A New Method for Position Control of a 2-DOF Robot Arm Using Neuro – Fuzzy Controller

Jafar Tavoosi*, Majid Alaei*, Behrouz Jahani[1], Muhammad Amin Daneshwar[2]

[1]Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran
[2]Department of Electricel Engineering, Mahabad Branch, Islamic Azad University, Mahabad, Iran
mdaneshwar@yahoo.com

Abstract: Robot manipulators have become increasingly important in the field of flexible automation. So modeling and control of robots in automation will be very important. But Robots, as complex systems, must detect and isolate faults with high probabilities while doing their tasks with humans or other robots with high precision and they should tolerate the fault with the controller. This paper introduces a Neuro-Fuzzy Controller (NFC) for position control of robot arm. A five layer neural network is used to adjust input and output parameters of membership function in a fuzzy logic controller. The hybrid learning algorithm is used for training this network. In this algorithm, the least square estimation method is applied for the tuning of linear output membership function parameters and the error backpropagation method is used to tune the nonlinear input membership function parameters. The simulation results show that neuro fuzzy controller is better and more robust than the PID controller for robot trajectory control.

Keywords: Robot arm, Hybrid Learning, Neuro-Fuzzy Controller (NFC)

1. Introduction

The wider application of automatic control has developed rapidly in recent years. The reason for this is the complexity of modem plant and the constraints imposed by the increasing demand for higher quality products. Hence the design of a controller which possess learning capability becomes highly desirable. Robots are highly reliable, dependable and technologically advanced factory equipment. All commercial industrial robots have two physically separate basic elements the manipulator arm and the controller. In addition to applications like serial product lines, robots are assigned to missions like waste treatment in nuclear reactors, data collection in space and underwater tasks which can be very risky for humans. In very close years, Robots used to application with very precision such as surgical. Therefore, robots must controlled very precision, for this goal, we must use modern controller such as intelligent controller.

Arif ANKARALI, Mustafa TINKIR and Yusuf SAHIN (2010) Uses Neuro-Fuzzy controller for 3-DOF Scara Robot, they use Three adaptive networks based fuzzy logic controllers to used in control strategy as Neuro-Fuzzy controllers but third controller for wrist of robot is ineffective to track desired circular tool trajectory. These controllers are designed by training and checking data sets obtained from PID control of scara robot.

Nassim Nikpay, Mahdi Aliyari Shoorehdeli, and Mohammad Teshnehlab (2010) uses WRBF network and neuro-fuzzy network to control the hub position of Flexible Link Robot Arm. In this article, they have comparisons Neuro - Fuzzy controller, RBF controller, Fuzzy controller and LQR controller to control of hub angle And they have shown that, achieving desired performance with adaptive intelligent scheme and since fuzzy controller consumes less energy in comparison with LQR controller.

Zafer Bingul and Oguzhan Karahan (2011) introduced the PSO based tuning method for FLC and PID controller to control the given robot trajectory. The all parameters concerning the fuzzy controller and the PID controller were determined using PSO algorithm.

Kuo-Ho Su, Yih-Young Chen and Shun-Feng Su (2010) uses Neuro- Fuzzy controller for a two autonomously driven wheeled robot and they showed Neuro - Fuzzy controller is successfully applied to control the driving motor and to balance the two-wheeled robot.

Ouarda Ibrahim, Wisama Khalil (2010) presented recursive solutions for obtaining the inverse and direct dynamic models of hybrid robots. The hybrid structure is made up of n parallel modules, which are serially connected to a fixed base.

Srinivasan Alavandar, M.J. Nigam (2008) uses ANFIS to Inverse Kinematics Solution of 3-DOF Planar Robot. In this paper, They illustrated the ANFIS is able to identification and control of 2-DOF and 3-DOF robot manipulator and trained ANFIS can be utilized to provide fast and acceptable solutions of the inverse kinematics of robots.

In this paper, we uses Takagi-Sugeno Type Neuro-Fuzzy Network with hybrid learning algorithm for identification of robot then trained Neuro – Fuzzy Network is used to Neuro- Fuzzy controller for position control of 2-DOF robot arm.

Dynamic model of the 2-DOF robot arm

The dynamical analysis of the robot investigates a relation between the joint torques/forces applied by the actuators and the position, velocity and acceleration of the robot arm with respect to the time. Robot manipulators have complex non-linear dynamics that might make accurate and robust control difficult. Therefore, they are good examples to test performance of the controllers.

the 2-DOF robot shown in Fig. 1 was selected as an example problem. The dynamic equations of the serial robot are usually represented by the following coupled non-linear differential equations:

$$T = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) + F(\dot{\theta}) \qquad (1)$$

where T is Torque and Θ is position vector $\Theta = [\theta_1, \theta_2, \cdots, \theta_n]^T$, and $M(\theta)$ is the $n \times n$ inertia matrix, $V(\theta, \dot{\theta})$ is the $n \times 1$ coriolis/centripetal matrix and $G(\theta)$ is the $n \times 1$ gravitational toques and $F(\theta)$ is the $n \times 1$ vector of dynamic and static friction forces

State space equation of robot arm with $x_1 = \theta_1$, $x_2 = \theta_2$, $x_3 = \dot{\theta}_1$, $x_4 = \dot{\theta}_2$ can be written as follows

$$\dot{x}_1 = x_3$$
$$\dot{x}_2 = x_4$$

$$\dot{x}_3 = \frac{1}{\Delta}[(l_2^2 m_2)(\tau_1 + 2\alpha s_2 x_3 x_4 + \alpha x_4^2 - \beta c_1 - \gamma c_{12} - r_1 x_3) \qquad (2)$$
$$- (l_2^2 m_2 + l_1 l_2 m_2 c_2)(\tau_2 - \alpha s_2 x_3^2 - \gamma c_{12} - r_2 x_4)]$$

$$\dot{x}_4 = \frac{1}{\Delta}[-(l_2^2 m_2 + l_1 l_2 m_2 c_2)(\tau_1 + 2\alpha s_2 x_3 x_4 + \alpha x_4^2 - \beta c_1 - \gamma c_{12} - r_1 x_3)$$
$$+ (l_1^2(m_1 + m_2) + l_2^2 m_2 + l_1 l_2 m_2 c_2)(\tau_2 - \alpha s_2 x_3^2 - \gamma c_{12} - r_2 x_4)]$$

where $c_1 = \cos(\theta_1)$ and $c_{12} = \cos(\theta_1 + \theta_2)$. m_1, m_2 are masses, l_1, l_2 are lengths and r_1, r_2 are Friction coefficients of joints, and $\Delta = l_1^2 l_2^2[(m_1 + m_2)m_2 - m_2^2 c_2^2]$, $\alpha = l_1 l_2 m_2$, $\beta = l_1(m_1 + m_2)g$, $\gamma = l_2 m_2 g$
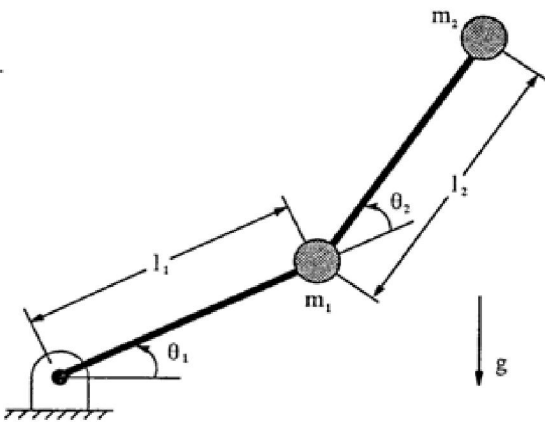

Fig 1. 2-DOF Robot Arm

Fuzzy Neural Network

In this study, adaptive network based fuzzy logic controller is applied for position control of robot arm. Established adaptive network based fuzzy inference system (ANFIS) uses a hybrid learning algorithm to identify parameters of Sugeno-type fuzzy inference system. It applies a combination of the least squares method and the back propagation gradient descent method for training fuzzy inference system (FIS) membership function parameters to emulate a given training data set. The control algorithm proposed to be used in control of the 2-DOF Robot Arm is shown in Fig. 1

Before adaptation to the system, the fuzzy logic controller rule-base is optimized offline by using artificial neural network (ANN). These controllers are designed by training and checking data sets that are obtained from PID control of the system.

First of all hierarchical PID controllers applied to the 2-DOF Robot Arm and then their inputs and outputs data are obtained to set up the adaptive network based fuzzy inference system (ANFIS). After training the network, fuzzy inference system structure is established. The network base is trained offline. Number of membership functions and type of membership functions are selected.

Fig.2 illustrate Takagi-sugeno type of Fuzzy Neural Network. In this model, output is linear equation of inputs. In Fig.2 we have two rule:

if x is A_1 and y is B_1 then
$$f_1 = p_1 x + q_1 y + r_1$$

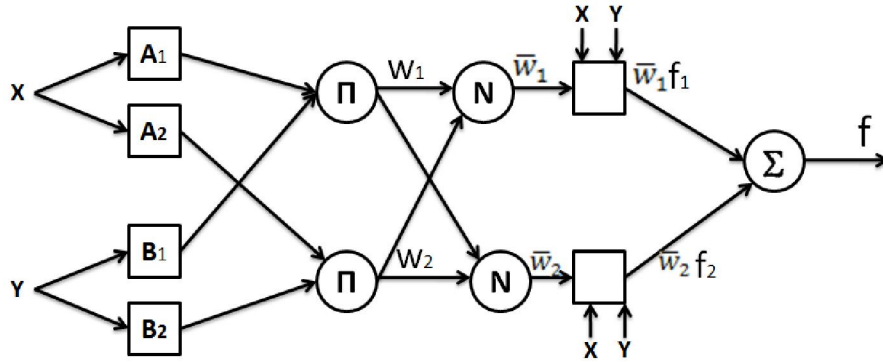if x is A_2 and y is B_2 then
$$f_2 = p_2 x + q_2 y + r_2$$

Fig.2 structure of T-S Fuzzy Neural Network

Neuro – Fuzzy Controller for 2-DOF Robot Arm

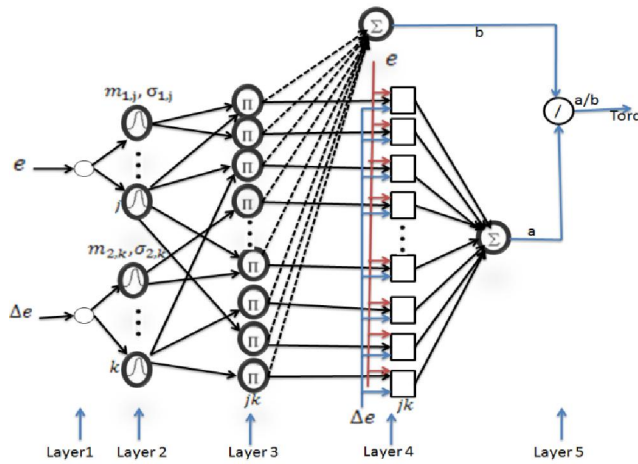In this paper we have used the structure of Takagi-sugeno type of Fuzzy Neural Network in Figure 3.

Fig 3. structure of used Fuzzy Neural Network

In Fig.3, e and Δe are inputs. In layer 1, each node performs a fuzzy set and the Gaussian function is adopted as membership function

$$O_{1,i}^1 = \mu_{A_i}(e) = exp\left[-\left(\frac{e - m_{1,i}}{\delta_{1,i}}\right)^2\right] \quad (3)$$

$$i = 1,2,\ldots,j$$

$$O_{2,l}^1 = \mu_{B_i}(\Delta e) = exp\left[-\left(\frac{\Delta e - m_{2,l}}{\delta_{2,l}}\right)^2\right] \quad (4)$$

$$l = 1,2,\cdots,k$$

The output of layer 2 as follows:

$$O_{il}^2 = w_{il} = \mu_{A_i} * \mu_{B_i} \quad (5)$$

The output of layer 3 as follows:

$$O_{il}^3 = w_{il}f_{il} = w_{il}(p_{il}e + q_{il}\Delta e + r_{il}) \quad (6)$$

where p_{il}, q_{il} and r_{il} are linear output parameters, that are determined by least square method.

In layer 4,

$$a = \sum_i \sum_l w_{il}f_{il} \quad (7)$$

$$b = \sum_i \sum_l w_{il} \quad (8)$$

$$O_0^4 = Torque = \frac{a}{b} \quad (9)$$

In the Fuzzy Neural Network, the aim of learning algorithm adjust the linear output parameters, p_{il}, q_{il} and r_{il} and the mean of the Gaussian function $m_{1,j}$ and $m_{2,k}$ and the variance of the Gaussian function $\sigma_{1,j}$ and $\sigma_{2,k}$.

This paper uses hybrid learning algorithm for learning Fuzzy Neural Network. The Hybrid Learning Algorithm is a combination of least square and back propagation method. we use least square method for estimate linear output parameters, and back propagation method for adjust gaussian nonlinear parameters. The equation of least square method as follows:

$$y = \beta_1 f_1(U) + \beta_2 f_2(U) + \cdots\cdots + \beta_n f_n(U) \quad (10)$$

where u is input vector, β_i are unknown parameters, f_i are known function and y is output. The equation (10) can be rewrite as follows

$$A\beta = Y \quad (11)$$

where A is $d \times n$ matrix, That d is the number of training data and n is number of unknown parameters in equation (10). Estimation of β is $\hat{\beta}$ and is given by

$$\hat{\beta} = (A^T A)^{-1} A^T Y \quad (12)$$

By using equations (10-12), output linear parameters are obtained. Then by using back propagation method, as follows

$$\alpha_{new} = \alpha_{old} + \Delta\alpha \quad (13)$$

where α is the parameter that to be optimized. In equation (13), $\Delta\alpha$ is given by

$$\Delta\alpha = -\eta\frac{\partial E}{\partial \alpha} \quad (14)$$

where E is difference between desired output T_m and network output T

$$E(d) = \frac{1}{2}\left(T_m(d) - T(d)\right)^2 = \frac{1}{2}e^2(d) \quad (15)$$

by using beck propagation algorithm, The error expression for Layer 4 as follows

$$\delta_0^4 = -\frac{\partial E}{O_0^4} = e \quad (16)$$

The error expression for Layer 3 as follows

$$\delta_{jk}^3 = -\frac{\partial E}{\partial O_{jk}^3} = \frac{\partial E}{O_0^4}\frac{\partial O_0^4}{\partial O_{jk}^3} = \frac{1}{b}\delta_0^4 \quad (17)$$

The error expression for Layer 2 as follows

$$\delta_{jk}^2 = -\frac{\partial E}{\partial O_{jk}^2} = -\frac{\partial E}{\partial O_{jk}^3}\frac{\partial O_{jk}^3}{\partial O_{jk}^2} = \delta_{jk}^3 f_{jk} \quad (18)$$

The error expression for Layer 1 as follows

$$\begin{cases} \delta_{1,j}^{II} = -\dfrac{\partial E}{\partial O_{jk}^{II}} = \sum_k \delta_{jk}^{III} w_{jk} \\ \delta_{2,k}^{II} = -\dfrac{\partial E}{\partial O_{jk}^{II}} = \sum_j \delta_{jk}^{III} w_{jk} \end{cases} \quad (19)$$

the equations 13-20 are the equation of update the antecedent parameters that they are the mean and the variance of gaussian function.
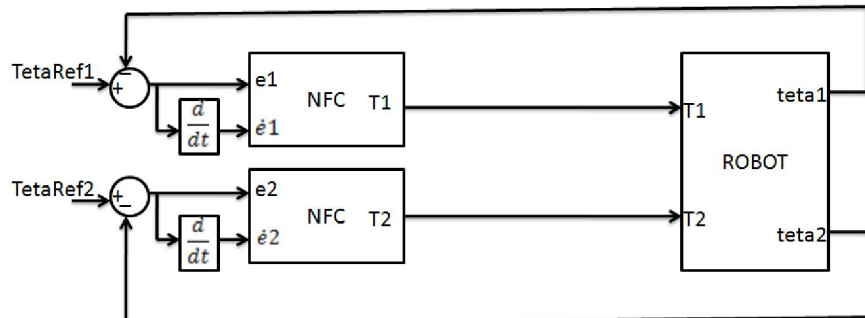
$$\Delta m_{1,j}(k) = -\frac{\partial E}{\partial m_{1,j}} = -\frac{\partial E}{\partial \mu_{1,j}}\frac{\partial \mu_{1,j}}{\partial m_{1,j}} = \delta_{1,j}^{II}\frac{2(x_{1,j} - m_{1,j})}{(c_{1,j})^2} \quad (20)$$

$$\Delta m_{2,k}(k) = -\frac{\partial E}{\partial m_{2,k}} = -\frac{\partial E}{\partial \mu_{2,k}}\frac{\partial \mu_{2,k}}{\partial m_{2,k}} = \delta_{2,k}^{II}\frac{2(x_{2,k} - m_{2,k})}{(\sigma_{2,k})^2} \quad (21)$$

$$\Delta\sigma_{1,j}(k) = -\frac{\partial E}{\partial \sigma_{1,j}} = -\frac{\partial E}{\partial \mu_{2,k}}\frac{\partial \mu_{2,k}}{\partial \sigma_{1,j}} = \delta_{2,k}^{II}\frac{2(x_{1,j} - m_{1,j})^2}{(c_{1,j})^3} \quad (22)$$

$$\Delta\sigma_{2,k}(k) = -\frac{\partial E}{\partial \sigma_{2,k}} = -\frac{\partial E}{\partial \mu_{2,k}}\frac{\partial \mu_{2,k}}{\partial \sigma_{2,k}} = \delta_{2,k}^{II}\frac{2(x_{2,k} - m_{2,k})^2}{(\sigma_{2,k})^3} \quad (23)$$

$$\begin{cases} m_{1,j}(k) = m_{1,j}(k-1) + \mu_m\Delta m_{1,j}(k) \\ m_{2,k}(k) = m_{2,k}(k-1) + \mu_m\Delta m_{2,k}(k) \end{cases} \quad (24)$$

$$\begin{cases} \sigma_{1,j}(k) = \sigma_{1,j}(k-1) + \mu_\sigma\Delta\sigma_{1,j}(k) \\ \sigma_{2,k}(k) = \sigma_{2,k}(k-1) + \mu_\sigma\Delta\sigma_{2,k}(k) \end{cases} \quad (25)$$

Fig 4. Simulink model of the neuro-fuzzy controller and robot

Simulation

The block diagram of robot arm and neuro-fuzzy controller is shown in Fig 4. In this block, two trained fuzzy neural networks are used that one of them is utilized for control of θ_1 and another is utilized for control of θ_2. The initial value of θ_1 and θ_2 are [-30 -40] (deg) and the final value of θ_1 and θ_2 are [30 40] (deg), respectively.

Two Neuro - Fuzzy controllers are effective to track desired trajectory which are designed for two joints. These controllers have five membership functions and gaussian type membership functions are used in their fuzzification process. The rule bases of controllers are made of 25 rules and these rules are determined by fuzzy neural network (FNN).

The desired position and the actual position for joints 1 and 2 are shown in Figs. 5 and 6, respectively.
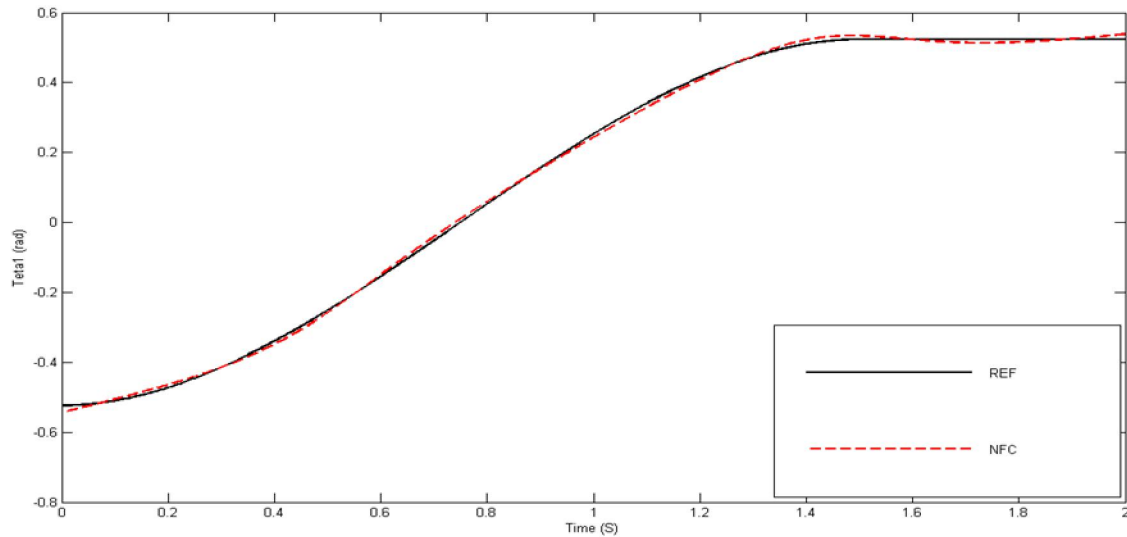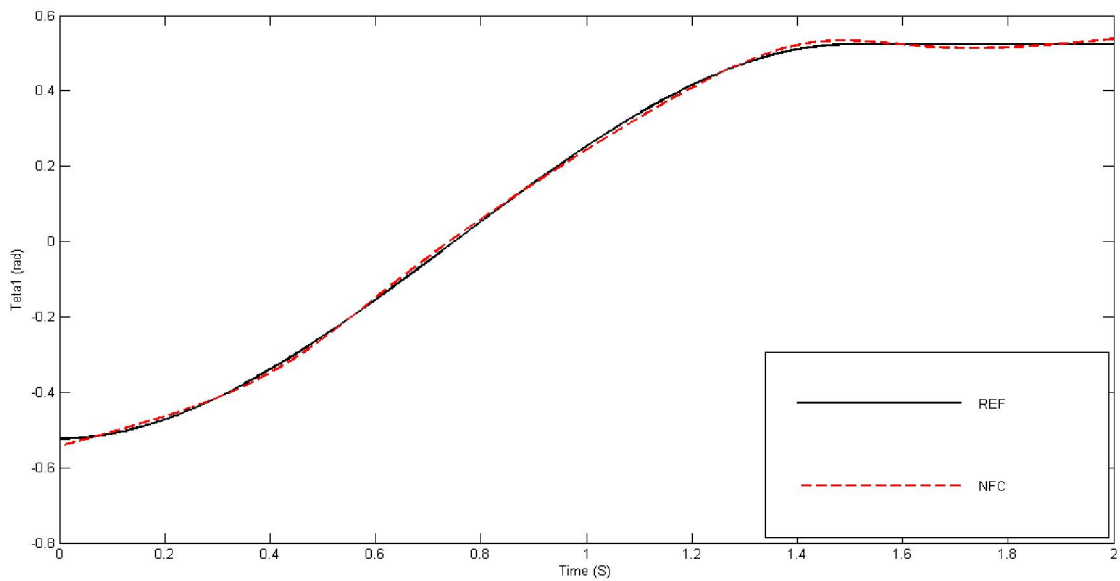

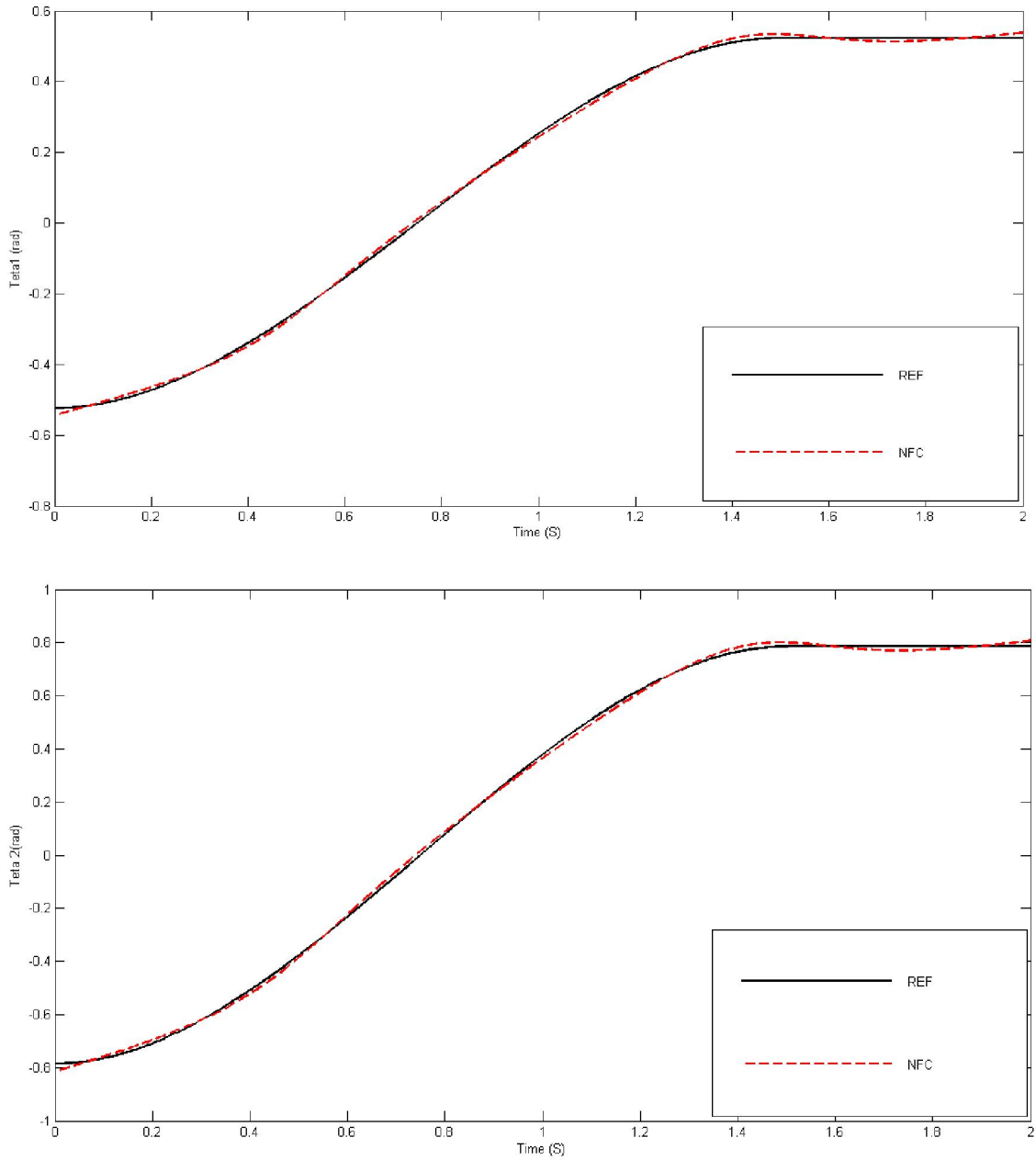**Fig. 5.** The desired position and the actual position for joints 1

Fig. 6. The desired position and the actual position for joints 2

The desired position and the actual position of the neuro-fuzzy and PID controller for joints 1 and 2 are given in Figs 7 and 8. These results show that performances of neuro-fuzzy controllers are better than PID controllers' performances over 2-DOF robot arm. Neuro- fuzzy controller has fast response and small errors for different rise functions over trajectory control of robot arm.
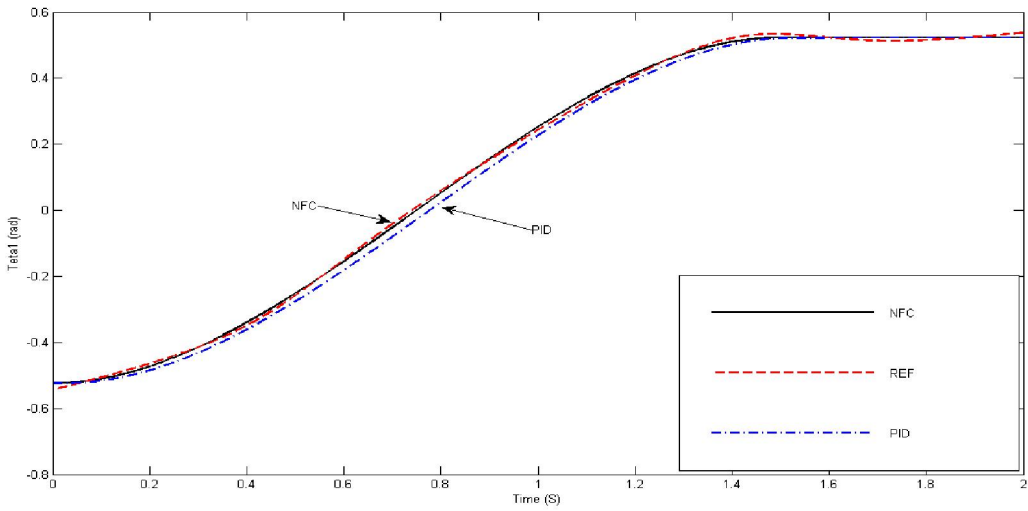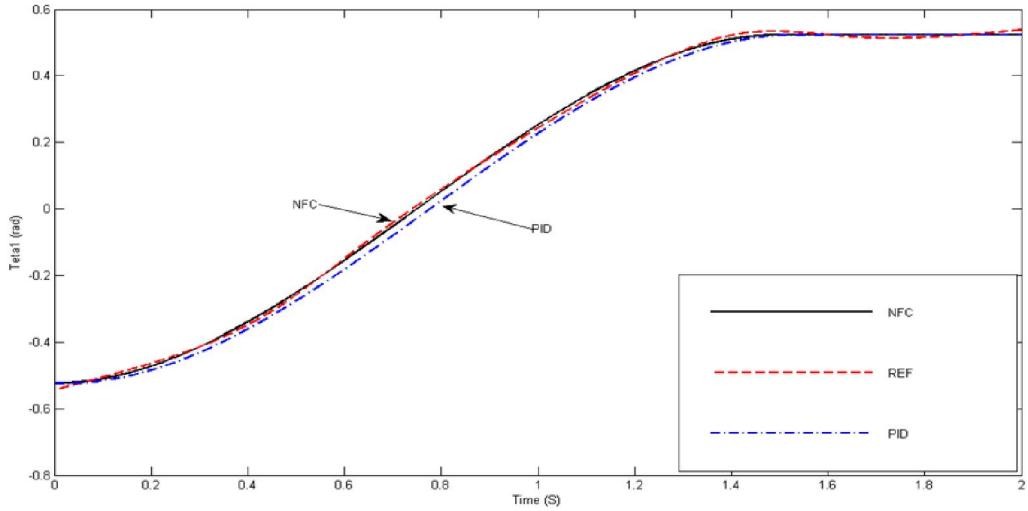
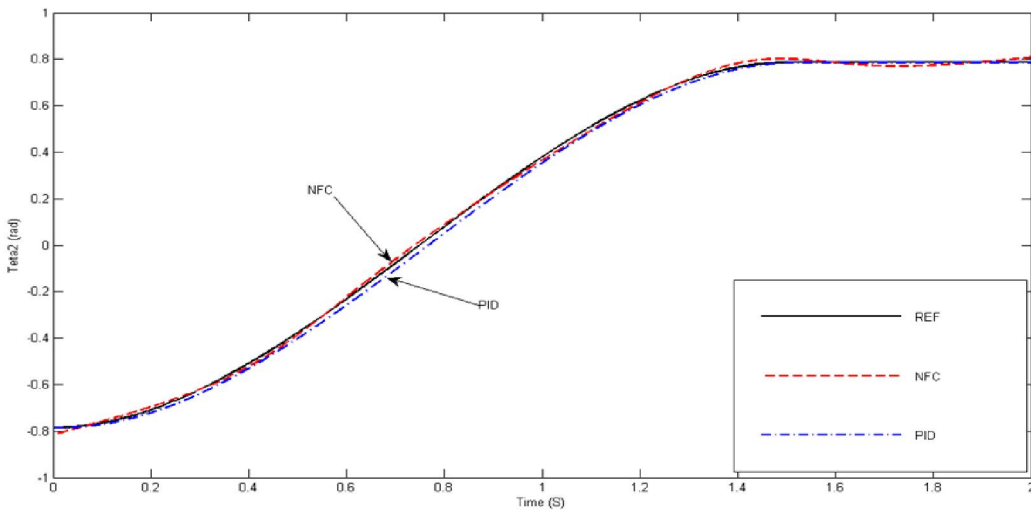**Fig. 7**. The desired position and the actual position of the neuro-fuzzy and PID controller for joints 1 (θ_1)

Fig. 8. The desired position and the actual position of the neuro-fuzzy and PID controller for joints 2 (θ_2)

Conclusions

This paper introduced two different Neuro–Fuzzy controllers are designed for trajectory tracking control of a robot arm. Neuro– Fuzzy controllers has provided best results for control of robotic manipulators as compared to the conventional control strategies. From the simulation results, the joint-position tracking responses can be controlled to follow the reference trajectories accurately under a wide range of operating conditions and the occurrence of uncertainties. Simulation results show that the Neuro– Fuzzy controller can achieve better accuracy and has less or no deviation from the trajectory compared to the PID controller. It is verified that the Neuro– Fuzzy controller has better control performance in robot trajectory control.

References

1. ANKARALI. A, TINKIR. M and SAHIN. Y (2010), Neuro-Fuzzy Trajectory Control of A Scara Robot. The 2nd International Conference on Computer and Automation Engineering (ICCAE), pages: 298- 302.
2. Alavandar. S, Nigam. M.J (2008), Inverse Kinematics Solution of 3DOF Planar Robot using ANFIS, Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844Vol. III (2008), 2008, pages. 150-155.
3. Bingul. Z and Karahan. O (2011), A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control, Journal of Expert Systems with Applications, Volume 38, Issue 1, January 2011, Pages: 1017-1031.
4. Kuo-Ho Su, Yih-Young Chen and Shun-Feng Su (2010), Design of neural-fuzzy-based controller for two autonomously driven wheeled robot,, Journal of Neurocomputing Volume 73 Issues 13-15, August 2010, Pages: 2478-2488.
5. Nikpay. N, Aliyari. Sh. M, and Teshnehlab. M (2010), Adaptive Intelligent Control of Flexible Link Robot Arm, 8th International Symposium on Intelligent Systems and Informatics September 10-11, 2010, Subotica, Serbia. pages: 653- 658.
6. Ouarda. I, Wisama. Kh (2010), Inverse and direct dynamic models of hybrid robots, Mechanism and Machine Theory, Volume 45, Issue 4, April 2010, Pages 627-640.

9/25/2017