [13]. Robotic manipulators are nonlinear multi-input multi-output systems subjected to uncertainties associated with their dynamics.

Different models of the robots, including actuator dynamics and the interaction between motors and joints have recently been considered in robotic control design [17], [18]. However, robot manipulators have to face various uncertainties in practical applications, such as pay load parameter, internal friction and external disturbance [25]. All the uncertain or time varying factors could affect the system control performance seriously.

Moreover, external disturbances are inevitable under practical operation conditions [3], [4]. For these reasons an efficient tracking controller for robot manipulators should be sufficient robust with respect to the modeling uncertainties, as well as external additive disturbances [12], [27].

To deal with the unknown nonlinearities and external disturbances, various control strategies have been proposed in the forms of the automatic tuning of PID controller, variable structure controller, feedback linearization, adaptive controller, intelligent controller, etc. [1]-[5].

Intelligent control approaches such as neural networks, fuzzy inference systems and neuro-fuzzy systems do not require mathematical models to be known exactly and have the ability to approximate nonlinear systems. With these features of intelligent control theory, many researchers have been attempting to use intelligent control approaches to represent complex plants and construct advanced controllers such as the model reference or direct adaptive controllers. [1], [11], [15]

Although nonlinear control methods have greatly evolved and been implemented on robotic manipulators, the proportional-integral-derivative (PID) control method is still widely accepted and used in industrial robots [8], [9]. The success of the PID control is attributed to its simple control structure, ease of design and tuning schemes and to its good performance in practice in presence of modeling uncertainties, and external disturbances [23], [26].

Neural network based methods, need a learning phase to determine the algorithm's parameters with trial and error. Also these methods are suitable for repetitive operations to have their convergence guaranteed. [13]

Beygi and Meybodi [6], have shown that learning automata based algorithms, have a faster convergence rate than standard backpropagation learning algorithms and can escape from local minima in optimization problems when standard BP fails to find the global minima.

There have been many attempts in the past to develop control techniques and algorithms to tune the PID gains. Kazemian studied a supervisory fuzzy controller to adjust a PID controller and showed fine accuracy in trajectory tracking of a two link revolute-joint robot-arm [16].

In this paper, we present a novel learning automata based hybrid fuzzy-PID controller which is very simple to implement and has a great accuracy and fine performance making it suitable for practical applications. Due to the nonlinear properties of robot dynamics, environment uncertainties and external disturbances, a learning automata based method has been chosen which can be effectively used to overcome the problem of operating in an unknown environment with enough fast convergence rate. The dynamics of robot arm and actuator with external noise are employed as a tool to study the behavior of the novel self-organizing controller.

The proposed intelligent controller doesn't need major prior learning, to find optimal parameters of the hybrid controller as in neural based controllers.

Also we have shown in simulation that the proposed controller has better performance than an optimally trained neural network based, ANFIS [14] and a conventional PID controller in trajectory tracking of a 6-DOF Puma 560 robot manipulator using robotic toolbox [20] as simulation tool and the Integral of the Absolute magnitude of the Error (IAE) as a measurement parameter.

The paper is structured as follows. In section II, the nonlinear dynamics of robot manipulators including the actuator dynamics are introduced. Section III presents the proposed controller and its stability analysis. The learning automaton is introduced in section IV. The simulation results for the 6-DOF Puma 560 robot are given in section V. Finally, in Section VI, the conclusions are presented.

II. Robot Manipulator Dynamics and Control

A. A dynamic model of robotic manipulator plus joints, driven by dc motor, including actuators is described by (1). The motion equations of a robotic manipulator with revolute joints can be expressed as [16]:

$$M'(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + d' \quad (1)$$

In which, $q,\dot{q},\ddot{q} \in R^n$ are vectors of joint positions, velocities, and accelerations; $M'(q) \in R^{n\times n}$ is the mass matrix, $C(q,\dot{q}) \in R^n$ is the vector of the centripetal and Coriolis forces; $G(q) \in R^n$ is the vector of gravitational force; $\tau \in R^n$ is the vector of torques generated at the joint side of gear box; and $d' \in R^n$ denotes the external disturbance. Note that the mass matrix M' is symmetric positive-definite.

Several fundamental properties of the robot model are given in the literature [12]:

- The matrix $\frac{1}{2}M'(q) - C(q,\dot{q})$ is skew-symmetric.
- There exists a non-negative constant k_{C1} such that for all $x, y, z \in R^n$,we have

$$\|C(x,y)\cdot z\| \le k_{C1}\|y\|\|z\| \quad (2)$$

- There exists a non-negative constant k_g such that for all $x, y, z \in R^n$,we have

$$\|G(x) - G(y)\| \le k_g\|x - y\| \quad (3)$$

Where $k_g >> \left\|\frac{\partial G}{\partial q}\right\|$ for all $q \in R^n$.

The relation between the joint position q and the motor-shaft position is given by

$$q_m = Nq \quad (4)$$

Where $N \in R^{n\times n}$ is a diagonal matrix of the gear ratios for the n joints and $N > 0$ (which means that the matrix N is positive-definite). By armature-controlled dc motors, the electrical model of the j th motor is characterized by

$$R_j i_j + L_j\frac{di_j}{dt} + K_{b_j}\frac{dq_{mj}}{dt} = u_j \;\; , j = 1,2,\ldots,n \quad (5)$$

Where R_j is the resistance of the armature circuit, L_j is the inductance of the armature circuit, K_{b_j} is the back electromotive force (EMF) constant of the motor, i_j is the armature current, q_{m_j} is the motor shaft position, and u_j is the armature input voltage.

Because of the model's complexity and nonlinearity, directly designing control laws is not easy. This situation is further compounded by the drift incurred in on-line measurements of acceleration, the frequent changes in load and model parameter, and the corruption of external disturbances.

Given a task of a continuously differentiable and uniformly bounded trajectory in the joint space q_d for which we wish the robot manipulator to follow. Therefore, we define the joint position error as

$$e = q - q_d \quad (6)$$

Tarn, Bejcay, Yun and Li [25], developed a feedback linearization plus decoupling technique based on differential geometric control theory to provide a nonlinear feedback control law for the regulation of robotic arms. However, this design is possible only while the dynamics of the robotic are well known. The system represents the robotic tracking error dynamics, which in terms of the terminology given by Garofalo and Leitmann [10], it is a nominally linear uncertain system.

In practical robotic systems, however, uncertainties due to parameter perturbations, unmodeled dynamics, and external noises are inevitable. These uncertainties deteriorate the tracking performance or even lead to system instability in the worst case. Hence, the effect of uncertainties on tracking error must be eliminated.

III. Controller Design

Nowadays, most industrial robotic manipulators are controlled by PID controllers. The wide use of robot manipulators in everyday applications is testament to the performance that can be achieved in a large variety of applications. PID controllers can minimize the steady-state error of robot manipulator but they are sensitive to parameter variation and uncertainties.

Fuzzy controllers do not require an accurate mathematical model of the robot manipulator to be effective and they have fast rise time and low overshoot.

Hybrid Fuzzy-PID controller has the advantages of both PID and fuzzy controller. Fig. 1, shows the block diagram of the proposed controller. Also a learning automata algorithm has been added to the hybrid Fuzzy-PID controller to readjust the parameters of the controller dynamically. The learning automata algorithm observes the error signal with a predefined sampling rate, and updates its internal parameters besides the values of controller's parameters.

The control law of the PID controller is given by

$$\tau = K_p q + K_v \dot{q} + K_i \int_0^t q(\delta) d\delta \tag{7}$$

Where the design matrices $K_p, K_v, K_i \; \varepsilon \; R^{n\times n}$, which are respectively called "position, velocity and integral gains", are symmetric positive definite matrices optimally selected by learning automata. From the stability analysis, we can draw a tuning procedure which is fairly simple for PID control. This method yields symmetric matrices K_p, K_v and K_i that guarantees achievement of the position control objective, locally. The procedure stems can be summarized in terms of the eigenvalues of the gain matrices as follows [18].

$$\lambda_{Max}\{K_i\} \ge \lambda_{Min}\{K_i\} > 0 \tag{8}$$

$$\lambda_{Max}\{K_p\} \ge \lambda_{Min}\{K_p\} > k_g \tag{9}$$

$$\lambda_{Max}\{K_v\} \ge \lambda_{Min}\{K_v\} > \frac{\lambda_{Max}\{K_i\}}{\lambda_{Min}\{K_p\} - k_g} \cdot \frac{\lambda_{Max}{}^2\{M\}}{\lambda_{Min}\{M\}} \tag{10}$$

This tuning procedure requires knowledge of the structure of the inertia matrix M(q) and of the vector of gravitational torques G(q) of the robot

$$k_g = n(Max_{i,j,q} \left| \frac{\partial G_i(q)}{\partial q_j} \right|) \tag{11}$$

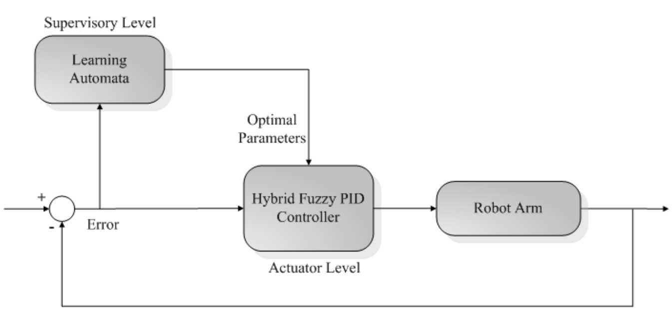

Figure 1. Block diagram of the proposed hybrid controller.

According to equations (8) to (10) the range of symmetric matrices K_p, K_v and K_i are achieved.

The fuzzy controller uses the error and its derivative as input and its single output is the control signal U. It has seven membership function of triangular shape for each input and one rule-base with 49 rules shown in Table I.

TABLE I
FUZZY CONTROL RULES

E	$\dot{E}$ NL	NM	NS	ZR	PS	PM	PL
NL	NL	NL	NL	NL	NM	NS	ZR
NM	NL	NL	NL	NM	NS	ZR	PS
NS	NL	NL	NM	NS	ZR	PS	PM
ZR	NL	NM	NS	ZR	PS	PM	PL
PS	NM	NS	ZR	PS	PM	PL	PL
PM	NS	ZR	PS	PM	PL	PL	PL
PL	ZR	PS	PM	PL	PL	PL	PL

The initial membership function of two inputs and one output of fuzzy controller is shown in Fig. 2.

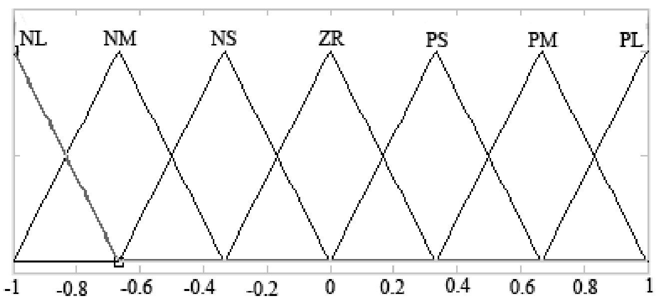

Figure 2. Membership function of error, derivative of error and control output

In the next section, the learning automaton is described which is used as an optimization method with fast convergence rate to obtain the best gains for the hybrid fuzzy-PID controller and optimize the range of fuzzified inputs and output.

IV. LEARNING AUTOMATA

a) Learning Automata Types

Classical and modern control techniques require a fair knowledge of the system, in the form of a mathematical model or statistical values such as mean and variances of the uncertainties. However, not all those assumptions on the system or uncertainties can be derived in practice. Therefore, it is necessary to obtain further knowledge of the system by observing it during operation. One approach is to view this as a problem in learning. The idea behind designing a learning system is to guarantee robust behavior without the complete knowledge, if any, of the system and/or environment. A crucial advantage of reinforcement learning compared to other learning approaches is that it requires no information about the environment except for the reinforcement signal.

The stochastic automaton attempts a solution of the problem without any a priori information on the optimal action. At each instant n, one action is selected according to the action probability distribution, the response from the environment is observed, action probabilities are updated based on that response and the procedure is repeated. A stochastic automaton acting as described to improve its performance is called a *learning automaton* [19].

The learning automata can be classified into two main groups: FALA and CALA [24]. As Fig. 3 shows, the action-set of FALA is finite, the action probability distribution is represented by an r-dimensional probability vector and is updated by the learning algorithm. CALA has a continuous action set with a probability distribution function which is updated based on the reinforcement signal as shown in Fig. 4 [6], [22].

A learning automaton may send its action to multiple environments at the same time. In that case, the action of the automaton results in a vector of responses from environments (or "teachers"). Then, the automaton has to find an optimal action that satisfies all the teachers [7]. Learning Automata can be classified into two main families: fixed structure learning automata and variable structure learning automata. In the following, the variable structure learning automata which will be used in this paper is described.

A VSLA is a quintuple < *α, β, p, T(α, β, p)* >, where *α, β,* and *p* are an action set with *r* actions, an environment response set, and the probability set *p* containing *r* probabilities, each being the probability of performing every action in the current internal automaton state, respectively. The function of *T* is the reinforcement algorithm, which modifies the action probability vector *p* with respect to the performed action and received response. If the response of the environment takes binary values, learning automata model is P-model and if it takes finite output set with more than two elements that take values

in the interval [0,1], such a model is referred to as Q-model, and when the output of the environment is a continuous variable in the interval [0,1], it is refer to as S-model.

b) Algorithm Description

In this paper we have used an S-model VSLA. Also weight factors are associated with specific teachers.

The algorithm of the discrete action set learning automata is as follows:

- The range of parameters is divided into N equal limits. Number of divisions does not have severely effect on design performance, yet it must be selected large enough.
- The probability vector is initialized to have equal probability for each action.

$$p_i^{(0)}(n) = \begin{cases} \frac{1}{N} & n = 1,2,\cdots,N \\ 0 & otherwise \end{cases} \tag{12}$$

Where $p_i^{(k)}(n)$ is the probability of selecting *n*th limit for *i*th parameter at *k*th iteration.

- The cost function is defined as a weighted sum of $\|e\|_\infty$, e_{ss} and $\int_0^T e(t)dt$. The time, T, is the simulation time and is selected large enough (here T=10_{sec}).

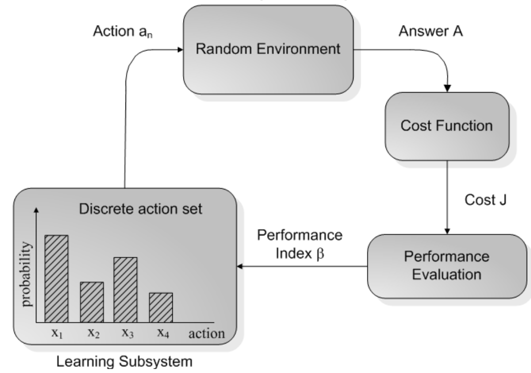

Figure 3. The discrete action set learning automata block diagram.

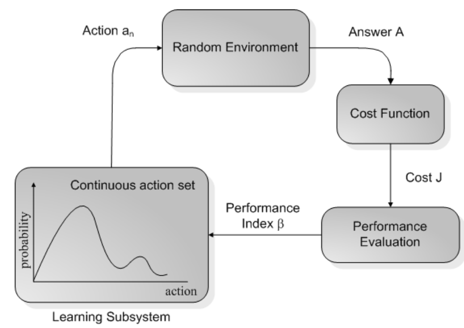

Figure 4. The continuous action set learning automata block diagram.

- At *k*th iteration, the $J^{(k)}$ (the cost function at iteration *k*th) is used to calculate the reinforcement signal β,

$$\beta^{(k)} = \min\left\{1, \max\left\{0, \frac{J_{mean} - J^{(k)}}{J_{mean} - J_{\min}}\right\}\right\} \tag{13}$$

Where $\beta^{(k)}$ is the reinforcement signal at *k*th iteration, and $J_{\min}$, J_{mean} are minimum and average of previous costs respectively. Defining reinforcement signal as (13) gives the average of costs with non-increasing behavior that guarantees convergence. The probability vector of each parameter is updated in each iteration as:

$$p_i^{(k+1)}(n) = \alpha_i^{(k)}\left(p_i^{(k)}(n) + \beta^{(k)} Q_i^{(k)}\right) \tag{14}$$

Where, $Q_i^{(k)}$ is an exponential function centered in selected limit and defined as (15).

$$Q_i^{(k)} = r_q 2^{-(n - \tilde{n}_i)^2} \tag{15}$$

in which, $\tilde{n}_i$ is a selected limit and r_q is a positive constant. $\alpha_i^{(k)}$ in (14) is a normalization factor calculated by (16).

$$\alpha_i^{(k)} = \frac{1}{\sum_{n=1}^{20} f_i^{(k)}(n) + \beta^{(k)} Q_i^{(k)}} \quad (16)$$

- The procedure is finished after a sufficient number of iterations or when the maximum of one of probability vectors is reached desirably to one.

After sufficient iterations, the probability of optimal limit for each parameter is maximized and the value of each parameter converges to the optimum value.

Here the supervisory level, using learning automata algorithm, adjusts the gains of PID controller and the width of membership functions of fuzzy controller.

V. Simulation and Results

In this section, we simulate our proposed method on the robust tracking design of a Puma560 robot manipulator as shown in Fig. 5 using robotic toolbox. [20]

Assume that the trajectory planning problem for a weightlifting operation is considered and the Puma560 robot manipulator suffers from time-varying parametric uncertainties and exogenous disturbances.

With the aim of testing in experiments the performance of the proposed controller, a rectangular reference trajectories in joint space has been selected.

TABLE II
DISTURBANCE AMPLITUDE

Joint Number	Disturbance
Joint 1	0.15Sin2t
Joint 2	0.1 Sin2t
Joint 3	0.1 Sin2t
Joint 4	0.05 Sin2t
Joint 5	0.06 Sin2t
Joint 6	0.04 Sin2t

Also, the exogenous disturbances d_1 through d_6 are assumed which their values are mentioned in Table II. [21] Obviously, the parameter uncertainties and exogenous disturbances are extremely large. Therefore, the proposed learning automata based tracking control algorithm is employed to treat this robust tracking control design.

Fig. 6 shows that the trajectory tracking error is very small and robot arm is following our desired trajectory with an acceptable and fairly small difference. After initial time, the internal variables of learning automata algorithms such as the probability vector and array of costs are sufficiently updated and the algorithm has adapted itself to the dynamics of system and environment's parameters so the performance gets better and the accuracy of path tracking increases.

Control signals are also shown in Fig. 7 which denotes that after the convergence of the parameters of learning automata algorithm, the system has a more stable control effort with appropriate magnitude.

In this intelligent controller, the accuracy of algorithm increases as the controller interacts with the system and environment. So the parameters of the algorithm can be adjusted to minimize the error of trajectory tracking without having any affect on the controller's performance.

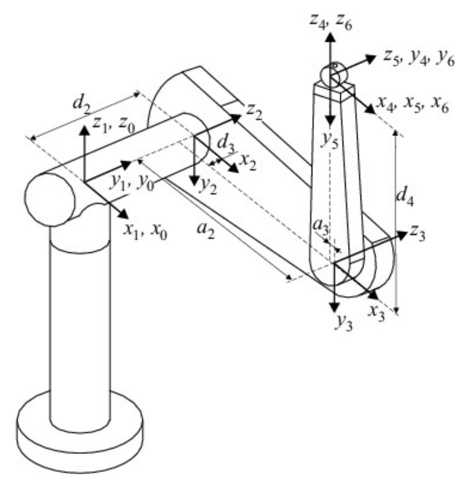

Figure 5. Puma 560 robot manipulator

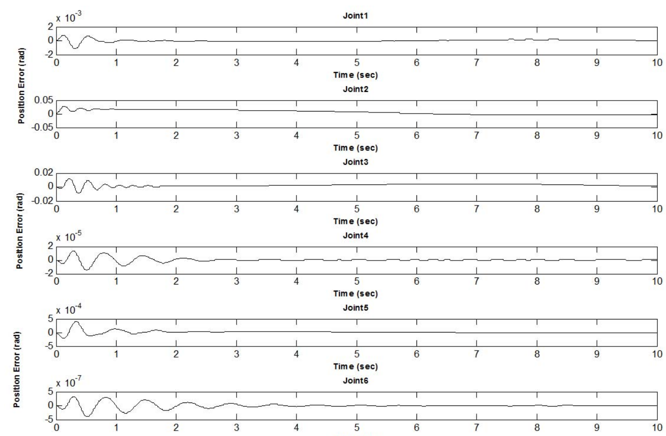

Figure 6. Simulation results of the proposed method.

Table III shows the results of IAE comparison between our novel learning automata based hybrid fuzzy-PID controller, an optimally trained neural network based PID controller a conventional one and an ANFIS controller. The results show that after the trained ANFIS controller, the learning automaton has better performance in matter of IAE value and adaptation with unknown environments.

TABLE III
THE IAE OF TRAJECTORY TRACKING SIMULATION

	Joint Number					
Controller	1	2	3	4	5	6
Automata based Hybrid Fuzzy-PID	0.24	7.37	2.06	0.0019	0.13	0.00001
ANFIS	0.09	5.14	0.83	0.0002	0.07	0.00003
Neural Based	0.34	8.23	2.72	0.009	0.16	0.0001
Conventional PID	0.56	9.87	3.35	0.018	0.28	0.0004

VI. Conclusions

In this paper it is proposed to use learning automata to tune the parameters of a hybrid fuzzy-PID controller dynamically. The fast convergence of learning automata algorithm enables the proposed controller to adaptively adjust the parameters and keep the tracking error at a low level in spite of uncertainties and external disturbances. The simulation results on a 6-DOF Puma 560 robot using the proposed algorithm show that the suitable controller parameters can be obtained in order to minimize the cost function and IAE. The control scheme does not require the robot dynamics to be known exactly and can be used for tracking of robot manipulators with external disturbances. The simulation results indicate that a desired tracking performance can be guaranteed for an uncertain robotic system via the proposed algorithm under large time-varying parameter perturbations and external disturbances. Also the simulation results show that the proposed algorithm has better performance in comparison with conventional PID controller and a neural network based PID controller in trajectory tracking and adaptation with unknown environment and parameter variation. Also it doesn't need extensive learning like ANFIS controller and has much less computational complexity.

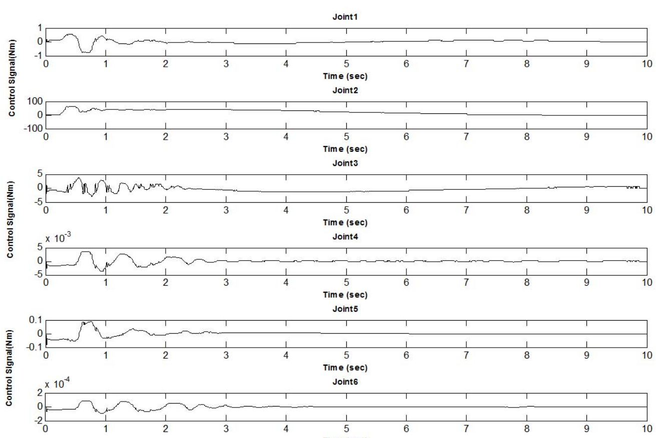

Figure 7. Control signals of thr poposed method.

REFERENCES

[1] A. Chatterjee , K. Watanabe, 2005, "An adaptive fuzzy strategy for motion control of robot manipulators", *Springer J. Soft Comput* , pp.185–193

[2] A. Green, J.Z. Sasiadek, "Dynamics and trajectory tracking control of a two-link robot manipulator," *J. Vibr. Control* 10 (10) (2004), pp. 1415–1440.

[3] A. Green, J.Z. Sasiadek, "Fuzzy and optimal control of a two-link flexible manipulator," in 2001 *Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics,* pp. 1169–1174

[4] Alessandro De Lucaa, Bruno Sicilianob, Loredana Zolloc, "PD control with on-line gravity compensation for robots with elastic joints: Theory and experiments", Automatica 41 (2005) 1809 – 1819. Available: http://www.elsevier.com/locate/automatica

[5] B. Daachi A. Benallegue, "A Neural Network Adaptive Controller for End-effector Tracking of Redundant Robot Manipulators", *J. Int. Robot System*, Springer, France, 2006, pp. 245–262

[6] Beigy H., Meybodi M.R. "A Note on Learning Automata Based Schemes for Adaptation of BP Parameters", IDEAL 2000, LNCS 1983, pp. 145-151, 2000.

[7] Cem U¨ nsal, Pushkin Kachroo, and John S. Bay., "Multiple Stochastic Learning Automata for Vehicle Path Control in an Automated Highway System", IEEE Trans Systems, MAN, And Cybernetics—part A: Systems And Humans, vol. 29, no. 1, January 1999

[8] Choi Y., Chung W. K, 2001, "On the Optimal PID Performance Tuning for Robotic Manipulators", in *Proc. IEEE Int. Conf.* Intelligent Robot and System, USA, 2001, no.3 , pp. 1656–1661.

[9] Choi Y., Chung W. K., 2004, "PID trajectory tracking control for mechanical systems", Lecture Notes in Control and Information Sciences, No. 298, Springer-Verlag: London.

[10] F. Garofalo and G. Leitmann, "Guaranteeing ultimate boundedness and exponential rate of convergence for a class of nominally linear uncertain systems," ASME J. Dynamic Systems, Measurements, Contr., vol. 111, pp. 584–588, 1989.

[11] F. L. Lewis, J. Campos, and R. Selmic, "Neuro-Fuzzy Control of Industrial Systems With Actuator Nonlinearities". Philadelphia, PA: SIAM, 2002.

[12] H.F. Ho, Y.K. Wong, A.B. Rad, "Robust fuzzy tracking control for robotic manipulators", Simulation Modelling Practice and Theory 15 (2007) pp. 801–816, 2007

[13] Jianguo Fu, Naresh K. Sinha, "An Iterative Learning Scheme for Motion Control of Robots Using Neural Networks: A Case Study", *J. Intelligent and Robotic Systems* , *Netherlands*, 1993, pp. 375-398

[14] J.-S. R. Jang. ANFIS:Adaptive-Network-Based Fuzzy Inference Systems. IEEE Transactions on Systems, Man and Cybernetics, 23(03):665-685, May 1993.

[15] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," IEEE Trans. Neural Networks, vol. 1, pp. 4–27, Mar. 1990.

[16] Kazemian. H.B. "Intelligent Fuzzy PID Controller". Foundations of Generic Optimization, Volume 2: Applications of Fuzzy Control, Genetic Algorithms and Neural Networks, 241–260.

[17] M. S. Mahmoud,1993, "Robust control of robot arms including motor dynamic", Int. *J. Control*, vol. 58, no. 4, pp. 853–873.

[18] M. W. Spong and M. Vidyasagar, Robot Dynamics and Control. New York: Wiley, 1989.

[19] Narendra, K. S., Thathachar, M. A. L., *Learning Automata: An Introduction,* Printice-Hall Inc, 1989.

[20] P.I. Corke, "A Robotics Toolbox for MATLAB", IEEE Robotics and Automation Magazine, Volume 3(1), March 1996, pp. 24-32.

[21] Ricardo Campa, Rafael Kelly, Victor Santiba, 2004, "Windows-based real-time control of direct-drive mechanisms: platform description and experiments", J. Mechatronics , Elseiver , pp. 1021–1036

[22] Santharam , G.; Sastry, P.S. and Thathachar, M.A.L. "Continuous Action set LearningAutomata for Stochastic Optimization", Journal of The Franklin Institute, vol. 331B, no. 5,pp. 607-628, 1994

[23] Tae-Yong Kuc, Woong-Gie Han, 2000, "An adaptive PID learning control of robot manipulators", J. Automatica , Elsevier, pp. 717-725

[24] Thathachar, M. A. L., Sastry, P. S., "Varieties of Learning Automata: An Overview", *IEEE Trans.* on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 32, No. 6, PP. 711-722, 2002

[25] T. J. Tarn, A. K. Bejcay, X. Yun, and Z. Li, 1991, "Effect of motor dynamics on nonlinear feedback robot arm control", *IEEE Trans. Robot. Automat.*,vol. 7, pp. 114–122.

[26] Vicente Parra-Vega, Suguru Arimoto, Yun-Hui Liu, "Dynamic Sliding PID Control for Tracking of Robot Manipulators: Theory and Experiments", *IEEE Trans.* on Robotics and Automation, vol. 19, no. 6, pp. 967-976. DECEMBER 2003

[27] W. Lia, X.G. Changa, F.M. Wahlb, Jay Farrellc, "Tracking control of a manipulator under uncertainty by FUZZY P+ID controller", J. Fuzzy Sets and Systems, Elsevier, pp. 125–137, 2001.