Modelling of neurofuzzy control of a flexible link

M Tınkır*, **Ü Önen,** and **M Kalyoncu**
Department of Mechanical Engineering, University of Selçuk, Konya, Turkey

The manuscript was received on 6 April 2009 and was accepted after revision for publication on 26 March 2010.

DOI: 10.1243/09596518JSCE785

Abstract: A modelling approach for neuro-fuzzy control of a single-link flexible robot manipulator that uses a computer-aided design (CAD) program is proposed. Initially, a CAD model of the flexible link is created using experimentally determined values of system parameters. This CAD model is then exported to MATLAB software and the Simulink/SimMechanics toolbox. An adaptive-network-based fuzzy logic controller is used for position and vibration control of the flexible link.

Experimental and simulation results are presented that validate the proposed approach.

Keywords: single-link flexible robot manipulator, CAD model, adaptive-network-based fuzzy logic controller, desired position signal, vibration control

1 INTRODUCTION

Robots are used to perform repetitive and labour-intensive jobs in many industrial applications. They also find use in dangerous environments such as arc welding shops, contaminated areas, nuclear reactors, outer space, and under water. For many tasks, currently available robots are not sufficiently economic, fast, or accurate. Flexible-link robotic manipulators have many advantages compared to conventional rigid robots and there is considerable interest in developing this technology. The dynamic analysis and control of flexible-link manipulators is much more complex than the analysis and control of the equivalent rigid manipulators. Therefore, the dynamic modelling [**1–8**] and control of flexible robots [**9–19**] have received considerable research attention. A first step towards the design of an efficient control strategy for these manipulators is the creation of accurate dynamic models that can characterize the flexibility of the links. Controller design solutions that minimize the effects of flexible displacements in light robots are of considerable interest in industrial and space applications that require accurate trajectory control. Controlled robot manipulators are usually designed to either reach a target or to follow a specified trajectory. In the first case, a short settling time is expected whereas in the tracking condition high-speed displacement of the robot arm is planned. Thus, strong control actions are applied and as a result, undesired controlled system features can appear if hidden vibrating modes become sufficiently excited.

Existing studies on flexible robot manipulators can be divided into two groups: those on dynamic effects and those on control effects. Several of the investigations on the control of flexible robot manipulators consider hierarchical fuzzy logic control [**20–34**] due to its advantages over other control techniques. Adaptive-network-based fuzzy logic control has not been considered for single-link flexible robot manipulators in most of the cited investigations despite its advantages that will be highlighted in this study.

Akbarzadeh-Totonchi *et al.* [**20**] developed a two-level hierarchical fuzzy controller for a single flexible link. The angular position measurements of the tip of the robot arm, and the torque of the motor were used as the basis of the first level of the hierarchical fuzzy controller. The second level controller monitored the behaviour of the robot arm and extracted features such as straight, oscillatory, and gently curved movements. Two-time-scale fuzzy logic controller were developed and applied to flexible-link robot arms in [**21**, **23**, **26**]. A singular perturbation approach was introduced to derive the slow and fast

**Corresponding author:* *Department of Mechanical Engineering, Faculty of Engineering and Architecture, University of Selçuk, Selçuk/Konya, Konya 42079, Turkey.*
email: mtinkir@selcuk.edu.tr

subsystems. The fast subsystem controller damped out the vibration of the flexible structure created by the two hierarchical fuzzy logic controllers. The slow subsystem fuzzy controller performed the control of the tracking of the trajectory. Caswara and Unbehauen [**22**] used a neurofuzzy controller as a nonlinear compensator for a flexible four-link manipulator. Two classes of neurofuzzy models, the Takagi–Sugeno fuzzy model and a rectangular local linear model network were tested as a potential feedforward controller to compensate the nonlinearities. Lin [**24**] applied a multi-time-scale fuzzy logic controller (FLC) to a robotic manipulator with link structural flexibility. The large-scale system was decomposed into a finite number of reduced-order subsystems using the singular perturbation approach. A hierarchical ordering of fuzzy rules was used to reduce the size of the inference engine. Chiang and Lin [**25**] used a hybrid position/force control approach and a multi-time-scale FLC to control the force and position of the end point of a flexible-link robot arm when the end effector moved on the constraint surface. Emara and Elshafei [**27**] considered robust control of robots including motor dynamics based on a dynamic game approach. Lin [**28**] developed an approach based on adaptive networks in a hierarchical fuzzy control approach applied to an active piezoelectric absorber of a smart panel. Jnifene and Andrews [**29**] considered the design and implementation of an active vibration control approach based on fuzzy logic and neural networks (NNs). The proposed controller was used to damp the end-point vibration in a single-link flexible manipulator mounted on a two-degree-of-freedom platform. The inputs to the FLC were the angular position of the hub and the end-point deflection of the flexible beam. Kamalasadan and Ghandakly [**30**] proposed a NN-based intelligent adaptive controller that possesses an intelligent supervisory loop. The scheme consisted of an online radial-basis-function NN in parallel with a model reference adaptive controller. Karimi and Yazdanpanah [**31**] and Karimi *et al.* [**32**] proposed a methodology for modelling a single-link flexible manipulator with an arbitrarily large (infinite) number of deflection modes based on the singular perturbation method. Lin *et al.* [**33**] and Lin and Huang [**34**] dealt with active damping control problems of robot manipulators with oscillatory bases. A two-time-scale FLC with a vibration stabilizer, which was required because the dynamics of the robotic system were strongly affected by disturbances due to the base oscillation, was proposed.

The aim of this study is to create an approach based on dynamic modelling and adaptive-network-based fuzzy logic control for the control of a flexible robot manipulator. The robot manipulator is considered to get as a single flexible link without tip mass. The physical configuration of the flexible manipulator considered in this study is shown in Fig. 1. The flexible link is assumed to act as an Euler–Bernoulli beam. A dynamic model of the flexible manipulator is obtained in terms of material properties using computer-aided design (CAD) software, so this detailed mathematical analyses are not needed. Then, an adaptive-network-based fuzzy logic controller (ANFLC) is applied to the system. The aim of using the FLC is to get the tip point of the flexible manipulator to the desired position and eliminate vibrations while it moves. The adaptive-network-based fuzzy inference system (ANFIS) uses a hybrid learning algorithm to identify the parameters of the Sugeno-type fuzzy inference systems. It applies a combination of the least-squares method and the backpropagation gradient descent method to train the fuzzy inference system (FIS) membership function parameters to emulate a given training data set. The control of the vibration and the tip position of the single-flexible-link manipulator are successfully realized by using this approach and the end-effector traces the desired position signal. Experimental and simulation results are presented, discussed, and compared to physical trends.

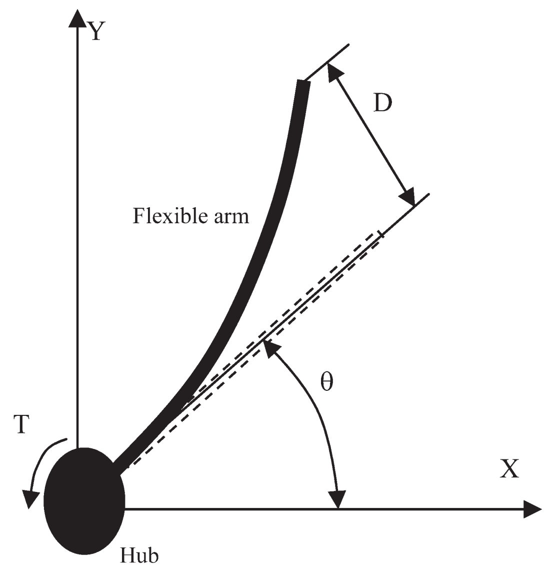


Fig. 1 Flexible manipulator with revolute joint

2 CAD MODEL OF THE FLEXIBLE-LINK SYSTEM

Figure 2 depicts the single-flexible-link arm module coupled to the SRV02 plant in the configuration used in the experimental studies. The module is attached to the load gear by two thumbscrews. The link is firmly attached to the module at its base. A strain gauge is mounted at the base of the link. This gauge is calibrated to output 1 V for each inch of tip deflection.

This system is similar in nature to control problems associated with the movement of structures in space where the weight constraints result in flexible structures that must be controlled using feedback techniques.

In the first step of the modelling study a model of the flexible-link module was created using SolidWorks CAD software. The properties used to create the model such as material type, dimensions, and thickness of the link were obtained from the experimental set-up. Then, the SRVO2 plant was drawn using the same software with the gears and DC motor being assumed to be revolute joints. This approach is verified by the comparison of the simulation and experimental results performed later in this paper. Figure 3 shows the CAD model of the rotary flexible link system.

After the SRVO2 plant and flexible link were modelled using the SolidWorks CAD software they were assembled together and the 'Solid to SimMechanics' software was used to translate the assembled system into MATLAB code. The SimMechanics model of the system is shown in Fig. 4. A three-dimensional view of the robot manipulator is shown in Fig. 5. The thickness and the length of the link are shown in Z and X axes respectively. The heights of the SRVO2 plant and the link are defined in the Y-axis. This model consists of two parts named the SRVO2 plant and flexible-link module. The SRVO2 is fixed to the ground and only the flexible link is allowed to move during the simulations. In Fig. 5, the SRVO2 plant is shown as an ellipsoid.

Using this approach means that the differential equations generally used in the dynamic modelling of such systems are no longer required. The physical and geometric characteristics of the flexible-link manipulator system are listed in Table 1. Figure 6 shows the modelling and control steps.

3 ANFLC DESIGN

In this study, an ANFLC is applied to the system. A block diagram of the controlled system is shown in Fig. 7. The ANFIS uses a hybrid learning algorithm to identify the parameters of the Sugeno-type fuzzy inference systems. It uses a combination of the least-

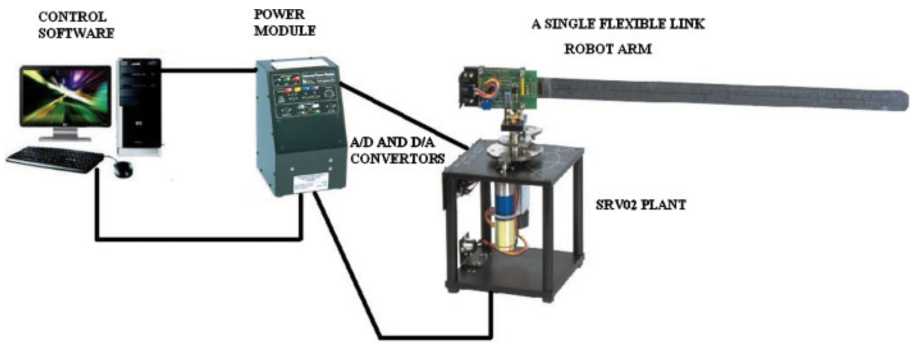


Fig. 2 The single-flexible-link arm module

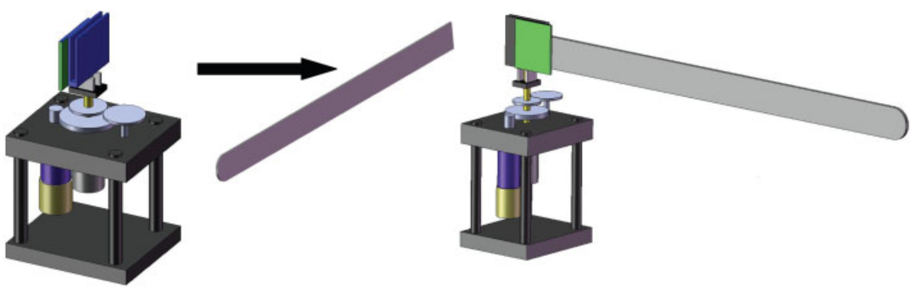


Fig. 3 CAD model of the flexible-link system

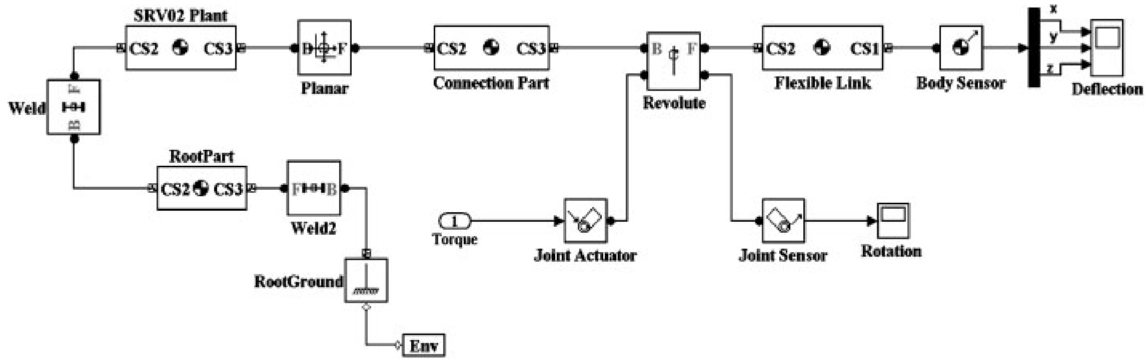


Fig. 4 SimMechanics model of the single-flexible-link manipulator system

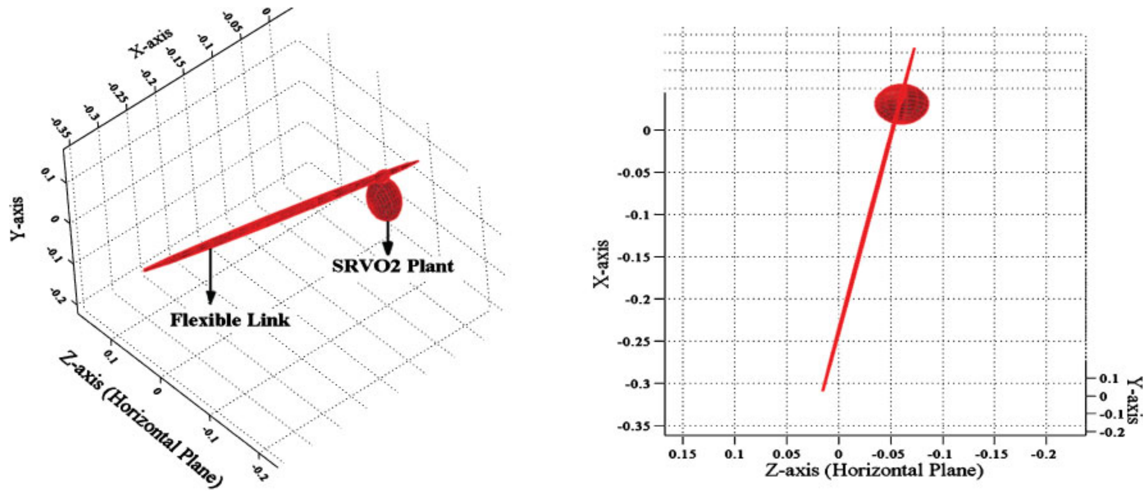


Fig. 5 Three-dimensional view of the SimMechanics model

Table 1 Physical and geometric characteristics of the system

Symbol	Parameter	Value
J_h	Flexible link inertia	0.0050 kg.m^2
L	Flexible arm length	0.300 m
h	Flexible arm height	0.0200 m
d	Flexible arm thickness	0.0008 m
ρ	Linear density	0.1333 kg/m
EI	Flexural rigidity	0.1621 N.m^2

squares method and the backpropagation gradient descent method to train the FIS membership function parameters to emulate a given training data set. The ANFLC is designed using the input and output data sets obtained from the proportional-derivative (PD) controller.

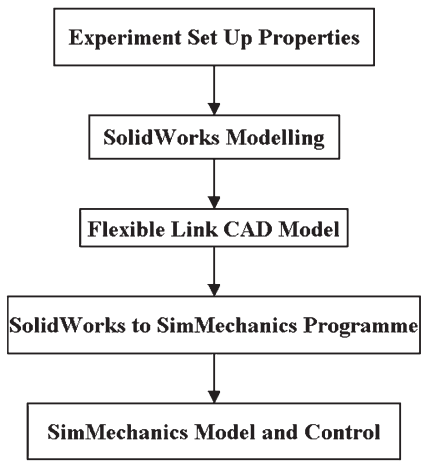


Fig. 6 Working plan

3.1 Membership functions and fuzzy rules

Two difficulties encountered while building the fuzzy controller are determining the shape of the membership functions and choice of the fuzzy rules. The key decision to be made is the way in which the controller output is generated using input fuzzy sets that consists of a database and a fuzzy control rule base.

A control system is said to be an adaptive fuzzy control system if a set of fuzzy rules are used to modify either the membership functions and/or rules of an existing FLC. The fuzzification process

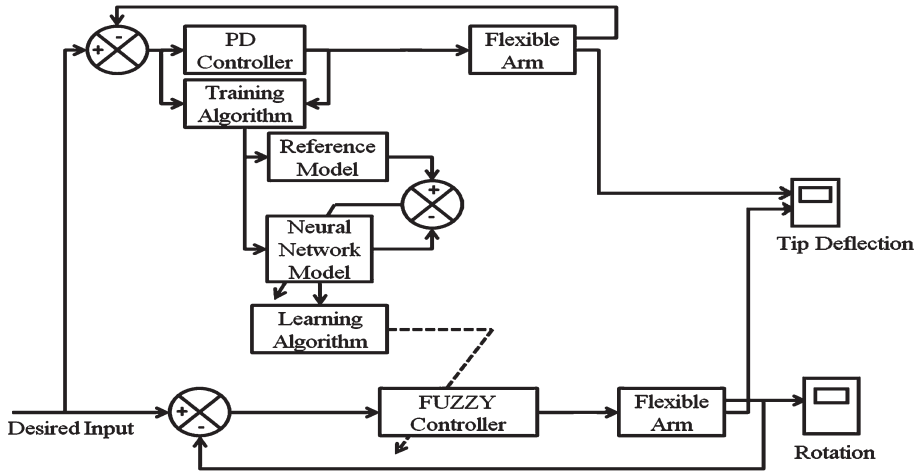


Fig. 7 Block diagram of the controlled system

uses membership functions to determine the degree of the inputs. The aim of the control action is to minimize the position error. The higher the error, the higher the control input. However, the rate of change of errors also affects the value of the control inputs. In a slow subsystem FLC, the position error is used in the control rules as a linguistic variable. This is defined as

$$\theta_e = \text{position error} = (\text{desired} - \text{actual})$$

In the ANFIS, the number and type of the membership functions is specified by the user. Then, the FIS rule base is generated automatically by the ANFIS. In this study, the ANFLC has three membership functions for the position error input. Gbell-type membership functions are used for the fuzzy inputs in the fuzzification process of the controller. A Sugeno-type fuzzy inference system is adopted and thus the controller's output is linear and not fuzzy. The ANFIS determines this output using given training data. Figures 8 and 9 show the membership

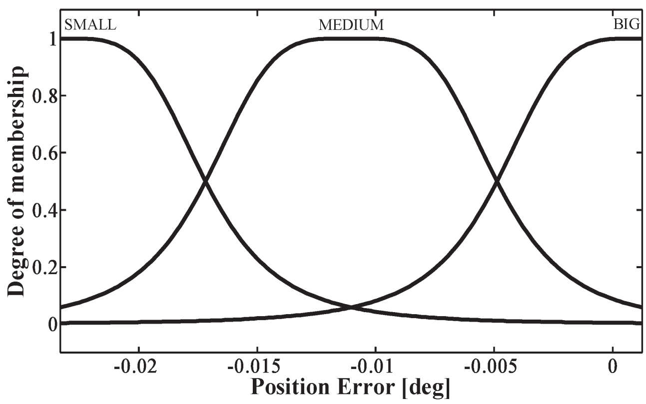


Fig. 8 Membership functions of the ANFLC

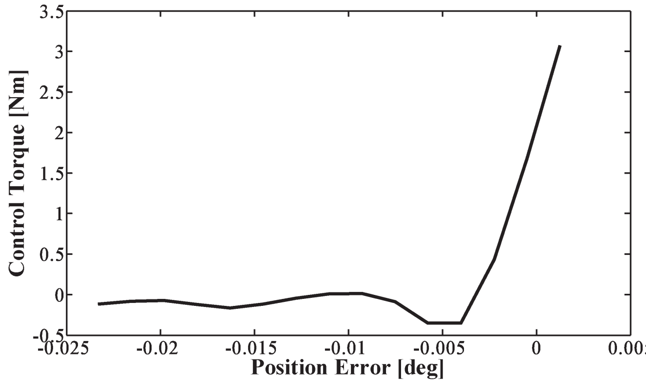


Fig. 9 Control torque of the ANFLC

functions and control torque of the ANFLC respectively.

The fuzzy rule base can be written as a look-up table. The rule base is composed of three rules. The ANFLC's rule base is shown in Table 2. The following example demonstrates the use of the natural language modelling approach for fuzzy control decision making. There are three fuzzy membership functions that correspond to small, medium, and large values of position error respectively. The control laws of the FLC consist of a complete set of these control rules defined as

$$\textit{If}\ \theta_e = S\ \textit{then Torque}1 = \text{Linear}\ [39.8\ 0.9234], \ldots$$

Table 2 Rule base of the ANFLC

	Position error (θ_e)		
Output	Small	Medium	Big
Control torque (Nm)	[39.8 0.9234]	[105.1 1.556]	[831.5 2.115]

3.2 Defuzzification process

Once the fuzzy controller is activated, the rule evaluation is performed and all the rules which are true are fired. Utilizing the true output membership functions, defuzzification is then applied to determine a crisp control action. The defuzzification action is to transform the control signal into an exact control output. For Sugeno-type inference, there is a choice of either the weighted average method or weighted sum defuzzification method. In this paper, the weighted average method is used

$$u=\frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \tag{1}$$

4 ADAPTIVE NN

NNs are composed of simple elements operating in parallel. These elements are inspired by biological systems. As in nature, the network function is determined largely by the connections between elements. It is possible to train a NN to perform a particular function by adjusting the values of the connections (weights) between elements. NNs are commonly adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown in Fig. 10.

Typically many input/target pairs are needed to train a network. NNs have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems.

The ANFIS architecture consists of five layers and in this paper the output of the nodes in each respective layer is represented by $O_{i,l}$ where i is the ith node of layer l.

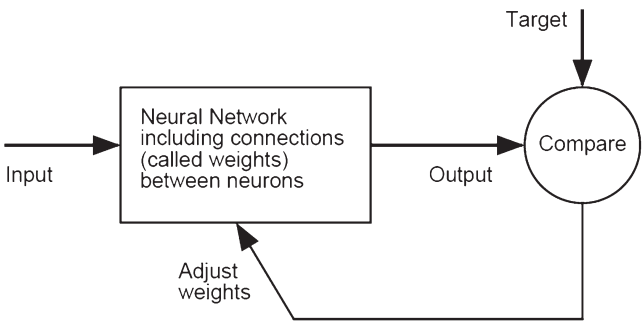


Fig. 10 Structure of the NN

Layer 1

Generate the membership grades

$$O_{i,l}=\mu_{A_i}(x),\ i=1,2 \tag{2}$$

or

$$O_{i,l}=\mu_{B_{i-2}}(y),\ i=3,4 \tag{3}$$

where x (or y) is the input to the node and A_i (or $B_{i\text{-}2}$) is the fuzzy set associated with this node.

Layer 2

Generate the firing strengths by multiplying the incoming signals and outputs the t-norm operator result

$$O_{2,}=w_i=\mu_{Ai}(x)\times\mu_{Bi}(y),\ i=1,2 \tag{4}$$

Layer 3

Normalize the firing strengths

$$O_{3,}=\overline{w}_i=\frac{w_i}{w_1+w_2},\ i=1,2 \tag{5}$$

Layer 4

Calculate the rule outputs based on the consequent parameters $\{p_i, q_i, r_i\}$

$$O_{4,}=\overline{w}_i f_i=\overline{w}_i(p_i x+q_i y+r_i) \tag{6}$$

Layer 5

Compute the overall outputs as the summation of all incoming signals

$$O_{5,}=\sum_i \overline{w}_i f_i=\frac{\sum_i w_i f}{\sum_i w_i} \tag{7}$$

The following steps were used to create the ANFLC:

Step 1: 1360 training and 700 checking data (from the PD controller) are obtained.

Step 2: The number and type of membership functions are determined.

Step 3: The hybrid learning algorithm and 40 epochs are chosen to train the network.

4.1 Hybrid learning algorithm

In this study, a feedforward hybrid learning algorithm was used for the NN part of the ANFLC shown in Fig. 11. Hybrid learning algorithms have been

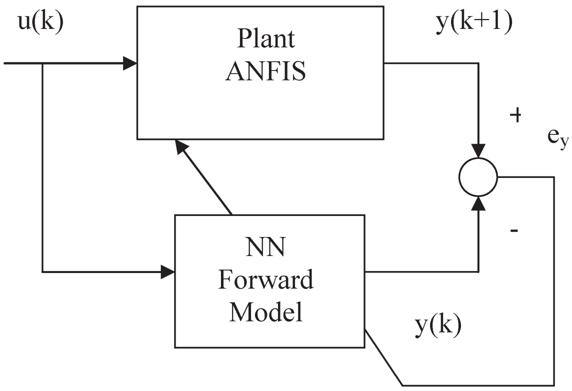


Fig. 11 Training the feedforward NN model

extensively described in the literature [**35–46**]. In the forward pass of the hybrid learning algorithm, the node outputs increase until layer 4 and the consequent parameters are identified by the least-squares method. When the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters.

The ANFIS is a fuzzy Sugeno-type model cast in the framework of adaptive systems to facilitate learning and adaptation [**35**]. Such a framework makes the ANFIS modelling more systematic and less reliant on expert knowledge. To present the ANFIS architecture, two fuzzy IF-THEN rules based on a first-order Sugeno model are considered:

Rule 1

If (x is A_1) and (y is B_1) *then* ($f_1 = p_1x + q_1y + r_1$)

Rule 2

If (x is A_2) and (y is B_2) *then* ($f_2 = p_2x + q_2y + r_2$)

where x and y are the inputs, A_i and B_i are the fuzzy sets, f_i are the outputs within the fuzzy region specified by the fuzzy rule, p_i; q_i and r_i are the design parameters that are determined during the training process [**45**]. The ANFIS architecture to implement these two rules is shown in Fig. 12, in which a circle indicates a fixed node and a square indicates an adaptive node.

$$\begin{aligned} f &= \frac{w_1}{w_1+w_2}f_1 + \frac{w_2}{w_1+w_2}f_2 \\ &= \overline{w}_1 f_1 + \overline{w}_2 f_2 \\ &= (\overline{w}_1 x)p_1 + (\overline{w}_1 y)q_1 + (\overline{w}_1)r_1 + (\overline{w}_2 x)p_2 + (\overline{w}_1 y)q_2 \\ &\quad + (\overline{w}_1)r_2 \end{aligned} \tag{8}$$

which is linear in the consequent parameters p_1, q_1, r_1, p_2, q_2, and r_2.

A hybrid learning rule [**35**] is presented which combines the gradient method and the least-squares estimate (LSE) to identify the parameters. For simplicity, it is assumed that the adaptive network under consideration has only one output

$$output = F(\bar{I}, S) \tag{9}$$

where $\bar{I}$ is the set of the input variables and S is the set of parameters. If there exists a function H such that the composite function $H \circ F$ is linear in some of the elements of S, then these elements can be identified by the least-squares method. More for-

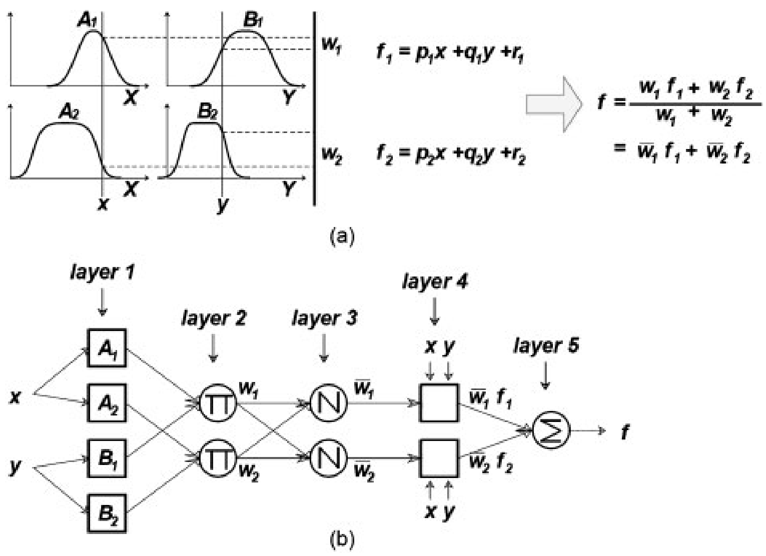


Fig. 12 (a) Takagi–Sugeno–Kang (TSK) fuzzy inference system with two inputs and two rules and (b) ANFIS architecture for first-order TSK model with two inputs

mally, if the parameter set S can be decomposed into two sets

$$S = S_1 \oplus S_2 \tag{10}$$

(where $\oplus$ represents direct sum) such that $H \circ F$ is linear in the elements of S_2, then on applying H to equation (9), it is possible to write

$$H(output) = H \circ F(\bar{I}, S) \tag{11}$$

which is linear in the elements of S_2. Now given values of elements of S_1, it is possible to substitute the P training data into equation (11) and obtain a matrix equation of the form

$$\mathbf{f} = [\,\overline{w}_1 x \quad \overline{w}_1 y \quad \overline{w}_1 \quad \overline{w}_2 x \quad \overline{w}_2 y \quad \overline{w}_1\,] \begin{bmatrix} p_1 \\ q_1 \\ r_1 \\ p_2 \\ q_2 \\ r_2 \end{bmatrix} = \mathbf{X}\boldsymbol{W} \tag{12}$$

where $\mathbf{X}$ is the input matrix and $\boldsymbol{W}$ is a weight vector (firing strength of each rule) whose elements are parameters in S_2. Let $|S_2| = M$, then the dimensions of $\mathbf{X}$, $\boldsymbol{W}$, and $\mathbf{f}$ are $P \times M, M \times 1$, and $P \times 1$, respectively. Since P (number of training data pairs) is usually greater than M (number of linear parameters), this is an overdetermined problem and generally there is no exact solution to equation (12). Instead, a LSE of $\mathbf{W}$, $\mathbf{W}^*$, is sought to minimize the squared error $\|\mathbf{X}\boldsymbol{W} - \mathbf{f}\|^2$ [**47**]. This is a standard problem that forms the grounds for linear regression, adaptive filtering and signal processing.

Lemma 1

The inverse $\mathbf{X}^{-1}$ of a matrix $\mathbf{X}$ exists only if $\mathbf{X}$ is square and has full rank. In this case, $\mathbf{f} = \mathbf{X}\boldsymbol{W}$ has the solution $\mathbf{W} = \mathbf{X}^{-1}\mathbf{f}$.

Lemma 2

The pseudo-inverse $\mathbf{X}^*$ (beware, it is often denoted otherwise) is a generalization of the inverse, and exists for any $m \times n$ matrix. It is assumed that $m \rangle n$, if $\mathbf{X}$ has full rank (n in this paper) then

$\mathbf{X}^* = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}$ and the solution of $f = \mathbf{X}\boldsymbol{W}$ is $\boldsymbol{W} = \mathbf{X}^*\mathbf{f}$

otherwise a pseudo-inverse ($\mathbf{X}^*$) of $\mathbf{X}$ is used to solve for $\boldsymbol{W}$

$$\boldsymbol{W} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{f} \tag{13}$$

where $\mathbf{X}^{\mathrm{T}}$ is the transpose of $\mathbf{X}$, and $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}$ is the pseudo-inverse of $\mathbf{X}$ if $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is non-singular. While equation (13) is concise in notation, it is expensive in computation when dealing with the matrix inverse and, moreover, it becomes ill-defined if $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is singular. As a result, sequential formulas are used to compute the LSE of $\mathbf{X}$. This method is more efficient (especially when M is small) and can be easily modified to an online version for a system with changing characteristics. Specifically, let the ith row vector of matrix $\mathbf{X}$ defined in equation (12) be $\boldsymbol{a}_i^{\mathrm{T}}$ and the ith element of $\mathbf{f}$ be $\boldsymbol{b}_i^{\mathrm{T}}$, then $\mathbf{W}$ can be calculated iteratively using the sequential formulas widely adopted in the literature [**48–51**]

$$\left.\begin{aligned} \boldsymbol{W}_{i+1} &= \boldsymbol{W}_i + \mathbf{S}_{i+1}\boldsymbol{a}_{i+1}\left(\boldsymbol{b}_{i+1}^{\mathrm{T}} - \boldsymbol{a}_{i+1}^{\mathrm{T}}\boldsymbol{W}_i\right) \\ \mathbf{S}_{i+1} &= \mathbf{S}_i - \frac{\mathbf{S}_i\boldsymbol{a}_{i+1}\boldsymbol{a}_{i+1}^{\mathrm{T}}\mathbf{S}_i}{1 + \boldsymbol{a}_{i+1}^{\mathrm{T}}\mathbf{S}_i\boldsymbol{a}_{i+1}},\ i = 0, 1, \ldots, P-1 \end{aligned}\right\} \tag{14}$$

where $\mathbf{S}_i$ is often called the covariance matrix and the least squares estimate $\boldsymbol{W}^*$ is equal to $\boldsymbol{W}_{\mathrm{P}}$. The initial conditions to bootstrap equation (14) are $\boldsymbol{W}_0 = 0$ and $\mathbf{S}_0 = \gamma\mathbf{I}$, where γ is a positive large number and $\mathbf{I}$ is the identity matrix of dimension $M \times M$. When dealing with multi-output adaptive networks (*output* in equation (9) is a column vector), equation (14) still applies except that $\boldsymbol{b}_i^{\mathrm{T}}$ is the ith row of matrix $\mathbf{f}$.

When the premise parameters are not fixed, the search space becomes large and the convergence of the training becomes slow. A hybrid algorithm combining the least-squares method and the gradient descent method is adopted to solve this problem. The hybrid algorithm is composed of a forward pass and a backward pass. The least-squares method (forward pass) is used to optimize the consequent parameters with the premise parameters fixed. Once the optimal consequent parameters are found, the backward pass starts immediately. The gradient descent method (backward pass) is used to optimally adjust the premise parameters corresponding to the fuzzy sets in the input domain. The output of the

ANFIS is calculated by using the consequent parameters found in the forward pass. The output error is used to adapt the premise parameters by means of a standard backward pass algorithm. It has been shown that the hybrid algorithm is highly efficient in training ANFISs [**35**].

In this study, the PD controller's input and output data are used to train the NN structure of the ANFLC. Checking data is chosen from some part of training data and the convergence of ANFLC is given in Fig. 13.

Fuzzy control is by far the most successful applications of fuzzy set theory and fuzzy inference systems. Due to the adaptive capability of ANFIS, its application to adaptive control and learning control are obvious [**35–46**]. In this paper the ANFIS is used to optimize the fuzzy IF-THEN rules and the membership functions to derive a more efficient fuzzy control. The proposed ANFIS model combines the NN adaptive capabilities and the fuzzy logic qualitative approach [**28**].

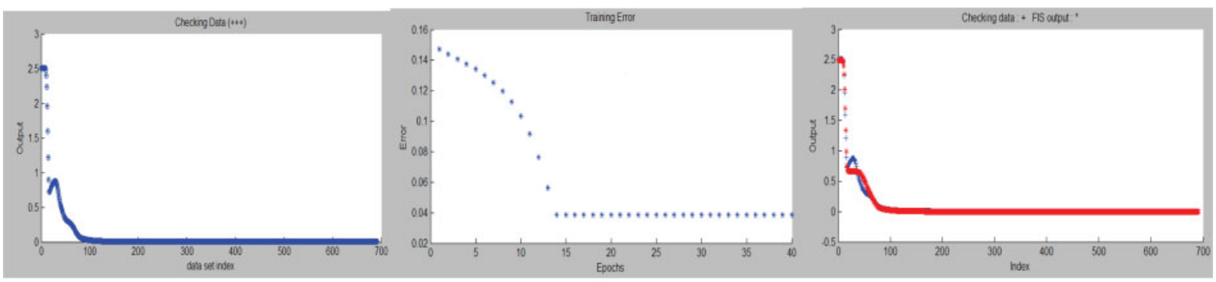
The proposed approach can replace almost any NN-based control system. For instance, the pioneering work of Narendra and Parthsarathy [**52**] that used NNs in adaptive control can be replicated using the ANFIS approach. Moreover, four of the generic NN control designs (i.e. supervised control, direct inverse control, neural adaptive control and back-propagation of utility), as proposed by Werbos [**53**] and Miller *et al.* [**54**], are also open for ANFIS applications. The active role of NNs in signal processing [**55**, **56**] also suggests the use of the ANFIS approach. The non-linearity and structured knowledge representation of ANFIS are the primary advantages over the classical linear approaches of adaptive filtering [**57**] and adaptive signal processing [**47**], used in areas such as identification, inverse modelling, predictive coding, adaptive channel equalization, adaptive interference (noise or echo) cancelling, etc. The most common design techniques for ANFLCs are derived directly from the NN counterpart methodology. However, certain design techniques apply exclusively to ANFIS.

4.2 The adaptive capability of the ANFLC

The adaptive capability of the ANFLC is improved by flexible manipulator with disturbances. It means that the controller responses to disturbances fast and appropriately. The ANFLC is able to easily adjust to varying system conditions (such as different input signals (step, sinusoidal, etc.), disturbances and non-linear system behaviour) and this feature makes it better than a PD controller. The optimization of the ANFLC depends on the selection of training, testing, and checking of the supplied data. The ANFLC uses these data to learn how it can change rules and membership functions. A typical fuzzy logic control technique does not have this adaptive learning capability and can only be changed by the user. The NN basis of the controller ANFLC eliminates this imperfection.

5 RESULTS AND DISCUSSION

The effectiveness of the proposed control scheme has been tested by simulation and laboratory experiments using a single-flexible-link manipulator. A photograph of the flexible robot manipulator is given in Fig. 14. The flexible link robot is a Quanser FLEXGAGE thin stainless steel beam with strain gauge mounted at its base. The gauge is calibrated to output 1 V per inch of tip deflection. The robot consists of a DC servo motor and a flexible link attached to the motor's shaft. The servo motor is a voltage-driven Quanser SRV02-ET model equipped with angular speed and position sensors. Interaction with the controlled plant is provided by a Quanser UPM 2405 power module and by Quanser MultiQ



a) b) c)

Fig. 13 (a) Input data, (b) convergence of ANFIS after training, and (c) checking of the ANFIS

Fig. 14 Photograph of the experimental set-up

PCI card, as shown in Fig. 14. WinCon software developed by Quanser allows communication with Simulink. This makes control by MATLAB possible. The schematic diagram of the system considered in this study is given in Fig. 15.

The objective in the experiment was to tune a controller for the single-flexible-link manipulator as it moves between -30° and 30°. A sinusoidal reference input signal with an amplitude of 30, and a frequency of 3.2 rad/s was used as the desired input signal of the system.

Initially the PD controller was used to control the single-flexible-link manipulator, and the input and output data collected during this experiment were used as the basis of the ANFLC. The gains of the PD controller were tuned using values determined through a trial and error method, ($K_p = 5$; $K_d = 4$ for experiment), ($K_p = 5$; $K_d = 3$ for simulation). In the simulation studies, the performances of the PD control and the ANFLC were obtained and compared with each other. For a desired motion, the controller was applied to the single-flexible-link robot arm. When the robot arm moves, the tip deflection of the arm must be close to zero. Figures 16 and 17 show the simulation results for the rotation and tip deflection for the PD control.

Rotation and tip deflection performances of the ANFLC control are given in Figs 18 and 19. From these results, it can be seen the ANFLC is more

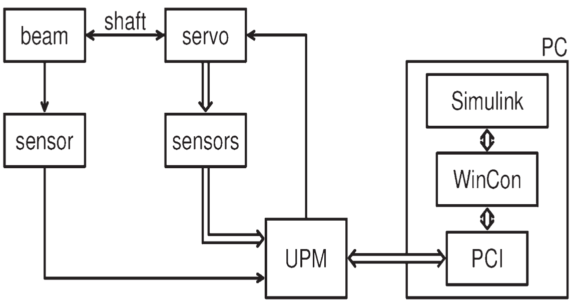


Fig. 15 Schematic diagram of the system

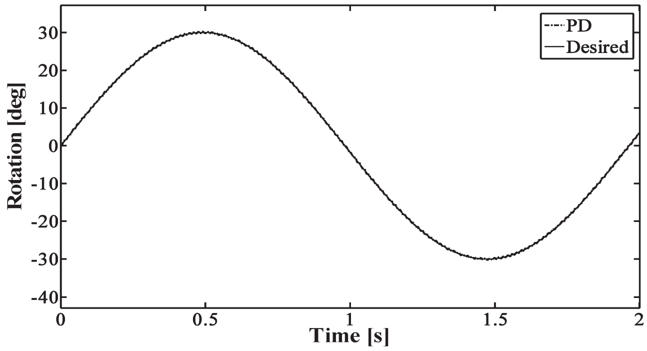


Fig. 16 Simulation of rotation under PD control

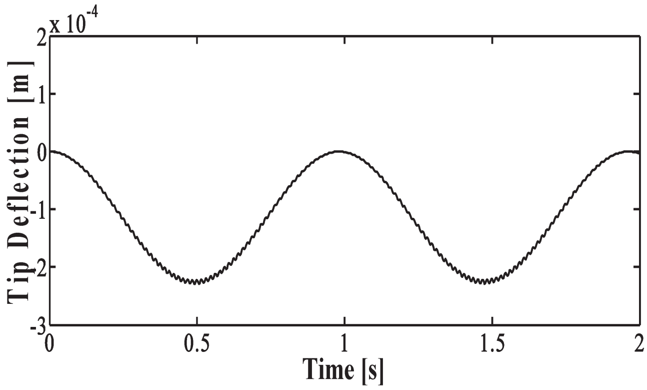


Fig. 17 Simulation of tip deflection under PD control

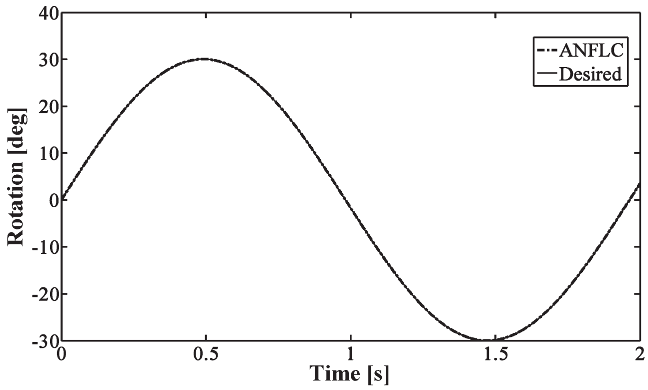


Fig. 18 Simulation of rotation under ANFLC control

effective than the PD controller. Tip deflection comparisons of the controllers show that the ANFLC is able to control the rotation of the arm with a maximum deflection of 2×10^{-4} m.

The experimental results show the effectiveness of the proposed controller in tracking the desired rotation (Fig. 20) and in reducing the end-point vibration of the flexible link (Fig. 21). From these figures it is clear that the CAD modelling and control results of the system are very close to the experimental values. These results validate the accuracy of

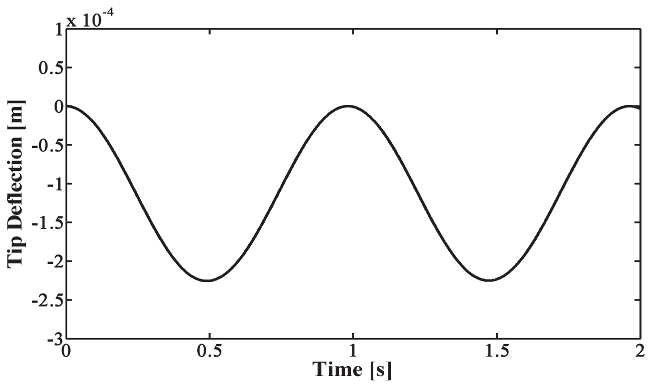


Fig. 19 Simulation of tip deflection under ANFLC control

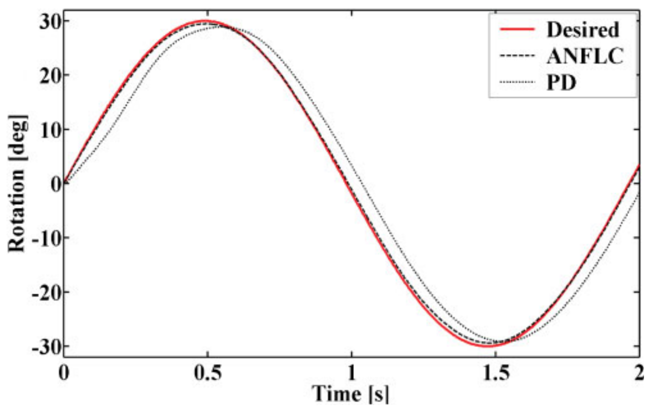


Fig. 20 Experimental results for the rotation control

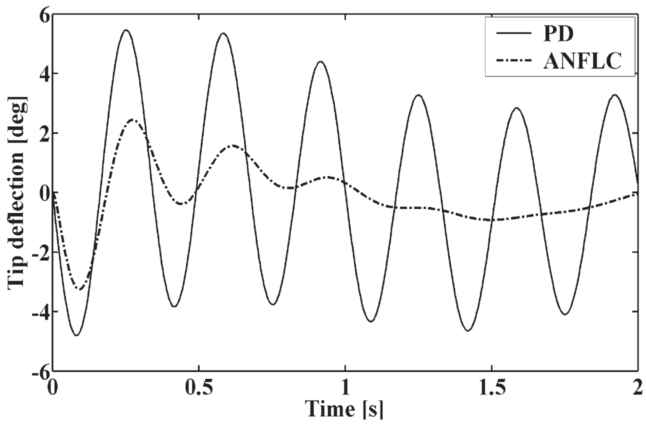


Fig. 21 Experimental results for the deflection control

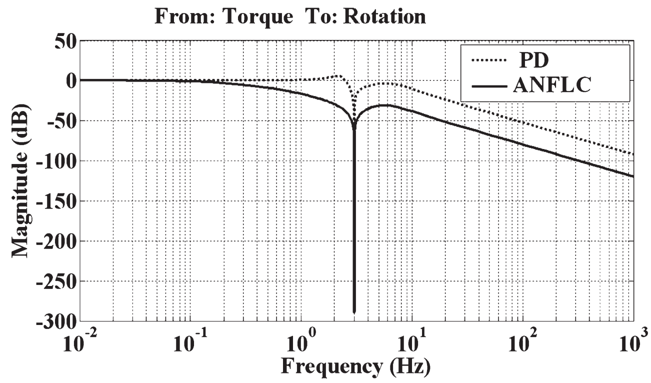


Fig. 22 Frequency responses for position control

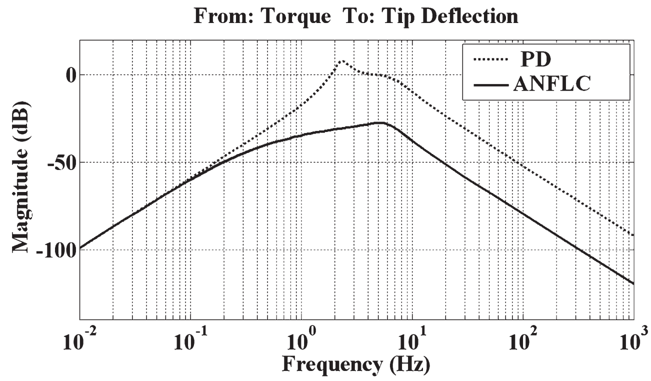


Fig. 23 Frequency responses for deflection control

simulation approach. Frequency responses of the controller for rotation and the tip deflection are given in Figs 22 and 23 respectively. It can be seen from the figures that the ANFLC increases the system damping; thus, increasing the isolation in the region of the system resonance, and producing higher isolation at higher frequencies, lower isolation at lower frequencies for both angular displacement and tip deflection compared to the PD controller. A comparison of the control voltages for both controllers is given in Fig. 24. The proposed controller requires a smaller voltage compared to the control voltage required by the PD controller. The proposed controller shows a smaller fluctuation in the control voltage compared to the PD controller.

In this study the desired input parameters were used for the ANFLC, however, the PD gain parameters and architecture of ANFLC were not changed. Rotation and tip deflection results for the PD

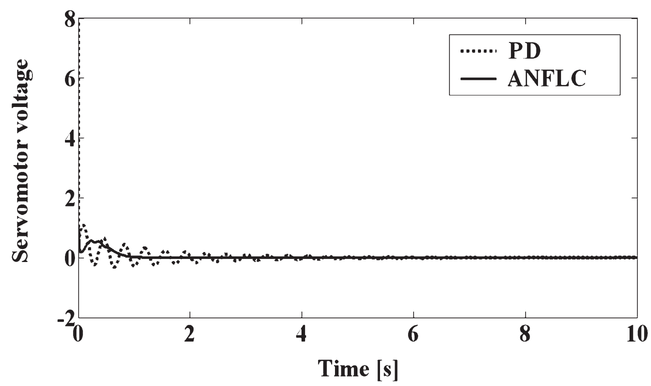


Fig. 24 Servo motor control voltages of the controllers

controller and ANFLC are given Figs 25 to 28. A block diagram of the controlled system with disturbance input is given in Fig. 29 and Fig. 30 shows the response of the ANFLC for rotation control with disturbance input. The disturbance input was chosen to be a sinusoidal wave (amplitude: 20, frequency: 1 rad/s) and it is applied 5s after the system begins to work As seen in Fig. 31, the ANFLC can give good results for different input conditions while the response of the PD controller is very poor. From

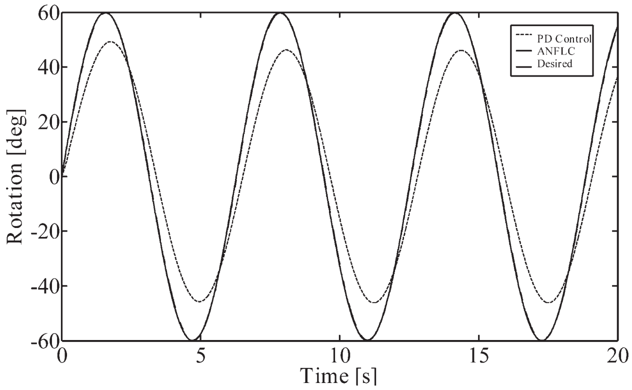


Fig. 25 Rotation control for sinusoidal input

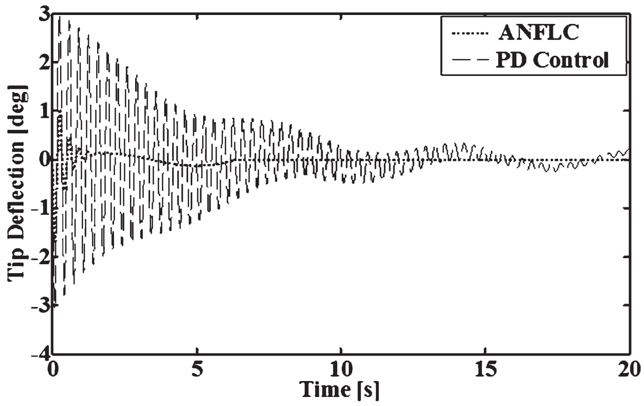


Fig. 26 Tip deflection control for sinusoidal input

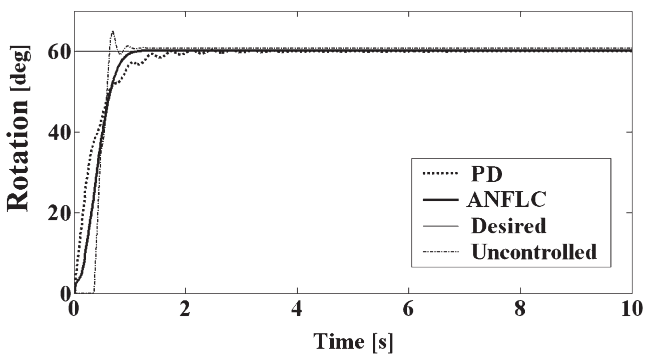


Fig. 27 Rotation control for step input

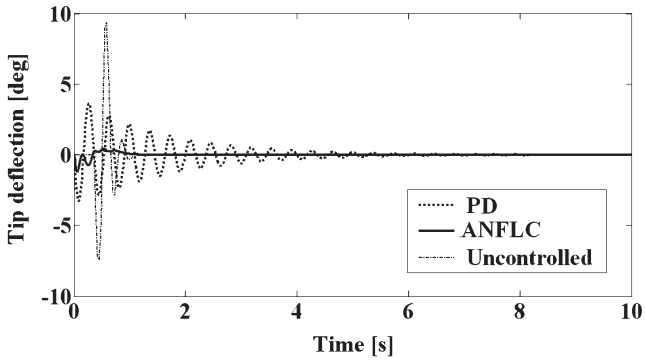


Fig. 28 Tip deflection control for step input

these results it is clear that the ANFLC is able to adapt to changing conditions such as different input signals (step, sinusoidal, etc.), disturbances, and non-linear system behaviour.

6 CONCLUSIONS

In this paper, dynamic modelling and position control of a single-flexible-link manipulator is studied using both simulation and experimental approaches. Initially, the dynamic model of the single flexible link is created in CAD software using experimentally determined parameter values. This model is then imported into MATLAB software and the Simulink/SimMechanics toolbox is used to implement the controller. The objective of this study is to move the tip of the flexible manipulator to a desired position and eliminate vibrations of the arm while it moves by using an ANFLC. The vibration control of the tip point is realized successfully while the end-effector traces the desired position signal. Experimental and simulation results are presented and composed. It is shown that the proposed controller has a good performance with the results obtained in the CAD modelling and in experiments being in close agreement. Thus, it is clear that the proposed approach can be used to study this kind of complex dynamic systems without the need to derive complex mathematical equations.

ACKNOWLEDGEMENT

This work is supported by the Coordinatorship of Selçuk University's Scientific Research Projects (Project no.: 06401020).

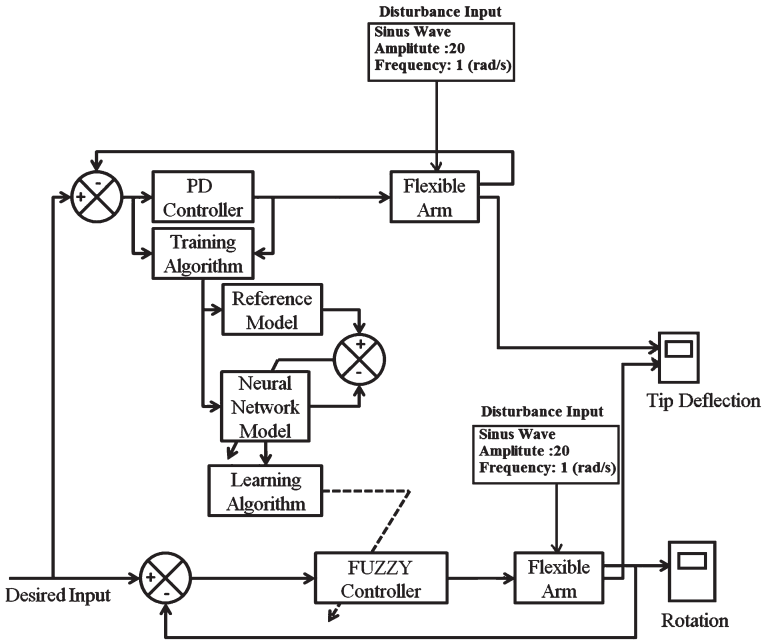


Fig. 29 Block diagram of the controlled system with disturbance

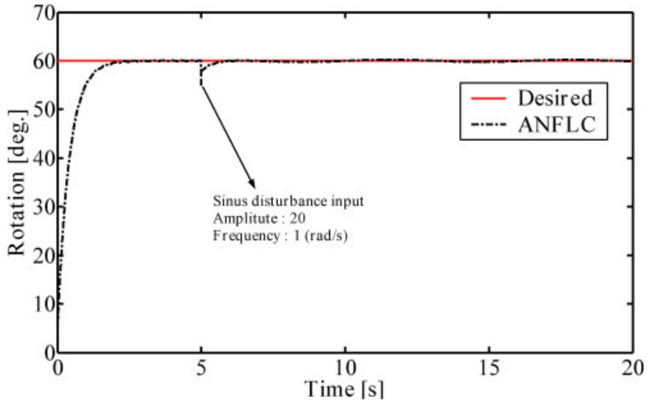


Fig. 30 Rotation control of the arm under disturbance for a step input

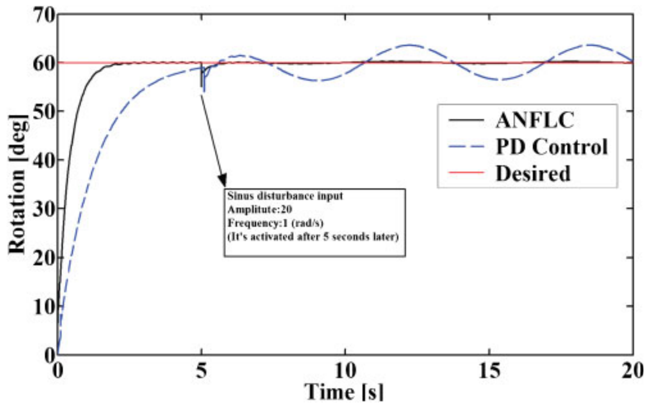


Fig. 31 Comparison of the controllers under disturbance for a step input

REFERENCES

1 **Kalyoncu, M.** and **Botsalı, F. M.** Effect of axial shortening on the vibration of elastic robot arm. In Proceedings of the Second Turkish–German Joint International Symposium on *Computer applications*, Konya, Turkey, 1998, pp. 313–324.

2 **Shi, Z. X., Fung, E. H. K.,** and **Li, Y. C.** Dynamic modeling of a rigid-flexible manipulator for constrained motion task control. *Appl. Math. Model.*, 1993, **23**, 509–525.

3 **Knani, J.** Dynamic modeling of flexible robotic mechanisms and adaptive robust control of trajectory computer simulation–part I. *Appl. Math. Model.*, 2002, **26**, 1113–1124.

4 **Ankaralı, A., Kalyoncu, M., Botsalı, F. M.,** and **Şişman, T.** Mathematical modeling and simulation of a flexible shaft-flexible link system with end mass. *Math. Comput. Model. Dyn. Sys.*, 2004, **10**(3), 187–200.

5 **Lin, J.** and **Lewis, F. L.** A symbolic formulation of dynamic equations for a manipulator with rigid and flexible links. *Int. J. Robot. Res.*, 1994, **13**(5), 454–466.

6 **Kalyoncu, M.** and **Botsalı, F. M.** Dynamic modeling of a robot manipulator with a translating and rotating elastic arm (in Turkish). In Proceedings of the Third International Workshop on *Mechatronics design and modeling*, Ankara, Turkey, 1997, pp. 237–247.

7 **Kalyoncu, M.** and **Botsalı, F. M.** Vibration analysis of an elastic robot manipulator with prismatic joint and a time varying end mass. *Arab. J. Sci. Engng*, 2004, **29**(1C), 27–38.

8 **Dwivedy, S. K.** and **Eberhard, P.** Dynamic analysis of flexible manipulators, a literature review. *Mech. Mach. Theory*, 2006, **41**, 749–777.

9 **Kalyoncu, M.** Mathematical modeling and dynamic response of a multi-straight-line path tracing flexible robot manipulator with rotating-prismatic joint. *Appl. Math. Model.*, 2007, **32**(6), 1087–1098.

10 **Monje, C. A., Ramos, F., Feliu, V.,** and **Vinagre, B. M.** Tip position control of a lightweight flexible manipulator using a fractional order controller. *IET Control Theory Appl.*, 2007, **1**(5), 1451–1460.

11 **Sanz, A.** and **Etxebarria, V.** Experimental control of a single-link flexible robot arm using energy shaping. *Int. J. Syst. Sci.*, 2007, **38**(1), 61–71.

12 **Subudhi, B.** and **Morris, A. S.** Fuzzy and neuro-fuzzy approaches to control a flexible single-link manipulator. *Proc. IMechE, Part I: J. Systems and Control Engineering*, 2003, **217**(I5), 387–400.

13 **Arteaga, M. A.** and **Siciliano, B.** On tracking control of flexible robot arms. *IEEE Trans. Autom. Control*, 2000, **45**(3), 520–527.

14 **Dul, I.** and **Karcz–Dul, I.** Algorithms of trajectory planning with constrained deviation from a given end-effector path. *Robotica*, 2004, **22**, 633–642.

15 **Ata, A. A.** and **Myo, T. R.** Optimal point-to-point trajectory tracking of redundant manipulators using generalized pattern search. *Int. J. Adva. Robot. Sys.*, 2005, **2**(3), 239–244.

16 **Burkan, R.** and **Uzmay, İ.** A model of parameter adaptive law with time varying function for robot control. *Appl. Math. Model.*, 2005, **29**, 361–371.

17 **Huang, P., Xu, Y.,** and **Liang, B.** Tracking trajectory planning of space manipulator for capturing operation. *Int. J. Adv. Robot. Sys.*, 2006, **3**(3), 211–218.

18 **Owen, W. S., Croft, E. A.,** and **Benhabib, B.** Real-time trajectory resolution for a two-manipulator machining system. *J. Robot. Syst.*, 2006, **22**, S51–S63.

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

20 **Akbarzadeh-Totonchi, M. R., Jamshidi, M.,** and **Vadiee, N.** Hierarchical fuzzy controller using line-curvature feature extraction for a single link flexible arm. In Proceedings of the Third IEEE International Conference on *Fuzzy systems*, Orlando, Florida, 26–29 June 1994, vol. 1, pp. 524–529 (IEEE, Piscataway, New Jersey).

21 **Lin, J.** Flexible link robot arm control by a hierarchical fuzzy logic approach. In Proceedings of the IEEE International Conference on *Plasma science*, Honolulu, Hawaii, 12–17 May 2002, vol. 2, pp. 1138–1143 (IEEE, Piscataway, New Jersey).

22 **Caswara, F. M.** and **Unbehauen, H.** A neurofuzzy approach to the control of a flexible-link manipulator. *IEEE Trans. Robot. Autom.*, 2002, **18**(6), 932–944.

23 **Kuo, K. Y.** and **Lin, J.** Fuzzy logic control for flexible link robot arm by singular perturbation approach. *Appl. Soft Comput.*, 2002, **2**, 24–38.

24 **Lin, J.** Hierarchical fuzzy logic controller for a flexible link robot arm performing constrained motion tasks. *IEE Proc. D, Control Theory Appl.*, 2003, **150**(4), 355–364.

25 **Chiang, T. S.** and **Lin, J.** A new design of hierarchical fuzzy hybrid position/force control for flexible link robot arm. In Proceedings of the *American control conference*, Denver, Colorado, 4–6 June 2003, vol. 6, pp. 5239–5244.

26 **Lin, J.** and **Lewis, F. L.** Two-time scale fuzzy logic controller of flexible link robot arm. *Fuzzy Sets Syst.*, 2003, **139**(1), 125–149.

27 **Emara, H.** and **Elshafei, A. L.** Robust robot control enhanced by a hierarchical adaptive fuzzy algorithm. *Engng Appl. Artif. Intell.*, 2004, **17**, 187–198.

28 **Lin, J.** A vibration absorber of smart structures using adaptive networks in hierarchical fuzzy control. *J. Sound Vib.*, 2005, **287**(4/5), 683–705.

29 **Jnifene, A.** and **Andrews, W.** Experimental study on active vibration control of a single-link flexible manipulator using tools of fuzzy logic and neural networks. *IEEE Trans. Instrum. Meas.*, 2005, **54**(3), 1200–1208.

30 **Kamalasadan, S.** and **Ghandakly, A. A.** A neural network parallel adaptive controller for dynamic system control. *IEEE Trans. Instrum. Meas.*, 2007, **56**(5), 1786–1796.

31 **Karimi, H. R.** and **Yazdanpanah, M. J.** A new modeling approach to single-link flexible manipulator using singular perturbation method. *Electr. Engng*, 2006, **88**(5), 375–382.

32 **Karimi, H. R., Yazdanpanah, M. J., Patel, R. V.,** and **Khorasani, K.** Modeling and control of linear two-time scale systems: applied to single-link flexible manipulator. *J. Intell. Robot. Syst.: Theory Appl.*, 2006, **45**(3), 235–265.

33 **Lin, J., Huang, Z. Z.,** and **Huang, P. H.** An active damping control of robot manipulators with oscillatory bases by singular perturbation approach. *J. Sound Vib.*, 2007, **304**, 345–360.

34 **Lin, J.** and **Huang, Z. Z.** A hierarchical fuzzy approach to supervisory control of robot manipulators with oscillatory bases. *Mechatronics*, 2007, **17**(10), 589–600.

35 **Jang, J.-S. R.** ANFIS: adaptive-network-based fuzzy inference system. *IEEE Trans. Sys., Man Cybern.*, 1993, **23**(3), 665–685.

36 **Zhang, L., Cao, Q.,** and **Lee, J.** A hybrid one-by-one learning with incremental learning algorithm in CMAC. In Proceedings of the Fifth World Congress on *Intelligent control and automation*, Hangzhou. People's Republic of China, 15–19 June, 2004, pp. 2415–2418.

37 **Sha, D.** and **Bajic, V. B.** An on-line hybrid learning algorithm for multilayer perceptron in identification problems. *Comput. Electr. Engng*, 2002, **28**, 587–598.

38 **Denai, M. A., Palis, F.,** and **Zeghbib, A.** ANFIS based modeling and control of nonlinear systems: a tutorial. IEEE International Conference on Systems, Man and Cybernetics, 2004.

39 **Siddique, M. N. H.** and **Tokhi, M. O.** GA-based neural fuzzy control of flexible-link manipulators. *Engng Lett.*, 2006, **13**(2), 2–13.

40 **Rezaeeian, A., Yousefi-Koma, A., Shasti, B.,** and **Doosthoseini, A.** ANFIS modeling and feedforward control of shape memory alloy actuators. *Int. J. Math. Mod. Meth. Appl. Sci.*, 2008, **2**(2), 228–235.
41 **Chopra, S., Mitra, R.,** and **Kumar, V.** A neurofuzzy learning and its application to control system. *Int. J. Comput. Intell.*, 2007, **3**(1), 73–79.
42 **Bodyanskiy, Y., Pliss, I.,** and **Vynokurova, O.** Adaptıve wavelet-neuro-fuzzy network in the forecastıng and emulatıon tasks. *Int. J. Inform. Theories Appl.*, 2008, **15**, 47–55.
43 **Kalaichelvi, V., Sivakumar, D., Karthikeyan, R.,** and **Palanikumar, K.** Prediction of the flow stress of 6061 Al–15% SiC – MMC composites using adaptive network based fuzzy inference system. *Mater. Des.*, 2009, **30**, 1362–1370.
44 **Denai, M. A., Palis, F.,** and **Zeghbib, A.** Modeling and control of non-linear systems using soft computing techniques. *Appl. Soft Comput.*, 2007, **7**, 728–738.
45 **Güler, İ.** and **Übeyli, E. D.** Automatic detection of ophthalmic arterystenosis using the adaptive neuro-fuzzyinference system. *Engng Appl. Artif. Intell.*, 2005, **18**, 413–422.
46 **Kim, D.** A design of CMAC-based fuzzy logic controller with fast learning and accurate approximation. *Fuzzy Sets Syst.*, 2002, **125**, 93–104.
47 **Widrow, B.** and **Stearns, D.** *Adaptive signal processing*, 1985 (Prentice-Hall, Englewood Cliffs, New Jersey).
48 **Aström, K. J.** and **Wittenmark, B.** *Computer controller systems: theory and design*, 1984 (Prentice Hall, New Jersey).
49 **Goodwin, G. C.** and **Sin, K. S.** *Adaptive filtering prediction and control*, 1984 (Prentice Hall, Englewood Cliffs, New Jersey).
50 **Ljung, L.** *System identification: theory for the user*, 1987 (Prentice Hall, Englewood Cliffs, New Jersey).
51 **Strobach, P.** *Linear prediction theory: a mathematical basis for adaptive systems*, 1990 (Springer-Verlag, New York).
52 **Narendra, K. S.** and **Parthsarathy, K.** Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Netw.*, 1990, **1**(1), 4–27.
53 **Werbos, P. J.** An overview of neural networks for control. *IEEE Control Syst. Mag.*, 1991, **11**(1), 40–41.
54 **Miller, W. T., Sutton, R. S.,** and **Werbos, P. J.** (Eds) *Neural networks for control*, 1990 (Massachusetts Institute of Technology Press, Cambridge, Massachusetts).
55 **Widrow, B.** and **Winter, R.** Neural nets for adaptive filtering and adaptive pattern recognition. *IEEE Comput.*, 1988, **21**(3), 25–39.
56 **Kosko, B.** *Neural networks for signal processing*, 1991 (Prentice Hall, Englewood Cliffs, New Jersey).
57 **Haykin, S. S.** *Adaptive filter theory,* second edition, 1991 (Prentice Hall, Englewood Cliffs, New Jersey).