Journal of Applied Soft Computing, Volume 9 , Issue 1 (January 2009),Pages 149-158
http://doi.dx.org/10.1016/j.asoc.2008.02.004

Applied Soft Computing 9(2009) 149-158

Soft computing methods applied to the control of a flexible robot manipulator

B. Subudhi[a] and A. S. Morris[b]

[a] Dept. of Electrical Engineering, National Institute of Technology, Rourkela-769008, India

[b] Dept. of Automatic Control & Systems Engineering, Mappin Street, Sheffield S1 3JD, University of Sheffield, United Kingdom

Abstract: The paper describes use of soft computing methods (fuzzy logic and neural network techniques) in the development of a hybrid fuzzy neural control (HFNC) scheme for a multi-link flexible manipulator. A manipulator with multiple flexible links is a multivariable system of considerable complexity due to the inter-link coupling effects that are present in both rigid and flexible motions. Modelling and controlling the dynamics of such manipulators is therefore difficult. The proposed HFNC scheme generates control actions combining contributions form both a fuzzy controller and a neural controller. The primary loop of the proposed HFNC contains a fuzzy controller and a neural network controller in the secondary loop to compensate for the coupling effects due to the rigid and flexible motion along with the inter-link coupling. It has been ascertained from the present investigation that the proposed soft-computing-based controller works effectively in the tracking control of such a multi-link flexible manipulator. The results are extendable to other multivariable systems of similar complexity.

Keywords: soft computing, fuzzy logic, radial basis function neural network, flexible manipulator

1. Introduction

The conventional approach to design of controllers for any plant, process or system requires knowledge of an accurate mathematical model of the system to be controlled, which is often difficult to derive analytically. In consequence, it is difficult or impossible to design controllers for complex systems such as nonlinear multivariable systems using these conventional approaches that require a plant model. Hence, control engineers frequently turn their attention to soft computing techniques such as fuzzy logic and neural networks (NNs), which can be exploited to achieve control objectives without the availability of accurate mathematical plant models being essential. The control of a number of complex systems such as robotic systems, electric drives, power system, communication channel have been achieved using fuzzy logic and or neural network control strategies without a-priori knowledge of the dynamics [1].

Robot control problems are highly nonlinear, heavily coupled and time-varying systems. Hence, an accurate mathematical model for such system is difficult to obtain, thus making it difficult to control using conventional techniques. More control difficulties such as under-actuation, distributed parameter nature of the system, non-minimum phase behaviour and inter-link flexible and rigid mode coupling effects compensation are encountered while controlling flexible robots [2-4]. A number of research investigations exploit soft computing approaches such as fuzzy logic and neural network techniques in designing improved controllers for flexible link manipulators [2-11]. Controlling the tip position of a single-link flexible manipulator has been achieved successfully by employing neural-network and fuzzy controllers [3, 4]. In [2], control of a flexible manipulator using a neuro-fuzzy control method has been described, where the weighting factor of the fuzzy logic controller (FLC) is adjusted by the dynamic recurrent identification network (RLN) and the controller works without any knowledge about the manipulator system. An intelligent optimal control for a nonlinear flexible robot arm driven by a permanent-magnet synchronous servo motor has been designed using a fuzzy neural network control approach [3]. The reported intelligent optimal control system consists of an optimal controller which minimizes a quadratic performance index and a fuzzy neural network controller that learns an uncertain nonlinear flexible manipulator dynamics together with an adaptive bound estimation algorithm to provide robust control for compensating the approximation error of the RLN. The proposed intelligent controller has been implemented in both simulated and real-time flexible arm experimental set-up for tip-position tracking and the controller exhibits good tracking accuracy. In [4], a fuzzy controller has been developed for a three-link robot with two rigid links and one flexible fore-arm. This controller design is based on fuzzy Lyapunov synthesis where a Lyapunov candidate function has been chosen to derive the fuzzy rules. In [5], four neural network based control schemes have been proposed for tip-position tracking of highly flexible link manipulator. These four neural controllers are as follows. Two of the NN control schemes used feedback-error-learning scheme to learn the inverse dynamics corresponding to the redefined output of the tip position, the third NN controller uses the steepest decent technique for training with an objective function that includes the tip deflection term and the fourth neural network structure consisted of two neural networks, where one network is trained as an online feedback controller, and the other is trained to determine an appropriate output for feedback (in the sense of ensuring minimum phase behavior of the system). Lin and Lewis [6] have exploited fuzzy logic and the singular perturbation approach for flexible-link robot arm control. By using the singular perturbation technique they obtain two-time scaled slow and fast subsystems of the flexible robot arm. Having obtained slow and fast subsystems they applied fuzzy controllers to these subsystems to derive a composite control for simultaneous damping of link deflection and trajectory tracking. Subudhi and Morris [7] have proposed a neuro-fuzzy scheme for tip position control of a single link flexible robot manipulator. In their proposed neuro-fuzzy controller, the scale factors of the fuzzy controller are adapted on-line using a neural network which is trained with an improved back-propagation training algorithm.

Two different neuro-fuzzy feedforward controllers namely a Takagi-Sugeno fuzzy model and a rectangular local model have been proposed in [12] to compensate the nonlinearities in a flexible manipulator. After having modeled the distributed link flexure by the popular assumed modes method, Jemil M. Remo et al. [13] have employed an inverse dynamics based fuzzy controller to obtain tracking and deflection control of single-link flexible manipulator. The controller in [13] generated the control action (torque) by summing up the contributions of two different FLCs i.e. one for the joint angle and the other for tip position. Recurrent neural networks have been employed in [14] to approximate the dynamics of a constrained flexible manipulator with uncertainty for hybrid position and force control.

Also, fuzzy logic and neural network techniques have been used in [15] to develop an adaptive self-organizing neuro-fuzzy controller for tip position tracking control of a single link flexible manipulator where fuzzy rules are generated during the control process using on-line neural network algorithm. The adaptive neuro-fuzzy controller proposed in [16] has been attributed several advantageous features namely self-organizing fuzzy rules generation, fast on-line learning, fast convergence and self-adaptive in presence of disturbances.

A fuzzy vector approach has been applied [17] to search the direction of the constraint surface of an unknown object so that an end-effector of a robot manipulator could efficiently follow the contour of an object. Further, a TSK type fuzzy-neural network control scheme has been developed in [18] for joint position control of n-link robot manipulator in which a five-layer fuzzy neural network has been trained using an adaptive tuning laws based on Lyapunov stability theorem. The neuro–fuzzy control for a robotic manipulator proposed in [19] has used the fuzzy dynamic model of the robot manipulator derived from input-output data from the robot control process and then the fuzzy model parameters are modified on-line thus making it a truly adaptive controller. A distributed in the sense of importance degrees of output variables of the system has been proposed for a flexible link manipulator in [20]. In this distributed fuzzy controller the two velocity variables which have higher importance have been grouped together as the inputs to the velocity FLC while the two displacement variables which have lower importance degrees have been used as the inputs of the displacement FLC. The resultant fuzzy control action has been obtained by summing up the contributions from these two FLCs.

After successfully dividing the complex dynamics of a flexible link robot by using the singular perturbation approach into a two-time scale slow subsystem and a fast subsystem, a two-time scale fuzzy controller has been developed for the slow-subsystem for trajectory tracking and a linear quadratic regulator (LQR) fast subsystem controller for damping of link deflections [21]. Adaptive control for flexible link manipulator has been achieved by using a dynamical time-delay neuro-fuzzy controller [22]. The time-delay neuro-fuzzy networks have been designed with a rule representation in TSK fuzzy system structure. The neuro-fuzzy controller for co-operative control of two robots has been designed by training the neural network with model predictive control signal via back propagation [23]. This neural-network provides optimal parameters of the fuzzy logic controller. Apart from neural network, other soft computing technique such as the genetic algorithm (GA) has been exploited to improve the FLC performance for flexible link and join manipulator [24].

Although, intelligent techniques such as neural-network, fuzzy-logic and genetic algorithm have been applied successfully applied to a large number of robotic applications as described above their applications to multi-link flexible manipulators, although a good number of research investigations focused on single link flexible manipulator tracking and position control together with damping link deflections[6-15]. Further in the work of single link flexible manipulator tip position control [5-7], only a simple two-input and one output fuzzy logic controller (FLC) was applied. Such a simple fuzzy controller is suitable in the case of single-link flexible manipulator, as there is no inter-link coupling due to rigid and flexible motion between the links. However, for a manipulator with many flexible links, it is difficult to model the non-linear characteristics of the coupling effects between links that have both rigid and flexible motion dynamics. In the past, control of such a complex multivariable system has been approached by using the model-free feature of fuzzy systems, such as the work on a two-link flexible manipulator described in [11] in which fuzzy control is exploited to compensate for the interaction between the links. In order to compensate for the interaction between the loops, this work proposed a fuzzy logic control (FLC) strategy, where the inputs to the fuzzy logic controller for link 2 comprised of its own joint position and tip acceleration information along with the tip acceleration coming from the first link. It was found that this coupled FLC performed better than an uncoupled FLC in which no information from the first link is used in the control of the second link. However, because the coupled FLC increased the number of inputs used for the second link fuzzy control loop, the effective size of the rule base was increased. This is because the size of the rule base is related to the number inputs as F^N , were F and N denote the number of fuzzy sets used for each input and the number of input variables respectively.

Thus the motivation of the present work was to address the limitations difficulties of fuzzy controller described in [11]. In recent years, a combined framework of fuzzy logic and neural network systems has been found to be suitable for controlling complex multi-input and multi-output (MIMO) systems [1, 25-28]. Motivated by various successful applications of combined fuzzy logic and neural network schemes to control other complex multivariable systems, this paper describes the supplementation of a fuzzy logic system with a neural network in order to achieve effective tip regulation performance in a multi-link flexible manipulator. In the hybrid fuzzy neural control (HFNC) scheme proposed, the primary loop consists of a fuzzy logic controller. A radial basis function neural network (RBFNN) is employed in the secondary loop to compensate for the coupling effects due to the rigid and flexible motion along with the inter-link coupling effects. The weights between the hidden and the output layers of the RBFNN used in this control scheme are adjusted on-line by a normalized least mean square (NLMS) technique [29]. To resolve the problem of storing a huge multi-dimensional rule matrix as in [11], an alternative way of implementing the FLC is adopted in this paper. The performances of the proposed HFNC scheme are compared with those of a multivariable fuzzy controller and a Lyapunov-based PD controller [30] that uses joint position, velocity and tip acceleration feedback signals.

The main contributions of the paper are as follows:

i. The development of a new hybrid fuzzy neural controller for a very complex multivariable system i.e. a two-link flexible manipulator which can be easily extended to control a multi-link flexible manipulator and other multivariable systems of similar complexity.
ii. Unlike most of the research works on addressing the control difficulties for a single-link flexible manipulator, the present work developed a hybrid fuzzy neural controller for a more complicated dynamical system i.e. a multi-link flexible manipulator and the controller performances have been verified for a two-link flexible manipulator case

study.

iii. A new data handling procedure for handling multi-dimensional array of fuzzy rules has been developed. This implementation has the ability to handle rule matrices of arbitrary dimension.

The rest of the paper is organized as follows. A review of soft computing techniques has been included in Section 2. A Lyapunov stability based control for a multi-link flexible manipulator has been presented Section 3. Section 4 describes about the design of multivariable fuzzy controller for tip position control of a multi-link flexible manipulator. A new hybrid fuzzy neural control scheme has been designed in section 5. Implementation of the above controllers, results and discussions are provided in Section 6. Section 7 concludes the work.

2 Review of soft computing techniques

Soft computing, a collection of fuzzy logic technique, neural-networks, evolutionary computation techniques has proven to be a powerful tool for adding autonomy to many complex systems [31]. These soft computing techniques excel over classical control methods in many aspects, such as algorithm simplicity and tolerance for imprecision. Fuzzy control deals effectively with a noisy and imprecise environment. The knowledge from a human operator is embedded into rules of the fuzzy controllers, which efficiently increases their robustness against noise and parameter variations. An excellent recent survey on analysis and design methods of model based fuzzy control systems has been provided in [32]. It is believed that the different fuzzy methods such as Takagi-Sugeno provide a systematic approach to analysis and design of model based fuzzy control systems and in turn analysis of the fuzzy controller stability. Combined use of fuzzy logic, neural networks, as well as other control algorithms has been recognized as promising approaches to develop intelligent control of many complex uncertain dynamical systems [6–8]. Neural network provides learning ability using the nonlinear optimization algorithm such as back propagation training method and many other fast learning algorithms such as Levenberg Marquardt technique. It has also been demonstrated that neural networks are well suited to the control of complex dynamical systems [1]. The principle of neuro-control is based on a learning and function approximation capability. The learning characteristics of neuro-controllers in a changing operating environment distinguish them from classical controllers with invariant structures and parameters and thus enable them to provide better results. Good overviews of recent research work on neuro-control are given in [13, 15] and the use of genetic algorithms for optimising the parameters of diverse kinds of controllers is reviewed in [1]. NNs and GAs are also combined with fuzzy-logic-based schemes to enhance adaptation and learning ability [1]. The combination of fuzzy logic and neural networks to form more intelligent control systems is a particularly popular research topic [1, 28, 29]. Soft computing approaches have been successfully exploited to design intelligent controllers for a number of robotic applications [32].

3 Proportional Derivative and Acceleration Feedback Controller

Fig.1 shows the structure of the Proportional Derivative and Acceleration Feedback Controller (PDAC) for tip position regulation of a multi-link flexible manipulator.

The control structure for the ith link loop is obtained by utilizing three variables of the manipulator which are the joint angle θ_i, its rate $\dot{\theta}_i$ and the tip acceleration, a_{ti}. The closed-loop multi-link flexible manipulator system is stable with application of n control torque signals given by

$$u_i(t) = -k_{pi}[\theta_i(t) - \theta_{di}] - k_v\dot{\theta}_i - k_{ai}a_{ti}(t)\int_0^t \dot{\theta}_i(\tau)a_{ti}(\tau)d\tau \quad i = 1,2,.....,n \tag{1}$$

where τ is a dummy time variable, a_{ti} is the tip acceleration signal of the ith link, θ_{di} and θ_i are respectively the desired and actual joint positions of the ith joint, u_i is the torque generated by the ith actuator, k_{pi} and k_{vi} are the proportional and derivative control gains and k_{ai} is the controller gain related to tip acceleration. It is assumed that k_{pi}, k_{vi} and $a_{ti} > 0$.

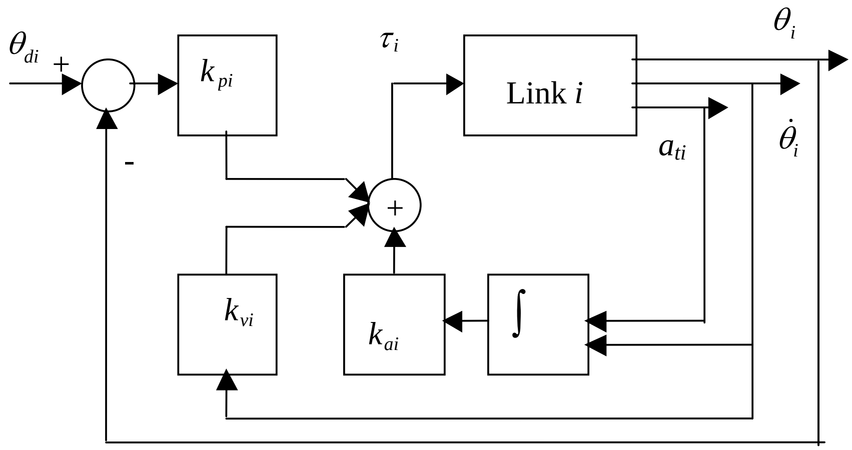

Fig.1 Structure of the Proportional Derivative and Acceleration Controller

Let L_v be a Lyapunov function as chosen in [11] given by

$$L_v = E_k + E_p + 0.5k_{pi}[\theta_i(t) - \theta_{di}]^2 + 0.5k_{ai}a_{ti}(t)[\int_0^t \dot{\theta}_i(\tau)a_{ti}(\tau)d\tau]^2 \tag{2}$$

where E_k, E_p are the total kinetic and potential energies of the manipulator. Assuming operation of the robot without gravitational influence, the total energy of the multi-link flexible manipulator system must be equal to the work done by the *n* actuating control signals, i.e.

$$E_k - E_{k0} + E_p - E_{p0} = \sum_{i=1}^{n} \int_0^t u_i \dot{\theta}_i(\tau)d\tau \tag{3}$$

where E_{k0} and E_{p0} are the kinetic and potential energy of the system at the initial time. Differentiating equation (3) with respect to time gives:

$$\dot{E}_k + \dot{E}_p = \sum_{i=1}^{n} u_i \dot{\theta}_i(t)dt \tag{4}$$

The time derivative of equation (1) after substituting equation (4) can be written as

$$\dot{L}_v = \sum_{i=1}^{n} u_i \dot{\theta}_i(t) + k_{pi}\dot{\theta}_i(t)[\theta_i - \theta_{di}] + k_{ai}\dot{\theta}_i\, a_{ti}(t)[\int_0^t \dot{\theta}_i(\tau)a_{ti}(\tau)d\tau] \tag{5}$$

Now substituting equation (1) into the above equation gives

$$\dot{L}_v = -\sum_{i=1}^{n} k_{vi}\dot{\theta}_i^2 \tag{6}$$

From equation (6), it is clear that $\dot{L}_v$ is negative semi-definite. Therefore, application of the control torques given in equation (1) guarantees the asymptotic stability of the closed-loop multi-link flexible manipulator system.

4. Multivariable fuzzy logic controller (FLC)

The design of a multivariable fuzzy logic controller for a multi-link flexible manipulator with rigid joints is discussed in this section. Unlike the fuzzy logic controller described in [8,9], a greater number of input and output variables are used because each output variable ($\theta_i, i = 1,2,.....,n$) in the multi-link flexible manipulator is actively influenced by more than one control input ($u_i, i = 1,2,.....,n$). Therefore, to handle this situation, a multivariable fuzzy controller is designed for this MIMO system. Nevertheless, the intended FLC also consists of the same five basic blocks as in a multi-input single output (MISO) FLC, namely, fuzzifier, rule sets of fuzzy rule description, inference engine, defuzzifier and the scaling unit. The design of this MIMO FLC can be accomplished by considering a parallel scheme involving individual MISO FLCs [33, 34].

Therefore, it now becomes simpler to use the design procedure for the two inputs and single output fuzzy logic controller described in [7] to develop a multi-input and single output fuzzy logic controller to be applied to a multi-link flexible manipulator. The same compositional rule of inference scheme [33] is used here to obtain the fuzzy inference. But in this case, the minimum of all the membership functions corresponding to three antecedent variables is taken first. Then, a maximum operation is applied. The structure of the coupled MIMO FLC for a multi-link flexible manipulator with rigid joints is shown in Fig.2.

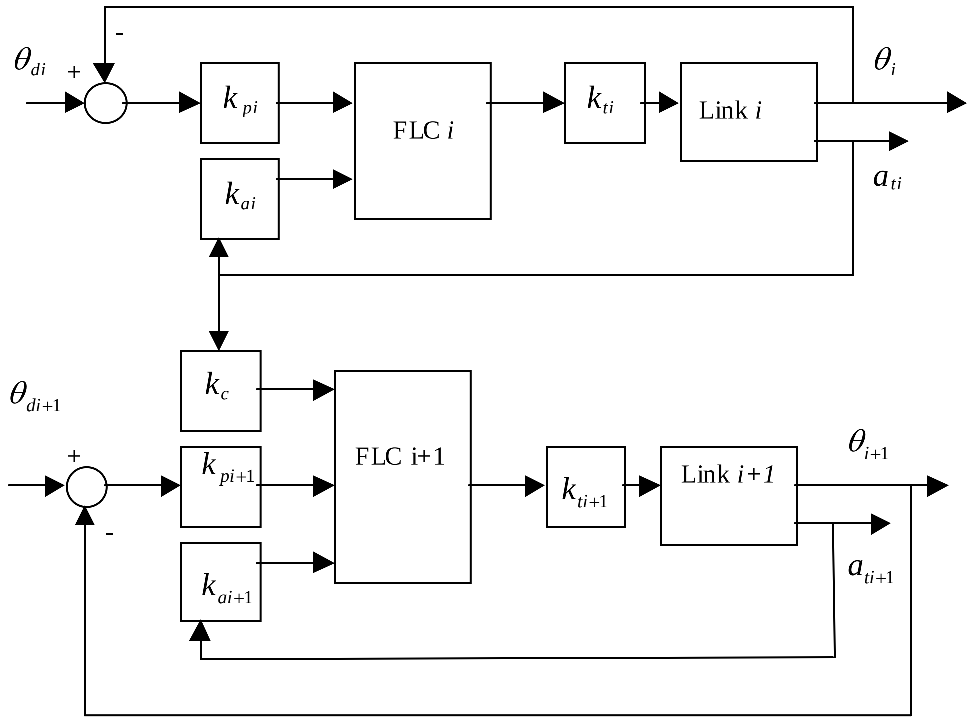

Fig.2 Structure of the Coupled MIMO FLC

In Fig.2, two consecutive links (i and $i+1$) are considered. The inputs to the FLCi consist of the joint angle error ($e_i = \theta_{di} - \theta_i$) and the tip-acceleration signal, a_{ti}. The scaling factors related respectively to tracking error, tip acceleration and the output torque signals for the ith FLC are k_{pi}, k_{ai} and k_{ti}. As the effect of the tip acceleration signal of the ith link significantly affects the tip position of the ($i+1$)th link [8], the ($i+1$)th FLC takes an additional input (i.e. the tip acceleration of the ith link) in addition to its own joint position error and tip-acceleration. A scale factor k_c is used for this additional signal normalization in FLC$i+1$. The purpose of using these scale factors is to convert the actual variables into their respective normalized universe of discourse in the range [-1.0, 1.0] with view to achieve computational simplicity.

5. Hybrid fuzzy neural control scheme (HFNC)

Whilst the back-propagation algorithm is commonly used for training multi-layer perceptron (MLP) neural networks, it can lead to problems of local minima and slow convergence and is therefore best suited to control applications where off-line training is possible [26]. It is therefore necessary to consider alternative neural networks such as cerebellar model articulation controller (CMAC) and B-Spline [29]. These have fixed mappings between the input layer and the hidden layer, and the only adjustable parameters are the connecting weights between the hidden layer and the output layers. This topology allows use of a linear optimisation approach in the parameter space rather than the non-linear optimisation approach used for MLP's, resulting in the achievement of much faster convergence to a global solution.

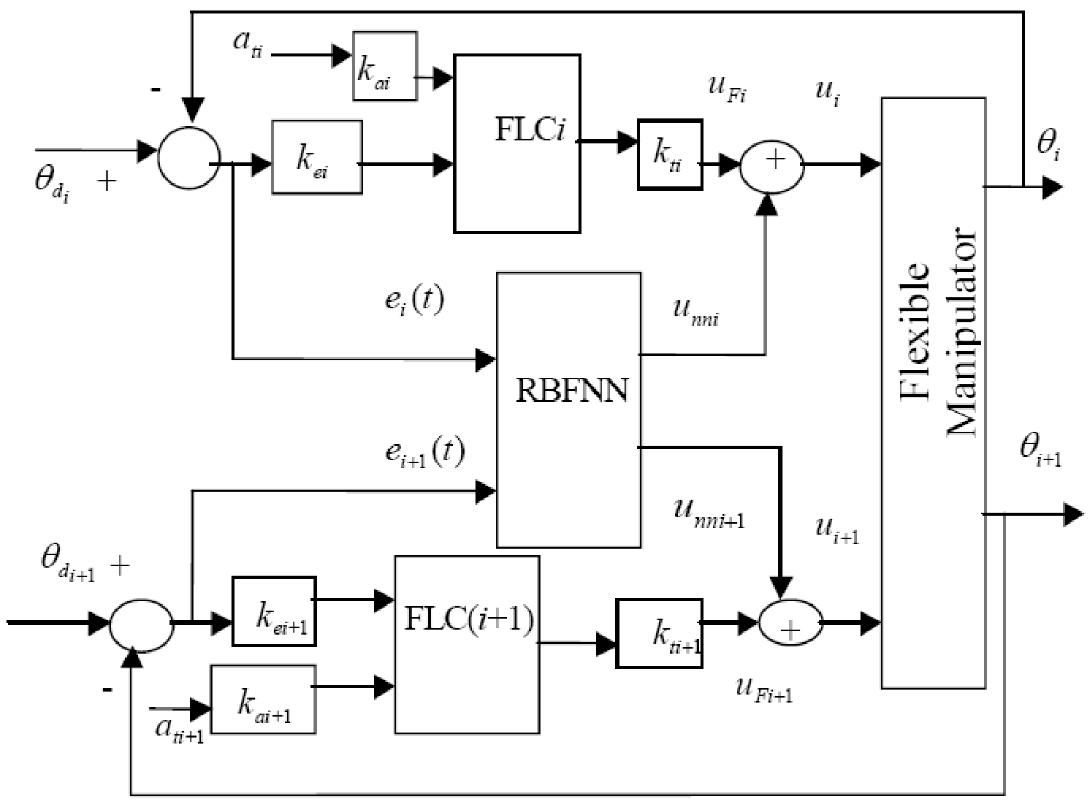

Fig. 3 Structure of Hybrid Fuzzy Neural Scheme

The radial basis function neural network (RBFNN), whose complexity lies midway between that of MLP and B-Spline networks, can be used to obtain the relative benefits of both these networks. Although the nonlinearities of RBFNN schemes have global support, the energy of the basis function is mostly local and the parameters between the input and the hidden layers are fixed. Therefore, linear optimisation schemes can be used for a RBFNN to minimize the training cost function, thus enabling on-line convergence to a global minimum. The hybrid fuzzy neural control scheme adopted (Fig.3) therefore uses a RBFNN to compensate the dynamic coupling effect of a multi-link flexible manipulator system.

The net torque u_i for the ith actuator is:

$$u_i = u_{nni} + u_{Fi} \tag{7}$$

where u_{Fi} is the control action generated by ith fuzzy system.

5.1 Training of radial basis function neural network

The centres and widths of the Gaussian functions are kept fixed during the training of the RBFNN, producing a fixed mapping between the input and the hidden. The output weights of the RBFNN can be updated using any instantaneous learning algorithms such as least mean square (LMS) or normalised least mean square (NLMS) techniques. In LMS, the weight vector is updated as:

$$w_i = w_i(t-1) + \eta' e_{net}(t) a_i(t) \tag{8}$$

where a_i is the output of the ith Gaussian function, η' is the learning rate, w_i is the vector of output weights, and e_{net} is the vector of error signals between the desired neural network outputs and the actual outputs.

In the case of the NLMS algorithm, the magnitude of the weight change is normalized by the magnitude of the transformed input vector and the learning rule for the weights is:

$$w_i = w_i(t-1) + \eta e_{net}(t) \frac{a_i(t)}{a^T(t)a(t)} \tag{9}$$

Comparing Eq.s (8 and 9), it is clear that the NLMS learning rule is equivalent to the LMS algorithm when $\eta = \dfrac{\eta' e_{net}}{\|a(t)\|_2^2}$ $\forall t$. Therefore, the search directions of the weight updates are the same in both training algorithms but the step sizes are different. Thus, NLMS gives more stable learning with bounded weights and a faster convergence can be achieved [29]. For this reason, the NLMS technique was used to adjust the weights between the hidden-layer and the output layer of the RBFNN.

Unfortunately, as the desired network outputs are not known, e_{net} required in Eq. (9) cannot be calculated. A solution to this is to approximate e_{net} from the error

$e(t)$ obtained from the difference between the actual and desired position of the manipulator:

$$e(t) = \begin{bmatrix} \theta_i(t) - \theta_{di}(t) \\ \theta_{i+1}(t) - \theta_{di+1}(t) \end{bmatrix} \tag{10}$$

The learning rate η is kept fixed during the training process, but its correct choice is important to the speed of the NLMS training.

6. IMPLEMENTATION OF, RESULTS AND DISCUSSIONS

The performance of the multivariable fuzzy control (FLC) and hybrid fuzzy neural control (HFNC) schemes developed were compared for a two-flexible-link robot manipulator (Fig.4) using the parameters given in Table 1 [35, 36].

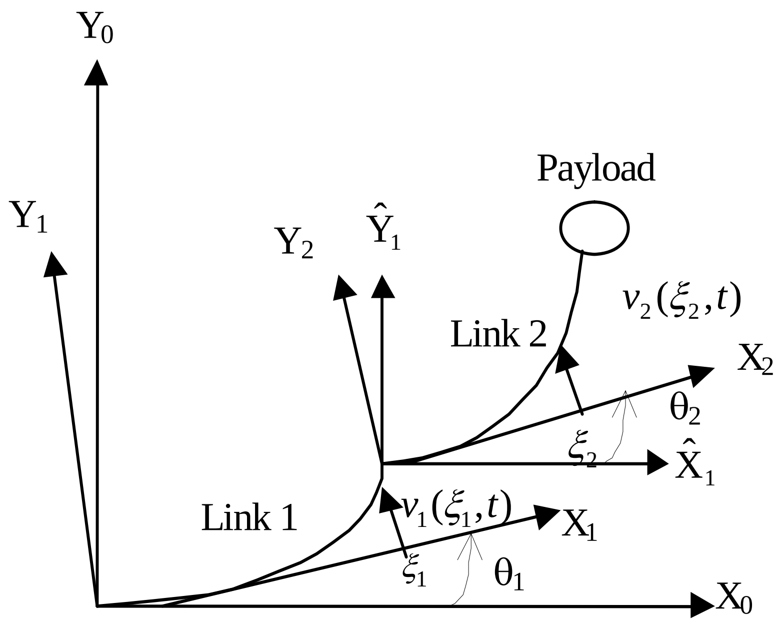

Fig.4 Schematic Diagram of a two-link flexible robot manipulator

Table 1 Parameters of the manipulator

Parameter	Value
Mass density (ρ)	0.2 Kgm^{-1}
Flexural rigidity(EI)	1.0 Nm^2
Length (l)	0.5m
Rotor and hub Inertia(I_r)	0.02 Kgm^2
Payload mass (M_p)	0.1 Kg
Payload Inertia (I_p)	0.005 Kgm^2
Damping constants (d_{11}, d_{12}, d_{13}, d_{14})	0.015, 0.02, 0.015, 0.02

For the purpose of comparison, a conventional, Lyapunov-based, PD adaptive controller (PDAC) [30] was also simulated. For the simulation, the initial conditions for the joints and flexible co-ordinates of the manipulator were assumed to be zero, i.e. $\theta = [0 \quad 0]$, $\dot{\theta} = [0 \quad 0]$, $q = [0 \quad 0]$ and $\dot{q} = [0 \quad 0]$. The manipulator was commanded to move from an initial position $\theta_s = [0 \quad 0]^T$ to a final position $\theta_d = \begin{bmatrix} \frac{\pi}{4} & \frac{\pi}{6} \end{bmatrix}^T$ rad.

6.2 Design of PD adaptive controller (PDAC)

To determine the gains of PDAC, let the flexible links be assumed rigid. Then, applying joint PD control leads to the following rigid motion error equation [30]:

$$I_{ei}\ddot{e}_i(t) + k_{vi}\dot{e}_i(t) + k_{pi}e_i(t) = 0 \quad i = 1,2. \tag{11}$$

where I_{ei} denotes the equivalent inertia of the ith joint, $e_i = \theta_i - \theta_{di}$. The gains of the PDAC were determined using the following formulas.

$$\left.\begin{aligned} k_{pi} &= I_{ei}\omega_{ni}^2 \\ k_{vi} &= 2I_{ei}\omega_{ni} \end{aligned}\right\} \tag{12}$$

where ω_{ni} is the natural frequency of the ith link. Using Eq. (14) with $\omega_{n1} = 1.5 \; rad/\sec$ and $\omega_{n2} = 1.5 \; rad/\sec$, $I_{e1} = 0.15883 kgm^2$ and $I_{e2} = 0.1338 kgm^2$, the gains of the PD control were found as: $k_{p1} = 0.3573$, $k_{p2} = 0.3011$, $k_{v1} = 0.4765$ and $k_{v2} = 0.40$. The controller gains k_{a1}, k_{a2} were set as: $k_{a1} = 1.5$ and $k_{a2} = 3.0$ after manual tuning to achieve good performance.

6.3 Design of multivariable fuzzy logic controller (FLC)

In the design of the multivariable FLC, triangular membership functions on a normalised universe of discourse of the input and output variables were used, as shown in Figs. 5 and 6.

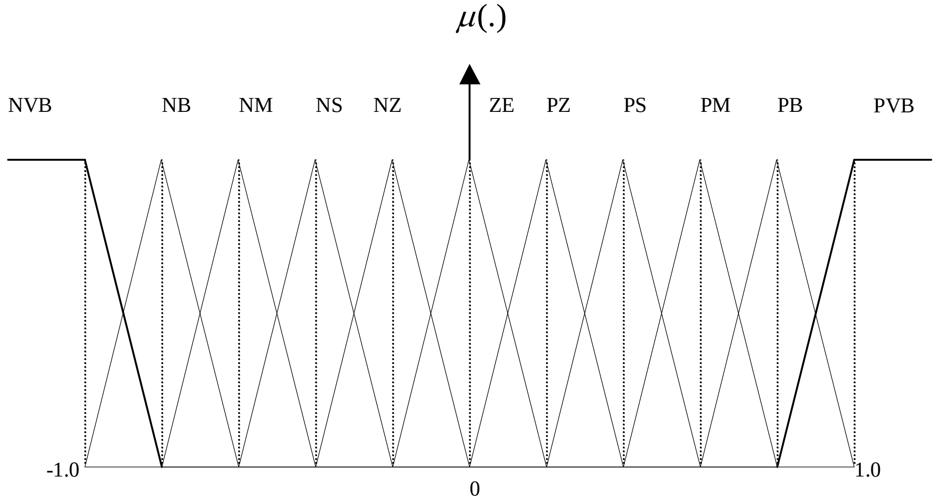

Fig. 5 Membership Functions for the first FLC Input and Output variables

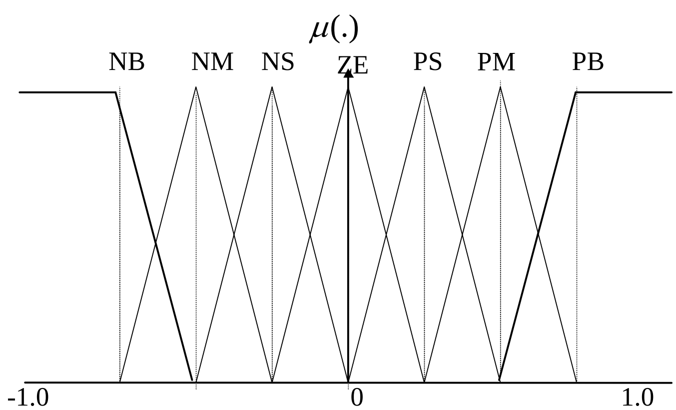

Fig. 6 Membership Functions for the Second FLC Input and Output variables

For the first FLC, the fuzzy term set for the input variables $(e_1(t), a_{t1}(t))$ and output variable $\theta_1(t)$ was assumed to have same cardinality of 11 as: $F = \{NVB, NB, NM, NS, NZ, ZE, PZ, PS, PM, PB, PVB\}$, where NVB, NB, NM, NS, NZ, ZE, PZ, PS, PM and PB denote Negative Very Big, Negative Big, Negative Medium, Negative Small, Negative Zero, Zero, Positive Zero, Positive Small, Positive Medium and Positive Big respectively. For the second FLC, there are three inputs and, to keep the rule base to a reasonable size, 7 fuzzy sets were used for the inputs ($e_2(t)$, $a_{t1}(t)$ and $a_{t2}(t)$) and output, $\theta_2(t)$ such that $\mathrm{F} = \{NB, NM, NS, ZE, PS, PM, PB\}$, where NB, NM, NS, ZE, PS, PM and PB denote Negative Big, Negative Medium, Negative Small, Zero, Positive Small. The decision-making fuzzy IF-THEN rules are necessary for successful operation of the fuzzy control system, but their derivation is very difficult. There are many possible ways to derive these fuzzy control rules [33,34], but in this case general control engineering intuition was used to derive the rules [11] in the following manner.

For the first link, if there is a positive error i.e., $e_1(t) > 0$ and a positive acceleration $a_{t1}(t) > 0$, then the controller will input a positive torque $u_1 > 0$ for this situation, so that the link is not properly aligned but is moving in the proper direction. As the error and the acceleration decrease, the controller applies smaller torque to avoid overshoot. Table 2 gives the complete listing of the rule set for the first link FLC. Similar rules were derived for the second link, although it should be noted that the second FLC involves the tip acceleration of the first link. Therefore, the rule base is a three-dimensional array. This is shown in parts through Table 2 to 10.

Table 2 Coupled FLC Rule Base for the First Link

Torque	Position Error										
Tip Acc.	***NVB***	***NB***	***NM***	***NS***	***NZ***	***ZE***	***PZ***	***PS***	***PM***	***PB***	***PVB***
NVB	NVB	NVB	NVB	NB	NB	NM	NM	NS	NS	NZ	ZE
NB	NVB	NVB	NB	NB	NB	NM	NS	NS	NZ	ZE	PZ
NM	NVB	NB	NB	NM	NM	NS	NS	NZ	ZE	PZ	PS
NS	NB	NB	NM	NM	NM	NS	NZ	ZE	PZ	PS	PS
NZ	NB	NM	NM	NS	NS	NZ	ZE	PZ	PS	PS	PM
ZE	NB	NM	NS	NS	ZE	ZE	ZE	PZ	PS	PM	PB
PZ	NM	NS	NS	NZ	ZE	PZ	PS	PS	PM	PM	PB
PS	NS	NS	NZ	ZE	PZ	PS	PS	PM	PM	PB	PVB
PM	NS	NZ	ZE	NS	PS	PS	PM	PM	PB	PB	PVB
PB	NZ	ZE	PZ	NM	PS	PM	PM	PB	PB	PVB	PVB
PVB	ZE	PZ	PS	PB	PM	PM	PB	PB	PVB	PVB	PVB

Table 3 Coupled FLC Rule Base for the Second Link, corresponding to First Link Tip Acceleration (NB)

Torque	Second Link Tip Acceleration						
Error	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NM	NM	NS	ZE
NM	NB	NB	NM	NS	NS	ZE	PS
NS	NB	NM	NM	NS	ZE	PS	PS
ZE	NM	NM	NS	ZE	PS	PM	PM
PS	NM	NS	ZE	PS	PS	PM	PM
PM	NS	ZE	PS	PS	PM	PM	PM
PB	ZE	PS	PS	PM	NM	PM	PB

Table.4 Coupled FLC Rule Base for the Second Link, corresponding to First Link Tip Acceleration (NM)

Torque	Second Link Tip Acceleration						
Error	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NM	NM	NS	ZE
NM	NB	NB	NM	NS	NS	ZE	PS
NS	NB	NM	NM	NS	ZE	PS	PS
ZE	NM	NM	NS	ZE	PS	PM	PM
PS	NM	NS	ZE	PS	PS	PM	PM
PM	NS	ZE	PS	PS	PM	PM	PM
PB	ZE	PS	PS	PM	PM	PM	PB

Table 5 Coupled FLC Rule Base for the Second Link, corresponding to First Link Tip Acceleration (NS)

Torque	Second Link Tip Acceleration						
Error	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NM	NM	NM	NS	ZE	PS
NS	NM	NM	NM	NS	ZE	PS	PS
ZE	NM	NM	ZE	ZE	ZE	PS	PM
PS	NM	NS	ZE	PS	PM	PM	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PM	PB	PB	PB

Table 6 Coupled FLC Rule Base for the Second Link, corresponding to First Link Tip Acceleration (ZE)

Torque	Second Link Tip Acceleration						
Error	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NB	NB	NM	ZE
NM	NB	NB	NB	NB	NS	ZE	PS
NS	NB	NM	NM	NS	ZE	PS	PM
ZE	NM	NS	ZE	ZE	ZE	PS	PM
PS	NM	NS	ZE	PS	PM	PM	PB
PM	NS	ZE	PM	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

Table 7 Coupled FLC Rule Base for the Second Link, corresponding to First Link Tip Acceleration (PS)

Torque	Second Link Tip Acceleration						
Error	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NM	NM	NS	ZE
NM	NB	NB	NM	NM	NS	ZE	PS
NS	NB	NM	NM	NS	ZE	PS	PM
ZE	NM	NS	ZE	ZE	ZE	PS	PM
PS	NM	NS	ZE	PS	NM	PB	PB
PM	NS	ZE	PS	PM	NB	PB	PB
PB	ZE	PS	PM	PB	NB	PB	PB

Table 8 Coupled FLC Rule Base for the Second Link, corresponding to First Link Tip Acceleration (PM)

Torque	Second Link Tip Acceleration						
Error	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NB	NM	NM	NS	NS	NS	ZE
NM	NB	NM	NM	NS	NS	ZE	PS
NS	NM	NM	NS	NS	ZE	PS	PM
ZE	NM	NS	NS	ZE	PS	PS	PM
PS	NS	NS	ZE	PS	PS	PM	PB
PM	NS	ZE	PS	PS	PM	PB	PB
PB	ZE	PS	PM	PM	PB	PB	PB

Table 9 Coupled FLC Rule Base for the Second Link, corresponding to First Link Tip Acceleration (PB)

Torque	Second Link Tip Acceleration						
Error	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NM	NM	NM	NS	NS	NS	ZE
NM	NM	NM	NS	NS	NS	ZE	PS
NS	NM	NM	NS	NS	ZE	PS	PM
ZE	NM	NS	NS	ZE	PS	PS	PM
PS	NS	NS	ZE	PS	PS	PM	PM
PM	NS	ZE	PS	PS	PM	PM	PB
PB	ZE	PS	PS	PM	PM	PB	PB

Table 10 Coupled FLC Rule Base for the First Link

Torque	Position Error										
Tip Acc.	***NVB***	***NB***	***NM***	***NS***	***NZ***	***ZE***	***PZ***	***PS***	***PM***	***PB***	***PVB***
NVB	NVB	NVB	NB	NB	NM	NM	NS	NB	NM	NZ	ZE
NB	NVB	NB	NB	NM	NM	NM	NZ	NM	NS	ZE	PZ
NM	NB	NB	NM	NM	NM	NM	ZE	NS	ZE	PZ	PS
NS	NB	NM	NM	NM	NS	NS	PZ	ZE	PS	PS	PS
NZ	NB	NM	NM	NS	NS	NZ	PS	NS	PZ	PS	PM
ZE	NB	NM	NS	NZ	ZE	ZE	PS	NM	PS	PM	PB
PZ	NM	NS	NS	NZ	ZE	ZE	PM	PB	PS	PM	PB
PS	NS	NS	NZ	ZE	PZ	NS	PM	ZE	PM	PM	PB
PM	NS	NZ	ZE	PZ	PS	NS	PM	NS	PM	PB	PB
PB	NZ	ZE	PZ	PS	PS	NM	PB	NM	PB	PB	PVB
PVB	ZE	PS	PS	PS	PM	NM	PB	PB	PB	PVB	PVB

It may be noted here that the rules are shown in multi-dimensional arrays to achieve clarity in presentation. But, as indicated in the introduction, an alternative way of storing rules in a multi-dimensional array of fixed dimension was adopted with index information to specify the location in a virtual multi-dimensional array. Using this approach, it is necessary to store only the array entries that are actually used rather than the full array. The rule entries are stored consecutively and the order in which they are stored is not important. For example, the first rule may refer to a location (3, 2,1) in a virtual rule matrix and the next consecutively stored rule may refer to location (3,1,1). The data structure used for the rules of the fuzzy system in the C programming language is given in Fig.6. The one-dimensional array ***inp_ind*** holds the index values for the input variables. The one-dimensional array ***in_fuzset*** holds the index values for the corresponding fuzzy sets. These are the index values that specify the location in the virtual rule matrix. Therefore, for a small price in storing the index information in this way, there is a huge benefit in flexibility and ease of implementation of the fuzzy system. This implementation has the ability to handle rule matrices of arbitrary dimension.

A compositional rule of fuzzy inference was used to compute the outcome of each fuzzy rule. A centre of gravity (COG) defuzzification method [33, 34], was used to convert the results of fuzzy inference to crisp control action. The coupled fuzzy controller gains were set by a manual tuning strategy similar to that described in [2]: $k_{e1} = 0.75$, $k_{a1} = 0.15$, $k_{t1} = 5.0$, $k_{e2} = 0.55$, $k_{a2} = 0.19$, $k_{a12} = 0.58$, $k_c = 0.35$ and $k_{t2} = 3.5$.

6.4 Implementation of hybrid fuzzy neural control scheme (HFNC)

In this case, the rule base for the first link is assumed to be the same as for the coupled FLC (Table 1). However the rules for the second link, as shown in Table 9, are different. The FLC gains used in HFNC were set as: $k_{e1} = 0.8$, $k_{a1} = 0.05$, $k_{t1} = 5.5$, $k_{e2} = 0.6$, $k_{a2} = 0.9$ and $k_{t2} = 4.0$. Corresponding to the two position error signals and two compensating control signals, the input and output layer of the RBF neural network comprised of two nodes each. 100 nodes were chosen for the hidden layer. The learning rate for the weight tuning algorithm given in Eq. (14) was set to be 0.7 and this algorithm was implemented by using a trapezoidal integration method with a step size of 5 ms.

The inputs to the RBFNN were normalized in a two-dimensional input space $[-1,1]\times[-1,1]$. The centers of the RBFs were placed uniformly throughout the input space. For each basis function, the width parameter was chosen to have a value of 0.5.

```
typedef struct
{
int inp_ind [MAXIN];
in_fuzset [MAXIN];
out_fuzset;
} rule;
```

Fig.7 Data Structure for Rule Matrix

6.4 Comparison of controller performance

It can be observed from the joint motion trajectories of link1 shown in Fig.8(a), that HFNC has the fastest rise time and the PDAC the slowest. The PDAC also has the largest overshoot, whilst there is negligible overshoot with HFNC. The joint motion curve of PDAC shows a small steady state error but there are no steady state errors with either FLC or HFNC. Fig.8(b) and 8(c) show that the amplitudes of the first and second modes of vibration for link1 are largest for PDAC and smallest for HFNC.

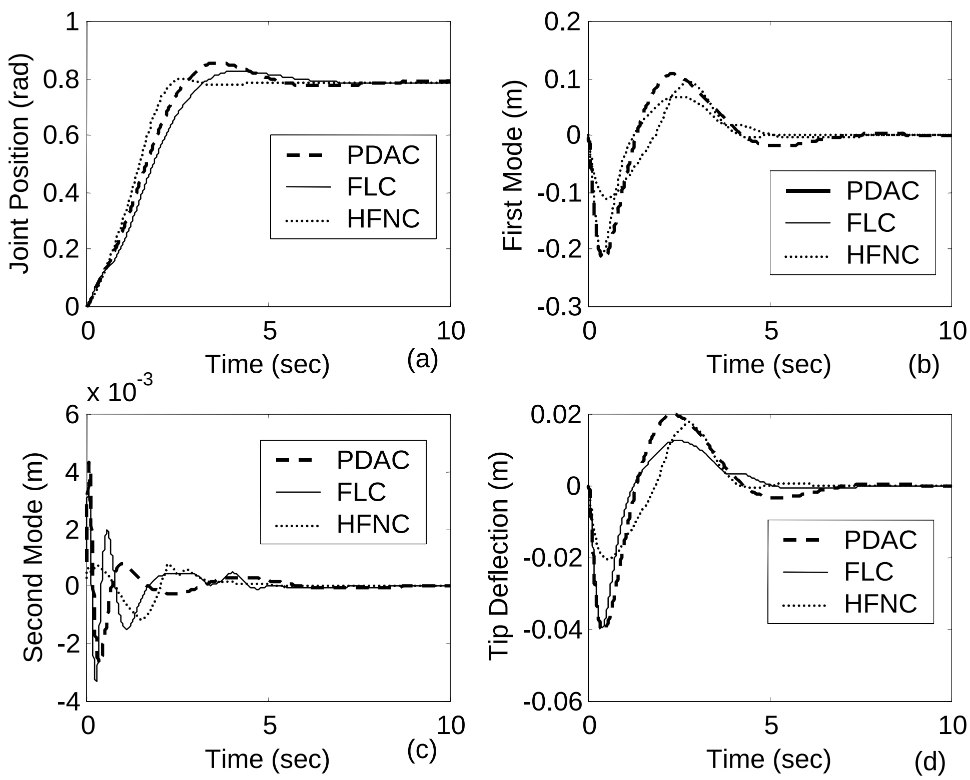

Fig. 8 Comparison of Joint Position, Modes of Vibration and Tip Deflection Trajectories of Link 1

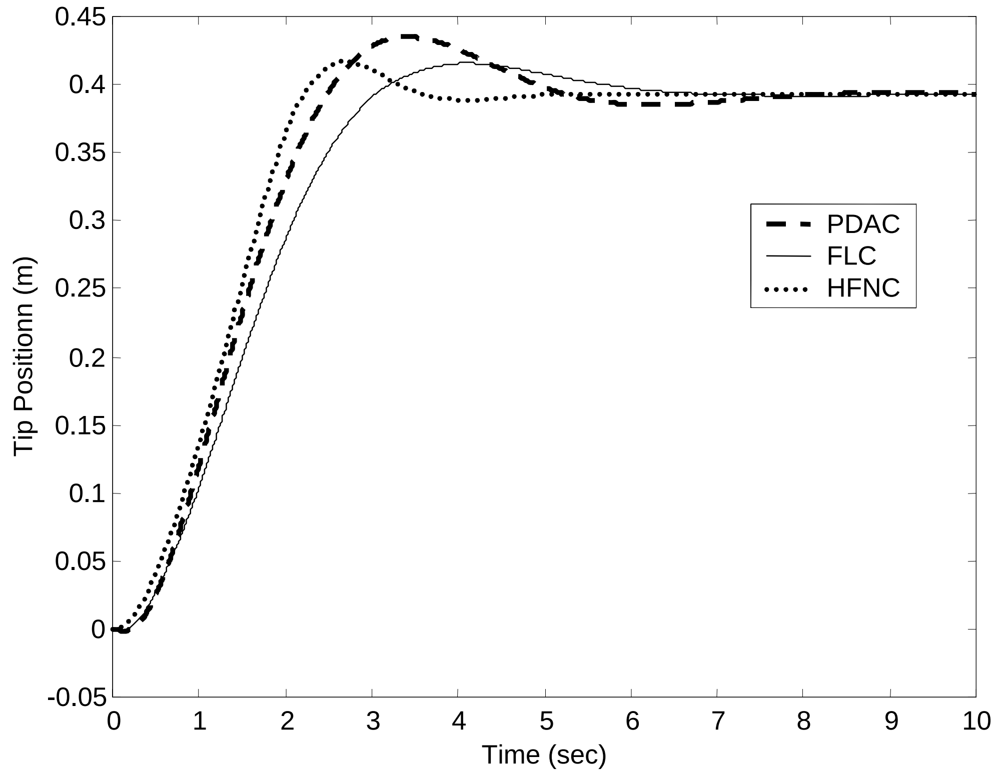

Fig. 9 Comparison of Tip Position Trajectories for the First Link

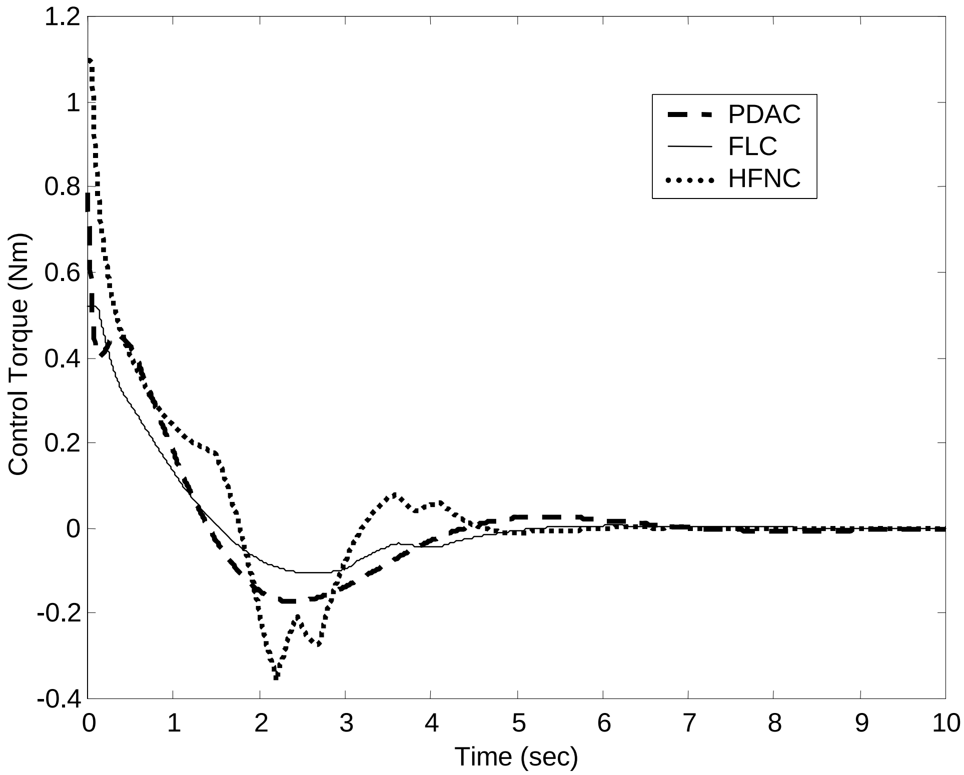

Fig.10 Comparison of Torque Profiles of the Control Schemes for the First Link

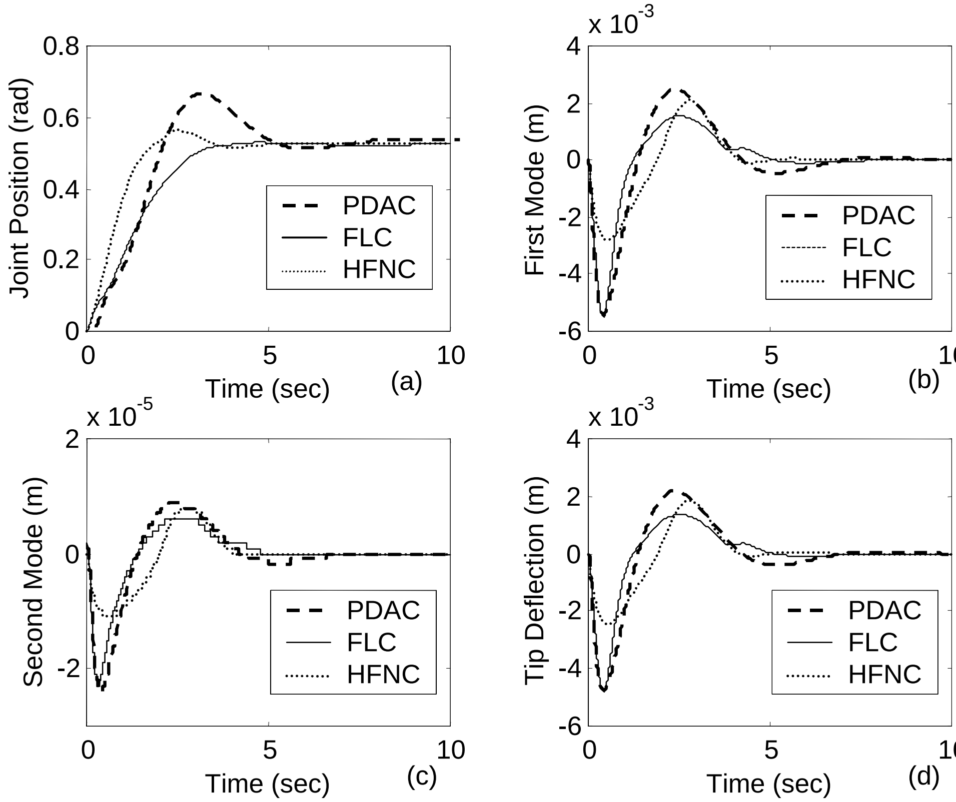

Fig.11 Comparison of Joint Position, Modal Vibration and Tip Deflection Trajectories of Link 2

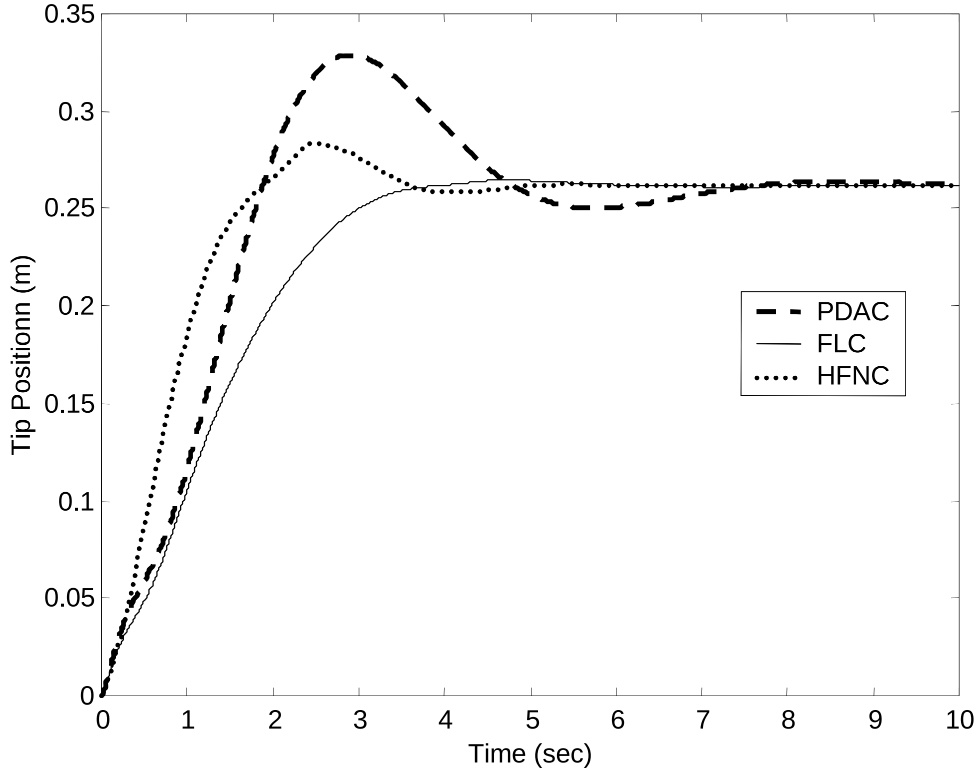

Fig. 12 Comparison of Tip Position Trajectories for the Second Link

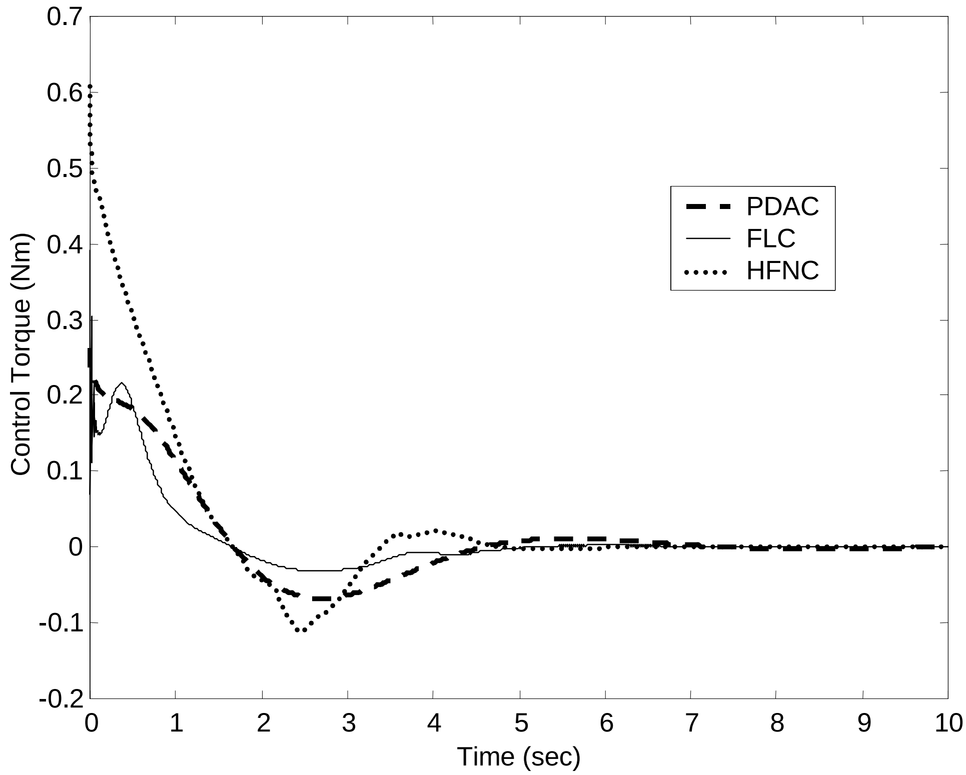

Fig. 13 Comparison of Torque Profiles of the Control Schemes for the Second Link

Also, HFNC also takes the least time to damp out these modal vibrations. Fig.8(d) shows that the tip deflection is corresponding is greater for PDAC compared to that for FLC and HFNC because of the excitation of both modes of vibration. It can be observed from the tip position trajectories given in Fig.10, that the link achieves a stable tip motion with HFNC and FLC without causing excessive vibration, while there is significant vibration in the case of PDAC. The uncoupled nature of the PDAC control strategy is the probable reason for the overshoot and vibration appearing in the trajectories. Fig. 10 compares the profiles of control torques generated for the first joint with these control schemes. The maximum values of joint torques are 1.1 Nm, 0.85Nm and 0.78 Nm respectively for HFNC, FLC and PDAC. Whilst the maximum control torque is a little greater for HFNC compared with the requirement of FLC and PDAC, this is acceptable in view of the better tip regulation performance of HFNC.

The second link motion control performances for HFNC, PDAC and FLC are given in figures 11 and 13. It is clear from the joint position trajectories shown in

Fig.11(a) that the rise time is fastest for HFNC and slowest for FLC. Overshoot is greatest with PDAC whereas FLC gives no overshoot. The first and second mode oscillations are greatest with PDAC and least with HFNC (Fig.11(b) and Fig. 11(c)). Because of this, PDAC gives the largest magnitude of tip deflection and takes more time to dampen the oscillations as compared to FLC and HFNC. HFNC damps out the tip deflection in the smallest time (Fig.11(d)). The tip position trajectories given in Fig.12 show that PDAC has the largest overshoot, while there is no overshoot in case of FLC. While HFNC has a small overshoot, it has the smallest amplitude of vibration compared to both PDAC and FLC. Therefore, it is clear that HFNC is the best control algorithm for tip positioning. The control torque signals of the controller are given in Fig.13. Like the first joint, the second joint in the case of HFNC also requires a little more energy for achieving the effective tip positioning compared to both PDAC and FLC.

7. CONCLUSIONS

The paper has developed a hybrid fuzzy neural control scheme for a two-link flexible manipulator and has compared the tip positioning control performance with that of a Lyapuvnov based PD controller and a multivariable fuzzy controller. The tip acceleration signal augmented to a joint PD control for a single-link flexible arm has been extended to a multi-link flexible manipulator, and its stability property has been verified using the Lyapuvnov theory. By using tip acceleration feedback, better vibration control was achieved but, due to the uncoupled nature of the control (no information from the other link was used in generating the control action), a little degradation in the tracking performance was observed. The multivariable fuzzy controller performs better than PDAC because it does consider the coupling effect, but implementation is difficult because of the large number of rules and the number of scaling factors to be tuned. The hybrid fuzzy neural control, in which a radial basis function neural network is supplemented by a conventional fuzzy logic controller compensates for the coupling effects between the links produces better performance. The implementation of the HFNC scheme requires joint position and tip acceleration information, but these can be obtained in a practical implementation using easily available position sensors and accelerometers. Since the fuzzy logic controller for each channel is used only for the primary control, the coupling NN provides the necessary control torque to compensate for the coupling effects. As the learning time of RBFNN depends greatly on faster execution of the FLC, an efficient implementation of FLC has therefore been adopted. The scale factor tuning for HFNC is easier than for FLC, as observed during simulation.

The stability of the fuzzy neural network control scheme has been ensured by normalizing the inputs to the neural network such that they are kept within the range of ± 1. The scheme is also stable during its operation, because the neural network has not been placed directly within the feedback loop. Rather, the NN controller simply supplements the FLC, which performs the major control in the primary loop. The resulting hybrid fuzzy neural control scheme provides good tip regulation performance as discussed in section 6.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the referees and the Editor for their useful comments and suggestions. The research was supported by the University of Sheffield, United Kingdom and AICTE, New Delhi, India.

REFERENCES

[1]. D.A. Linkens and H.O. Nyongesa, Learning systems in intelligent control: an appraisal of fuzzy, neural and genetic algorithm control applications, IEE Proc. of Control Theory Applications, (1994), 143(3), 367-386.

[2]. Lianfang Tian and Curtis Collins, Adaptive neuro-fuzzy control of a flexible manipulator, Mechatronics, (2005),15(10), 1305-1320

[3]. Rong-Jong Wai, and Meng-Chang Lee, Intelligent Optimal Control of Single-Link Flexible Robot Arm, IEEE Transactions on Industrial Electronics, (2004), 51(1), 201-220

[4]. H.A. Talebi, K. Khorasani, and R.V. Patel, Neural Network Based Control Schemes for Flexible-Link Manipulators: Simulations and Experiments, "Neural Networks", Special Issue, Neural Control and Robotics: Biology and Technology, (1998), 11, 1357-1377.

[5]. A. Mannani, H.A. Talebi, A fuzzy lyapunov-based control strategy for a macro-micro manipulator: experimental results", IEEE transactions on Control System Technology, (2007), 15(2), 375-383.

[6]. J. Lin, and F.L. Lewis, Fuzzy controller for flexible-link robot arm by reduced-order techniques, IEE Proceedings: Control Theory and Applications, (2002), 149(3), 177-187

[7]. B. Subudhi and A.S. Morris, Fuzzy and neuro-fuzzy approaches to control a flexible single-link manipulator, Journal of Systems & Control Engineering, Proc. IMechE, (2003), 217, 387-399.

[8]. E. Kubica and D. Wang, A fuzzy control strategy for flexible single-link robot, In: Proceedings of the IEEE International Conference on Robotics and Automation, 236-241, 1993.

[9]. J. Lin, and F.L. Lewis, Fuzzy controller for flexible-link robot arm by reduced-order techniques, IEE Proceedings: Control Theory and Applications, (2002), 149(3), 177-187

[10]. L.B. Gutierrez, F.L Lewis and J.A. Lowe, Implementation of neural network tracking controller for a single flexible link: comparison with PD and PID controllers, IEEE Trans. Industrial Electronics, (1998), 45(2), 307-318.

[11]. V.G. Moudgal, K.M. Passino and S. Yourkovitch, Rule-based control for flexible-link robot, IEEE Transaction on Control Systems Technology, (1994), l2(4), 393-405.

[12]. F.M. Caswara and H. Unbehauen, A neuro-fuzzy approach to the control of a flexible-link manipulator, IEEE Transactions on System, Man, and Cybernetics, (2002), 18(6), 932-944.

[13]. Jamil M. Renno, Inverse Dynamics Based Tuning of a Fuzzy Logic Controller for a Single-Link Flexible Manipulator, Journal of Vibration and Control, (2007), 13(12), 1741-1759.

[14]. Lianfang Tian, Jun Wang, and Zongyuan Mao, Constrained Motion Control of Flexible Robot Manipulators Based on Recurrent Neural Networks, IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, (2004), 34(3), 1541-1552

[15]. Samaneh Sarraf, Ali Fallah and T. Seyedena, A Neurofuzzy Controller for a Single Link Flexible Manipulator, Lecture Notes in Computer Science, Artificial Neural

Networks , ICANN 2007, Volume 4669/2007, Springer Berlin / Heidelberg, 621-629

[16]. Meng Joo Er and Yang Gao, Robust adaptive control of robot manipulators using generalized fuzzy neural networks,: Industrial Electronics, IEEE Transactions on Industrial Electronics, (2003), 50(3), 620- 628

[17]. Kiguchi, K. and Fukuda, T., Position/force control of robot manipulators for geometrically unknown objects using fuzzy neural networks, IEEE Transactions on, Industrial Electronics, (2000), 47(3), 641-649.

[18]. Wai, R., and Chen, P., Intelligent tracking control for robot manipulator including actuator dynamics via TSK-type fuzzy neural network, IEEE Transactions on Fuzzy Systems, (2004), 12(4), 552-560.

[19]. Sun F , Sun Z , Li L , Li H , Neuro-fuzzy adaptive control based on dynamic inversion for robotic manipulators, Fuzzy Sets and Systems, (2003), 134(1), 117-133.

[20]. Linda Z. Shi and Mohamed B. Trabia, Design and tuning of importance-based fuzzy logic controller for a flexible-link manipulator, Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, (2006), 17(3), 313 – 323.

[21]. K. Y. Kuo and J. Lin, Fuzzy logic control for flexible link robot arm by singular perturbation approach, Applied Soft Computing, (2002), 2(1), 24-38.

[22]. Hui D., Fuchun S., and Zengqi S, Observer-Based Adaptive Controller Design of Flexible Manipulators Using Time-Delay Neuro-Fuzzy Networks, Journal of Intelligent and Robotic Systems, (2002), 34(4), 453-466.

[23]. V. B. Nguyen and A. S. Morris, Genetic Algorithm Tuned Fuzzy Logic Controller for a Robot Arm with Two-link Flexibility and Two-joint Elasticity, Journal of Intelligent and Robotic Systems, (2007), 49(1),3-18.

[24]. Manish Kumar, Devendra P. Garg , Neuro-fuzzy control applied to multiple cooperating robots, Industrial Robot: An International Journal, (2005), 32(3), 234 – 239.

[25]. L. H. Tsoukalas and R. E. Uhrig, *Fuzzy and neural approaches in engineering*, (USA, John Wiley & Sons, 1997).

[26]. W.T. Miller, R.S. Sutton and P.J. Werbos, *Neural networks for control*, (Cambridge, and MA: MIT Press, 1990).

[27]. K.S. Narendra, Neural networks for control: theory and practice, Proc.IEE, (1996), 84(10), 1385-1406.

[28]. S. Omatu, M. Khalid and R. Yusof, *Neuro-control and its applications*, (Springer, London, 1996).

[29]. M., Brown and C. Harris, *Neuro-fuzzy Adaptive Modelling and Control*, (Prentice-Hall Int. Ltd., London, 1994).

[30]. S.S. Ge, T.H. Lee and G. Zhu, Improving regulation performance of a single-link flexible manipulator with strain feedback, IEEE Transactions on Robotics and Automation, (1998), 14(1), 181-185.

[31]. Mohammad Jamshidi, Autonomous control of complex systems: robotic applications, Applied Mathematics and Computation ,Volume 120, Issues 1-3, 10 May 2001, Pages 15-29

[32]. Gang Feng, A Survey on Analysis and Design of Model-Based Fuzzy Control Systems, IEEE Transactions on System, Man, and Cybernetics, (2006), 14(5), 676-696.

[33]. C.C. Lee, Fuzzy logic in control systems-part I, IEEE Transactions on System, Man, and Cybernetics, (1990), 20(2), 404-415.

[34]. C.C. Lee, Fuzzy logic in control systems-part II, IEEE Transactions on System, Man, and Cybernetics, (1990), 20(2), 416-433.

[35]. B. Subudhi and A.S. Morris, Dynamic modelling, simulation and control of a manipulator with flexible links and joints, Robotics and Autonomous Systems, (2002), 41(4), 257-270.

[36]. A. D. Luca and B. Siciliano, Closed-form dynamic model of planar multi-link lightweight robots. IEEE Trans. Systems, Man, and Cybernetics, (1991), 21(4), 826–839.