International Journal of Systems Signal Control and Engineering Application 7 (4-6): 61-69, 2014
ISSN: 1997-5422

Fuzzy Control of Flexible Serial Robots with Neural Tuner

Seyed Mohammad Reza Faritus and Hadi Homaei
Department of Mechanical Engineering, University of Shahrekord, Shahrekord, Iran

Abstract: In this study, trajectory tracking control of planar serial robots with the last flexible arm is studied. The EOMs are derived using Lagrangian mechanics and the assumed modes method. The robot has a fuzzy controller with neural tuner. The control system consists of a fuzzy logic controller in the feedback configuration with error and change in error of the joints angular as input variables. Set parameters of membership functions of inputs in fuzzification and output in defuzzification are fuzzy control challenges. Utilizing a three-layer perceptron neural network a new method to estimate the on-line self-tuning parameters of the membership functions is presented. In this method, symmetric triangular membership functions are used in fuzzification and defuzzification units. Each of them is function of a productive parameter which is calculated on-line using neural network. The network inputs depending on which membership function is set, can be deformation, joint angular error or its derivative. The back propagation learning algorithm is used to update the network weights and the biases. To validate the proposed method simulation is done and the results are investigated.

Key words: Flexible robot, fuzzy logic controller, perceptron neural network, network, propagation learning algorithm

INTRODUCTION

Utilization of rigid and massive components to mitigate vibration and improve the accuracy of robots results in higher operating and manufacturing costs. Moreover, high inertia and vibration during rapid movement would limit their operating speeds. Using light arms not only improves efficiency but also increases safety and speed of operation. Other advantages include need for smaller actuators, achieve higher maneuverability and easy transportation, assembly and installation. However, lighter and more slender arms are more deformable. This would make target tracking more challenging and would introduce residual vibration that needs to be dissipated after the target is met (Yue *et al*., 2002).

The vibration mitigation of robots without comprising their response time and load capacity is an interesting subject for the researchers (Karami *et al*., 2014). The design of high performance robots using the traditional controllers requires an accurate dynamic model. Inversion, adaptive, robust and sliding control strategies are among some traditional methods (De Luca *et al*., 1989; Lee *et al*., 2001; Caracciolo *et al*., 2005; Zhang *et al*., 2004). The derivation of the equations of motion of flexible robots often requires complicated calculations. These equations are either too bulky or not accurate due to simplifying assumptions. The dimensional model of such robots, because of their nonlinear nature, is infinite. Also non-minimum phase of such systems leads to under-actuated conditions. Moreover, errors in the estimation of the system characteristics, time-varying nonlinearities such as those due to friction forces or moments, actuator saturation effects and sometimes uncertainties in nonlinearity sources make traditional control strategies ineffective in designing an optimum system with low vibration and high operating speed (Qiu *et al*., 2015; Korayem *et al*., 2014; Alam and Tokhi, 2008).

Fuzzy control is an intelligent method that its important advantages include the ability to use the knowledge of expert in the system control and its robustness due to insensitivity to changes in the environment (Matia *et al*., 2014). Although, fuzzy control techniques do not require an accurate dynamic model, they require a mathematical model to predict the system response in linguistic terms as fuzzy role base uses the expert knowledge (Lochan and Roy, 2014). The use of fuzzy logic in robot controls was started from 80s and in 90s was extensively used to study flexible robots (Wu and Tzou, 1993; Wang and Li, 1997). Huang and Lee (2000) presented a self-organizing FLC for the position control of a robot.

Mirshekaran *et al*. (2013) used a PID type sliding mode modified fuzzy controller in which the gains were adjusted using fuzzy logic to reduce the tip vibration.

Corresponding Author: Hadi Homaei, Department of Mechanical Engineering, University of Shahrekord, Shahrekord, Iran

Mallikarjunaiah and Reddy (2013) used an adaptive neuro-fuzzy which consisted of two subsystems. One of them to control desired position and the other one to reduce vibration. Abdullahi *et al*. (2013) designed a model based pole placement and fuzzy FLC controller. They comprised trjectory tracking and vibration responses between the two methods. Sahamijoo *et al*. (2015) used PID like FLC controller with three inputs P, D and I. They used N linguistic variables to describe the system's behavior. They compared their results with PID controller using simulation. Tian and Collins (2005) used a neuro-fuzzy control scheme which consisted of fuzzy controllers and two neural networks. One of which was used for identification and prediction of the output and the other for tuning of the gain factors in the PD controller of the fuzzy role base. They validated their approach by experiments.

One of which was used for identification and prediction of the output and the other for tuning the gain factors of the PD controller of the fuzzy role base. They validated their approach by experiments. Meza *et al*. (2012) improved the performance of a robot using a self-tuning fuzzy controller by updating the gain factors of the PID controller of the robot based on the its actual state.

The rules and the membership functions of fuzzy controllers are not usually modified after they are selected. In this study, a novel approach is presented which utilizes neural network for tuning fuzzification and defuzzification membership function parameters of PD-type fuzzy controller.

In the study, the proposed method is described. Then, the equations of motion of the robot are derived using Lagrangian mechanics and assumed mode methods (Meirovitch, 1967). Fuzzy controllers with membership functions that are tuned using neural network are presented and finally a simulation is performed and the results are discussed.

Proposed approach: To build PD and PID-type fuzzy controllers, most researches have focused on tuning P, D or I gain factors by employing neural networks or evolutionary algorithms (Mirshekaran *et al*., 2013; Mallikarjunaiah and Reddy, 2013; Meza *et al*., 2012). Nevertheless, intelligent fuzzification and defuzzification sections have rarely been the subject of studies. This paper focuses particularly on building intelligent membership functions for those sections. To this end, symmetric triangular membership functions are used where it is a function of an individual productive parameter. The membership functions would depend on the selection and the design of this parameter. Three layer perceptron neural networks are used to build an intelligent productive parameter. To achieve our goals, reducing vibration and error in joints, we consider the cost function as a sum of the squares of the error, the derivative of error and the elastic deformation. We consider three networks for each arm, one network for the membership function of error, another one for the change in the error in the fuzzification and lastly one network for the membership function of the output torque in the defuzzification. The error, its derivative and the elastic deformation can be considered as inputs depending on the network application. Then, an adaptive weight learning algorithm is specified by the generalized delta rule (Rumelhart and McClelland, 1986). In this method, the weight and bias vectors are updated in the direction of negative gradient of the quadratic cost function for its minimization.

MATERIALS AND METHODS

Application: In this study, the application of the proposed approach is presented. Figure 1 demonstrates an N armserial robot in which the last arm is flexible. The hub mass, generalized torque and revolving angle at joint i are denoted by M_i, τ_i and θ_i, respectively. M and $\psi(x_N, t)$ are the payload mass and the elastic lateral displacement on the last arm that is x_N apart from point O_N, respectively. The XOY is an inertial coordinate frame with its origin at the hub and $x_iO_iy_i$ are moving coordinate systems with their origins associated with the ith joint.

The flexible beam is a Euler-Bernoulli Beam in which the effects of shear deformations and rotational inertia are

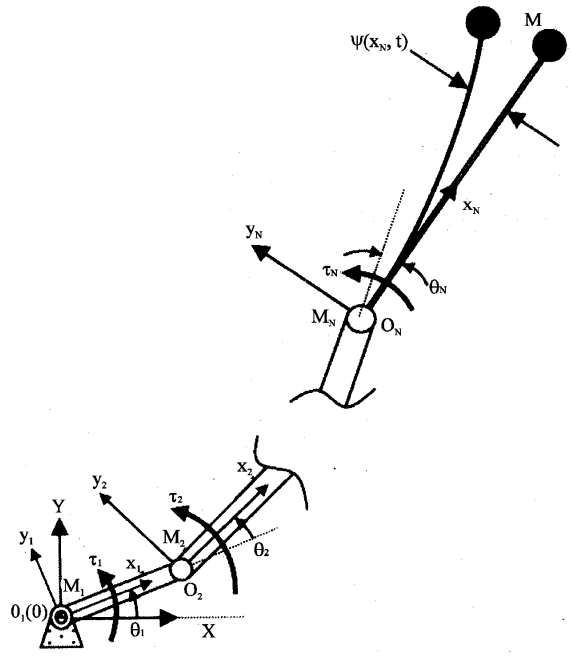

Fig. 1: Flexible serial robot with load

ignored. Since, the arms and the hubs are located on the horizontal plane, we can assume the deformations are also horizontal and the effects of gravity can be ignored.

Utilizing the assumed modes method, the bending deformation of the flexible beam is considered as a linear combination of the natural modes and the time-dependent harmonic functions. Therefore, we have:

$$\psi(x_N,t)=\sum_{j=1}^{M}\phi_j(x_N)\,W_j(t) \tag{1}$$

where, $\phi_j(x_N)$ are the normal modes and $W_j(t)$ is the time-dependent function. The value of the normal modes for the flexible arm with a payload at one end is specified as follows (Rao, 2007):

$$\begin{aligned}\phi_j(x_N) = {} & (\cos\beta_j x_N-\cosh\beta_j x_N)- \\ & \frac{\cos\beta_j l_N-\cosh\beta_j l_N}{\sin\beta_j l_N-\sinh\beta_j l_N}(\sin\beta_j x_N-\sinh\beta_j x_N)\end{aligned} \tag{2}$$

where, β_j are the first vibration frequencies of the flexible beam which are obtained using Eq. 3 as follows:

$$\begin{aligned}&1+\cos\beta_j l_N\times\cosh\beta_j l_N-\frac{M}{m_N}\beta_j l_N\times \\ &(\sinh\beta_j l_N\times\cos\beta_j l_N-\sin\beta_j l_N\times\cosh\beta_j l_N)=0\end{aligned} \tag{3}$$

where, l_N and m_N denote the length and the mass of the flexible arm, respectively. Also ω_j are natural frequencies which are obtained as follows:

$$\omega_j=\sqrt{\frac{E_N I_N}{\rho_N A_N}}\beta_j^2 \tag{4}$$

The total kinetic energy of the robot would be the summation of the kinetic energy of the arms, with those of the hubs and the payload:

$$\begin{aligned}T=\frac{1}{2}\sum_{i=1}^{N}\int_0^{L_i}&\rho_i A_i R_i^T R_i dx_i+\frac{1}{2}mR_m^T R_m+ \\ &\frac{1}{2}\sum_{i=1}^{N}(m_{M,i}R_{M,i}^T R_{M,i}+I_{M,i}\omega_{M,i}^2)\end{aligned} \tag{5}$$

Where:

$R_{M,i}$, $\omega_{M,i}$, $m_{M,i}$ and $I_{M,i}$ = Position vector, angular velocity, mass and the moment of inertia of motors at joint i, respectively

R_i and R_m = The position vector on the ith arm and the payload to initial frame OY

ρ_i, A_i = The density and the cross section of the arms at joint i, respectively

The potential energy due to the elastic deformation in the flexible arm is obtained as follows:

$$U_N=\frac{1}{2E_N I_N}\int_0^{l_N}\left[\frac{\partial^2\psi(S,t)}{\partial S^2}\right]^2 dS \tag{6}$$

After substitution Eq. 1 into Eq. 6, U_N can be obtained and the mode shapes $\phi_j(S)$ are calculated using Eq. 2. The total potential energy of the robot in absence of gravity forces would be U_N.

The equations of motion can be obtained using Lagrangian mechanics. The virtual work due to the torque τ_i would equal to $W_i = \tau_i\,\delta\theta_1$, therefore, the equations of motion can be calculated using Eq. 7 and 8:

$$T\frac{d}{dt}\left(\frac{\partial\mathfrak{L}}{\partial\dot{\theta}_i}\right)-\frac{\partial\mathfrak{L}}{\partial\theta_i}=\tau_i,\quad (i=1,2,...,N) \tag{7}$$

$$\frac{d}{dt}\left(\frac{\partial\mathfrak{L}}{\partial\dot{W}_k}\right)-\frac{\partial\mathfrak{L}}{\partial W_k}=0,\quad (k=1,2,...,M;\ \mathfrak{L}=T-U) \tag{8}$$

Where:

$\mathfrak{L}$ = The Lagrangian

θ_i and W_k = The generalized coordinates

After simplifying Eq. 7 and 8, the equations of motion in matrix form would be as follows:

$$\begin{bmatrix}B_{\theta\theta}(\theta,W) & B_{\theta W}(\theta,W)\\ B_{\theta W}^T(\theta,W) & B_{WW}(\theta,W)\end{bmatrix}\begin{bmatrix}\ddot{\theta}\\ \ddot{W}\end{bmatrix}+ \begin{bmatrix}0\\ KW\end{bmatrix}+\begin{bmatrix}h_\theta(\theta,W,\dot{\theta},\dot{W})\\ h_W(\theta,W,\dot{\theta},\dot{W})\end{bmatrix}=\begin{bmatrix}\tau\\ 0\end{bmatrix} \tag{9}$$

where, $W\in R^{M\times 1}$, $B_{\theta\theta}\in R^{N\times N}$, $B_{\theta W}\in\Re^{N\times M}$, $B_{WW}\in\Re^{M\times M}$, $h_\theta\in\Re^{N\times 1}$ and $h_W\in\Re^{M\times 1}$; K is the stiffness matrix with dimension M and $\tau\in\Re^{N\times 1}$ is the control torque vector of the hub. Also $\theta,\dot{\theta},\ddot{\theta}\in\Re^{N\times 1}$ are angular position, velocity and acceleration, respectively. The superscript T in $B^T_{\theta W}$ denotes the transpose matrix.

Fuzzy controller: The hub angle error and its change rate are two inputs for the PD-type fuzzy controllers. If $\theta(r)$ is the hub angular position at rth control step which is measured on-line, the error and its change are obtained as follows:

$$e(r)=\theta_d(r)-\theta(r) \tag{10}$$

$$\Delta e(r)=e(r)-e(r-1) \tag{11}$$

Where:

e and Δe = The error and change in the error, respectively

θ_d = The desired joint angle of the robot

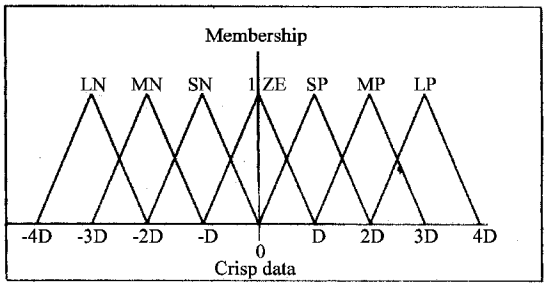

Fig. 2: The fuzzy membership functions for crisp inputs

Figure 2 demonstrates the membership functions of the crisp inputs including the error and the change in the error of the hub angle in fuzzification part as well as the controlled torque in defuzzification part as output. For every input/output seven fuzzy membership functions are considered as follows. LN, MN, SN, ZE, SP, MP and LP one-by-one correspond to large negative, medium negative, small negative, zero, small positive, medium positive and large positive, respectively.

Each part is a function of the productive parameter D which is updated on-line through a controlling process. The productive parameters for the error, change in the error and the control torque for the ith arm are D^{e}_{i}, $D^{\Delta e}_{i}$ and D^{u}_{i}, respectively. The fuzzy rules are determined based on the environmental and the performance conditions of the robot. The proposed fuzzy rules as a look-up table are presented in Table 1.

Due to the symmetry of the membership functions, the weighted average method is used for defuzzification (Ross, 2004).

Considering m = 2 measured input state variables, the output is obtained as follows:

$$u = \frac{\sum_{h=1}^{g} \bar{u}_h \prod_{j=1}^{m} \mu_j(x_j)}{\sum_{h=1}^{g} \prod_{j=1}^{m} \mu_j(x_j)} \tag{12}$$

Where:

u = The crisp torque signal

$\bar{u}_h$ = The centroid of the area for every triangular output membership function

g = The number of inputs

And x_j is j-type input error and μ_j is the input signal membership function x_j in jth linguistic set.

Neural network: The task of neural networks is to produce the D parameter for any membership function in the fuzzification and defuzzification parts. Figure 3 schematically shows the fuzzy controller with neural tune-up.

Table 1: The fuzzy controller role base

e\Δe	LN	MN	SN	ZE	SP	MP	LP
LN	LN	LN	LN	LN	ZE	MP	SP
MN	MN	MN	MN	MN	ZE	MP	MP
SN	LN	MN	SN	SN	ZE	SP	SP
ZE	SN	ZE	ZE	ZE	ZE	ZE	SP
SP	SN	SN	ZE	SP	SP	MP	LP
MP	MN	MN	ZE	MP	MP	MP	MP
LP	SN	SN	ZE	LP	LP	LP	LP

The tune-up neural networks of productive D^{e}_{i}, $D^{\Delta e}_{i}$ and that of D^{u}_{i} are denoted by NNe, NNΔe and NNu, respectively. Every network consists of three layers and the weight of the neurons for every layer is updated by optimizing the cost function. The cost function is defined as follows:

$$J^{k}_{i} = \frac{1}{2}\left[R^{k}_{i}\, e^{2}_{i}(t) + P^{k}_{i}\, \dot{e}^{2}_{i}(t) + Q^{k}_{i}\, \psi^{2}(L_3, t)\right] \tag{13}$$

Where:

$$R^{k}_{i} = \begin{bmatrix} R^{e}_{i} \\ R^{\Delta e}_{i} \\ R^{u}_{i} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ r^{u}_{i} \end{bmatrix}; \; P^{k}_{i} = \begin{bmatrix} P^{e}_{i} \\ P^{\Delta e}_{i} \\ P^{u}_{i} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ P^{u}_{i} \end{bmatrix}; \quad Q^{k}_{i} = \begin{bmatrix} Q^{e}_{i} \\ Q^{\Delta e}_{i} \\ Q^{u}_{i} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ q^{u}_{i} \end{bmatrix} \tag{14}$$

The cost function and arm number are denoted by j and i, respectively. And k determine the network type, NNe, NNΔe and NNu. The weight error vector is Q^{k}_{i} and the error derivative vector and the deformation vector of networks for robot arms are denoted by R^{k}_{i} and P^{k}_{i}, respectively. The error, the error derivative and the vibration are directly controlled by these weight vectors. The weights are modified to the lowest point of the cost function in the weights space, consequently:

$$\nabla w = -\eta \left(\frac{\partial J}{\partial w}\right)^{T} \tag{15}$$

Where:

∇w = The gradient of weight (or biases)

η = The learning rate which is positive

Now, we calculate the partial derivative of the cost function with respect to the network weights as follows:

$$\frac{\partial J^{k}_{i}}{\partial w^{k}_{i}} = \frac{\partial J^{k}_{i}}{\partial e_{i}} \frac{\partial e_{i}}{\partial w^{k}_{i}} + \frac{\partial J^{k}_{i}}{\partial \dot{e}_{i}} \frac{\partial \dot{e}_{i}}{\partial w^{k}_{i}} + \frac{\partial J^{k}_{i}}{\partial \psi} \frac{\partial \psi}{\partial w^{k}_{i}} \tag{16}$$

where, w^{k}_{i} are the weights of the network layers. After substituting Eq. 13 into Eq. 16, we have:

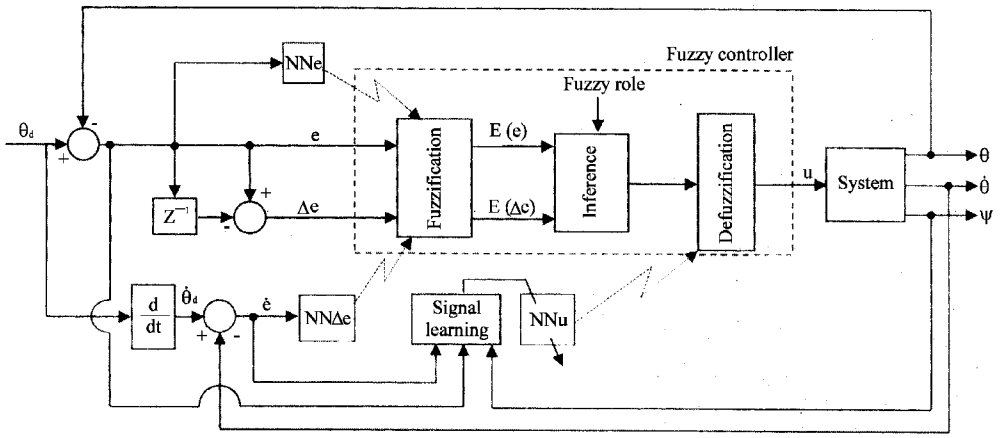

Fig. 3: Schematic plan of fuzzy controller with neural tuner

$$\frac{\partial J_i^k}{\partial w_i^k} = R_i^k e_i \frac{\partial e_i}{\partial w_i^k} + P_i^k \dot{e}_i \frac{\partial \dot{e}_i}{\partial w_i^k} + Q_i^k \psi \frac{\partial \psi}{\partial w_i^k} \quad (17)$$

Using the chain role, Eq. 17 can be rewritten as follows:

$$\frac{(\partial J_i^k)}{(\partial w_i^k)} = -R_i^k e_i \frac{(\partial \theta_i)}{(\partial u_i)} \frac{(\partial u_i)}{(\partial D_i^k)} \frac{(\partial D_i^k)}{(\partial w_i^k)} - P_i^k e_i^* \frac{(\partial \theta_i^*)}{(\partial u_i)} \frac{(\partial u_i)}{(\partial D_i^k)} \frac{(\partial D_i^k)}{(\partial w_i^k)} + Q_i^k \psi \partial \psi / (\partial u_i) \quad (18)$$

For network learning, Lightbody *et al.* (1990) proposed using the sign of terms

$$\frac{\partial \theta_i}{\partial u_i}, \frac{\partial \dot{\theta}_i}{\partial u_i} \text{ and } \frac{\partial \psi}{\partial u_i}$$

instead of their real values. Therefore, after modifying Eq. 18 and substituting it into Eq. 15, we have:

$$\Delta w_i^k = \eta \left(R_i^k e_i + P_i^k \dot{e}_i + Q_i^k \psi \right) \frac{\partial u_i}{\partial D_i^k} \frac{\partial D_i^k}{\partial w_i^k} \quad (19)$$

The sigmoid function is selected as the activating function:

$$\Phi(I) = \frac{2}{1 + e^{-aI}} - 1 \quad (20)$$

The first and second input layers are as follows:

$$\overset{2}{p}I_i^k = \overset{2}{p}b_i^k + x_i^k \overset{2}{p}w_i^k \ (p = 0,1,...,n) \quad (21)$$

$$\overset{3}{1}I_i^k = \overset{3}{p}b_i^k + \sum_{p=1}^{n} \phi \left(\overset{2}{p}I_i^k \right) \overset{3}{p}w_i^k \quad (22)$$

Where:

$\overset{2}{p}I_i^k$ = The pth input neuron in the second layer

$\overset{3}{1}I_i^k$ = The input for the only neuron (output neuron) in the third layer

x_i^k = The k-type network input at ith joint (arm)

And:

$$\overset{2}{p}w_i^k, \overset{2}{p}b_i^k, \overset{3}{p}w_i^k \text{ and } \overset{3}{p}b_i^k$$

are the weight and bias of the second and the third layers, respectively. Meanwhile, x_i^{NNe} and x_i^{NNu} which are the inputs for NNe and NNu networks are assigned as the angular position for the ith arm and similarly $x_i^{NN\Delta e}$ which is an input for the NNΔe network is assigned as the angular velocity for the ith arm. The output of the network external layer (third layer) is assumed to be equal to the productive parameter of the membership function, therefore:

$$D_i^k = \Phi \left(\overset{3}{1}I_i^k \right) \quad (23)$$

Figure 4 demonstrates three-layer perceptron neural network. Using Eq. 23-20 the following equations are derived:

$$\frac{\partial D_i^k}{\partial \left(\overset{3}{p}b_i^k \right)} = \frac{1}{2} \alpha \left\{ 1 - \left[\Phi \left(\overset{2}{p}I_i^k \right) \right]^2 \right\} \quad (24)$$

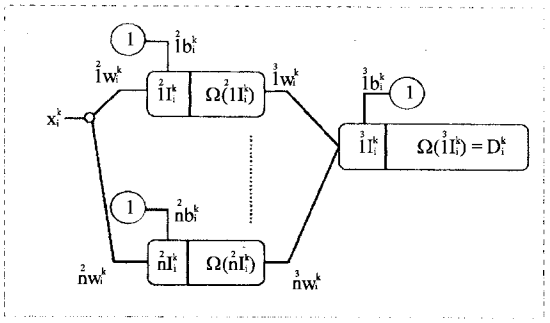

Fig. 4: Three-layer perceptron neural network

$$\frac{\partial D_i^k}{\partial\left(\overset{2}{p}b_i^k\right)}=\frac{1}{4}\alpha^2\left(\overset{3}{p}w_i^k\right)\left\{1-\left[\Phi\left(\overset{2}{p}I_i^k\right)\right]^2\right\}\left\{1-\left[\Phi\left(\overset{3}{1}I_i^k\right)\right]^2\right\} \tag{25}$$

$$\frac{\partial D_i^k}{\partial\left(\overset{3}{p}w_i^k\right)}=\frac{1}{2}\alpha\Phi\left(\overset{2}{p}I_i^k\right)\left\{1-\left[\Phi\left(\overset{2}{p}I_i^k\right)\right]^2\right\} \tag{26}$$

$$\frac{\partial D_i^k}{\partial\left(\overset{2}{p}w_i^k\right)}=\frac{1}{4}\alpha^2 x_i^k\left(\overset{3}{p}w_i^k\right)\left\{1-\left[\Phi\left(\overset{2}{p}I_i^k\right)\right]^2\right\}\left\{1-\left[\Phi\left(\overset{3}{1}I_i^k\right)\right]^2\right\} \tag{27}$$

Substituting Eq. 24-27 into Eq. 17 and using momentum to increase the convergence rate, the following equations are derived:

$$\Delta\left(\overset{2}{p}b_i^k\right)=\frac{1}{2}\alpha\eta\left(R_i^k e_i+P_i^k\dot{e}_i+Q_i^k\psi\right)\times\left\{1-\left[\Phi\left(\overset{2}{p}I_i^k\right)\right]^2\right\}\frac{\partial u_i}{\partial D_i^k}+\beta_i^k\Delta\left(\overset{3}{p}b_i^k\right)_{r-1} \tag{28}$$

$$\Delta\left(\overset{2}{p}b_i^k\right)=\frac{1}{4}\alpha^2\eta\left(R_i^k e_i+P_i^k\dot{e}_i+Q_i^k\psi\right)\left(\overset{3}{p}w_i^k\right)\times\left\{1-\left[\Phi\left(\overset{2}{p}I_i^k\right)\right]^2\right\}\left\{1-\left[\Phi\left(\overset{3}{1}I_i^k\right)\right]^2\right\}\frac{\partial u_i}{\partial D_i^k}+\beta_i^k\Delta\left(\overset{2}{p}w_i^k\right)_{r-1} \tag{29}$$

$$\Delta\left(\overset{3}{p}w_i^k\right)=\frac{1}{2}\alpha\eta\left(R_i^k e_i+P_i^k\dot{e}_i+Q_i^k\psi\right)\Phi\left(\overset{2}{p}I_i^k\right)\left\{1-\left[\Phi\left(\overset{2}{p}I_i^k\right)\right]^2\right\}\frac{\partial u_i}{\partial D_i^k}+\beta_i^k\Delta\left(\overset{3}{p}w_i^k\right)_{r-1} \tag{30}$$

$$\Delta\left(\left(p^2\right)w_i^k\right)=\frac{1}{4}\alpha^2\eta\left(R_i^k e_i+P_i^k e^{*i}+Q_i^k\psi\right)x_i^k\left(\left(p^3\right)w_i^k\right)\left\{1-\left[\Phi\left(?\left(\downarrow p^{\uparrow}2\right)Iw_i^k\right)\right]^2\right\}\left\{1-\left[\Phi\left(\left(p^3\right)Iw_i^k\right)\right]^2\right\}\frac{\left(\partial u_i\right)}{\left(\partial D_i^k\right)}+\beta_i^k\Delta\left(\left(p^2\right)w_i^k\right) \tag{31}$$

where, β and r are the momentum coefficient and the control stage, respectively.

RESULTS AND DISCUSSION

In this study, the performance and the efficiency of designed controller is studied by performing simulation and the results are discussed.

As an example, we consider a 3-link robot with a third flexible arm. The geometrical and mass properties are listed in Table 2. We assume that all the arms have rectangular cross section. The arms are made of aluminum with the density and Young's modulus of 2710 kg m^{-3} and 71 GPa, respectively. The mass, inertial moment and the gear ratio of the rotor are 0.1 kg, 0.025 kg m^{-2} and 1, respectively. The payload mass is 300 g.

Learning rate and momentum coefficient in the neural network are considered 0.35 and 0.9, respectively. Using Eq. 3 and 16, the first frequency of vibration of the flexible arm and two first natural frequencies would be 11.96 and 293.16, respectively.

The path of end effector for the 3-link planar robot with a payload is illustrated in Figure 5. The initial relative angles of the arms are zero before the motion starts and the selected path is a straight line that makes 60 degrees with the arms. The position of the robot arms and that of the tip payload mass are schematically shown during the motion.

Figure 6 demonstrates the vibration of the robot tip in the trajectory tracking by proposed controller. The maximum deflection of the robot tip is 3.7 mm.

Figure 7-9 present the control driving torques at the first, second and the third joints, respectively. The control torques that are produced by the controller for positioning the arms at the desired angles for trajectory tracking vary between 0 and 2 N cm^{-1}.

Figure 10 shows the angular position error of the joints 1, 2 and 3, respectively. The one-second settling time for all three arm controllers implies that the learning speed is suitable for the convergence of neural network r the tune-up of D parameter.

Table 2: The mass and dimension of the robot arms

		Arm		
Quantity	Units	First	Second	Third
Length	cm	20	20	40
Width	mm	10	10	20
Thickness	mm	15	15	2
Mass		81	81	43

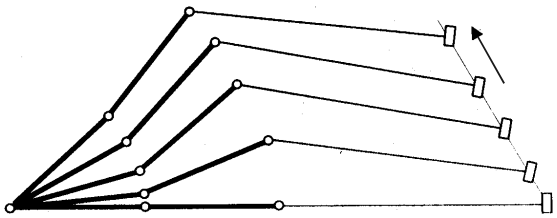

Fig. 5: Tip trajectory and robot arms position

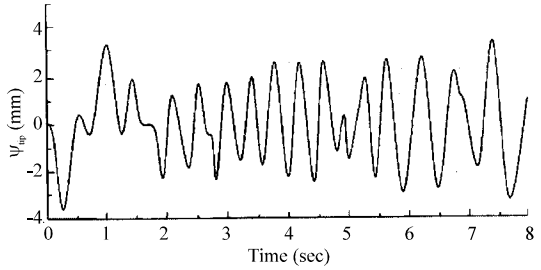

Fig. 6: Vibration of robot tip

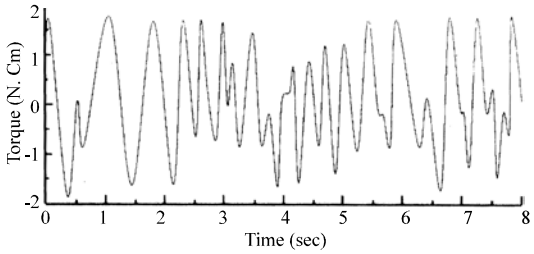

Fig. 7: Torque of the first motor's arm

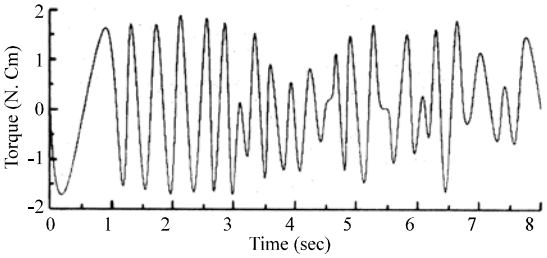

Fig. 8: Torque of the second motor's arm

Figure 11-14 indicate productive parameter D for the error, change in the error and the control torque of joints 1, 2 and 3, respectively. These parameters vary as step function in the joint 1, 2 approximately, whereas they

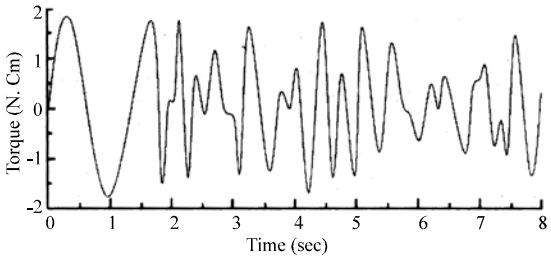

Fig. 9: Torque of the third motor's arm

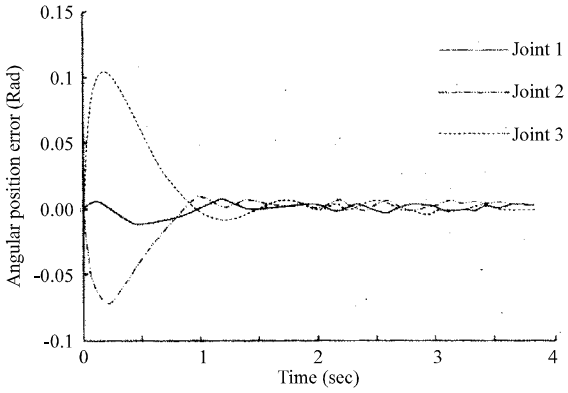

Fig. 10: Angular position error of joints 1, 2 and 3

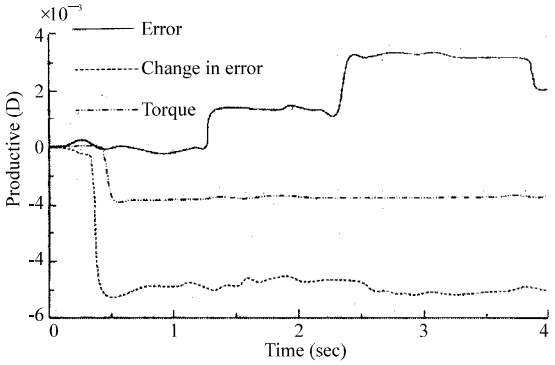

Fig. 11: Productive "D" related to membership functions of error, change in error and torque in the joint 1

become steady after reaching a peak close to the start point of the joint 3. We would like to compare the results of the proposed approach with those of an inverse control method which is a common model-based technique.

Figure 14 shows the tip tracking error for the three controllers, inversion and fuzzy controller and fuzzy controller with neural tuner. The fuzzy controller results in rather high settling time, overshoot and undershoot in the system response whereas these parameters are significantly decreased using the proposed controller.

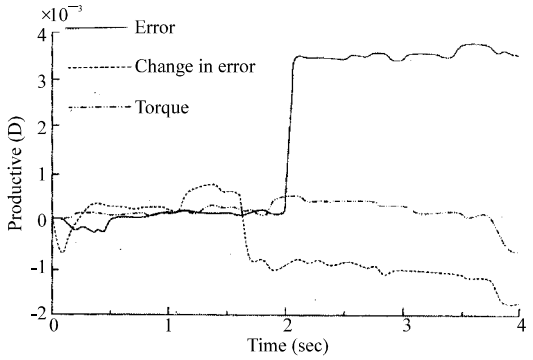

Fig. 12: Productive "D" related to membership functions of error, change in error and torque in the joint 2

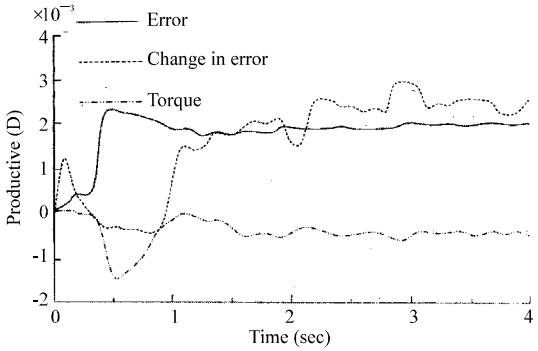

Fig. 13: Productive "D" related to membership functions of error, change in error and torque in the joint 3

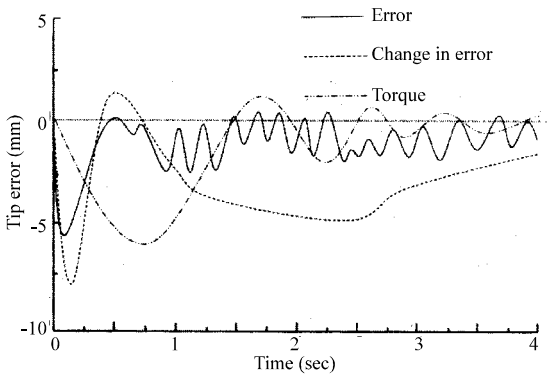

Fig. 14: Tip error for fuzzy, fuzzy with neural tuner and inversion controllers

CONCLUSION

In this study, a new fuzzy logic control with neural tune-up for a serial flexible robot was presented. The equations of motion were derived using Lagrangian mechanics and the assumed-modes method. A fuzzy controller consisting of parameter membership functions in fuzzification and in defuzzification parts was presented; and the crisp output was calculated using the weighted average method. Using an on-line learning three-layer perceptron neural network, the productive parameter D of the membership functions was produced while tracking the trajectory. A trajectory tracking simulation was carried out and the joints angular error and the tip error proved the suitability of design and affirmed the effectiveness of the fuzzy controller with neural tuner. The analyses and comparison of the results from the proposed controller with those from the two other commonly used controllers, showed the superiority of the proposed controller.

REFERENCES

Abdullahi, A.M., Z. Mohamed, M. Muhammad and A.A. Bature, 2013. Vibration control comparison of a single link flexible manipulator between fuzzy logic control and pole placement control. Intl. J. Scent. Technol. Res., Vol. 2, No. 12.

Alam, M.S. and M.O. Tokhi, 2008. Hybrid fuzzy logic control with genetic optimization for a single-link flexible manipulator. Eng. Application of Artificial Intelligence, 21: 858-873.

Caracciolo, R., D. Richiedei, A. Trevisani and V. Zanotto, 2005. Robust mixed-norm position and vibration control of flexible link mechanisms. Mechatronics, 15 (7): 767-791.

De Luca, A., P. Lucibello and G. Ulivi, 1989. Inversion techniques for trajectory control of flexible robot arms. Robotic Syst., 6 (4): 325-344.

Huang, S.J. and J.S. Lee, 2000. A stable self-organizing fuzzy controller for robotic motion control. IEEE Transactions, 47 (2): 421-428.

Karami, N., M.H. Korayem, A.M. Shafei and S.R. Nekoo, 2014. Theoritical and experimental investigation of dynamic load carrying capacity of flexible-link manipulator in pointto-point motion. Modares Mechanical Eng., 14 (15): 199-206.

Korayem, M.H., A.M. Shafei, M. Doosthoseini and B. Kadkhodaei, 2014. Dynamic modeling of visco-elestic robotic manipulators using Timoshenko beam theory. Modares Mechanical Eng., 14 (1): 131-139.

Lee, T.H., S.S. Ge and Z.P. Wang, 2001. Adaptive robust controller design for multi-link flexible robots. Mechatronics, 11 (8): 951-967.

Lightbody, G., W.H. Wu and G.W. Irwin, 1990. Control Application for feedforward Networks, Neural Networks for Control, MIT Press, Cambridge, MA., pp: 51-71.

Lochan, K. and B.K. Roy, 2014. Control of two-link 2-DOF Manipulator Using Fuzzy Logic Techniques: A Review. Proceedings of 4th International Conference on Soft Computing for Problem Solving, 1: 499-511.

Mallikarjunaiah, S. and S.N. Reddy, 2013. Adaptive neuro-fuzzy interface system controller for flexible link manipulator. ACTA Electrotechnica, Vol. 54, No. 2.

Matia, F., G.N. Marichal and E. Jimenez, 2014. Fuzzy Modeling and Control: Theory and Applications, Springer.

Meirovitch, L., 1967. Analytical methods in vibrations. Macmillan, New York, USA.

Meza, J.L., V. Santibanez, R. Soto and M.A. Llama, 2012. Fuzzy self-tuning PID semiglobal regulator for robot manipulators. IEEE Trans. Ind. Electron., 59 (6): 2709-2717.

Mirshekaran, M., F. Piltan, Z. Esmaeili, T. Khajeaian and M. Kazeminasab, 2013. Design sliding mode modified fuzzy linear controller with application to flexible robot manipulator. Intl. J. Modern Edu. Computer Sci., 5 (10): 53-63.

Qiu, Z.C., J.D. Han and J.G. Liu, 2015. Experiments on fuzzy sliding mode variable structure control for vibration suppression of a rotating flexible beam. J. Vibration and Control, 21 (2): 343-358.

Rao, S.S., 2007. Vibration of Continuous Systems. 4th Edn, Chapter 11, New Jersey: Wiley and Sons.

Ross, T.J., 2004. Fuzzy Logic with Engineering Application. 2nd Edn., Wiley and Sons, pp: 101.

Rumelhart, D.E. and J.L. McClelland, 1986. Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1 MIT Press, Cambridge, MA.

Sahamijoo, G., O. Avatefipour, M.R.S. Nasrabad, M. Taghavi and F. Piltan, 2015. Research on minimum intelligent unit for flexible robot. International journal of advanced science and technology. Intl. J. Adv. Sci. Technol., 80: 79-104.

Tian, L. and C. Collins, 2005. Adaptive neuro-fuzzy control of a flexible manipulator. Mechatronics, 15: 1305-1320.

Wang, L. and C.K. Li, 1997. Fuzzy logic adaptive control of constrained flexible-link robot. Systems Sci. Syst. Eng., 6 (3): 348-357.

Wu, J. and Y.G. Tzou, 1993. Comparison of fuzzy logic and self-tuning adaptive control of single-link flexible arm. Mechatronics, 3 (4): 451-464.

Yue, S., D. Henrich, W.L. Xu and S.K. Tso, 2002. Point-to-point trajectory planning of flexible redundant robot manipulators using genetic algorithms. Robotica, 20: 269-280.

Zhang, N., Y. Feng and X. Yu, 2004. Optimization of terminal sliding control for two-link flexible manipulators. The 30th IEEE Annual Conference of Industrial Electronics Society, 2: 1318-1322.