Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 9 (7): 1258-1266, 2009
ISSN 1812-5654

Stable Direct Adaptive Control as Nonlinear Hybrid Controller for Flexible Manipulator

[1]M. Naeimi, [2]M. Teshnehlab, [2]M. Aliyari Sh and [3]M. Aliasghary
[1]Islamic Azad University, Arak Branch, Astaneh Campus, Astaneh, Iran
[2]K.N. Toosi University, Seyyed Khandan, Shariati Street, Tehran, Iran
[3]Istanbul Technical University, Istanbul, Turkey

Abstract: Stable direct adaptive control has been developed in this study to control the flexible motion of a single-link robotic manipulator. The controller has been designed based on a simplified model of the arm, which only accounts for the first elastic mode of the beam. The controller consists of three parts: linear feedback, a nonlinear sliding mode (SMC) and an adaptive Fuzzy-Neural Network (FNN) controller.

Key words: Flexible link, linear feedback, sliding mode, adaptive fuzzy-neural network

INTRODUCTION

Regarding the recent develops of technology in using robot and also industrial demand to the high speed and quality robots, the idea of using light robots is proposed. Because, a relative deformation happens in the manipulator of high speed robots with heavy load, therefore, the elastic robot is proposed. Thus, modeling and controlling of the robots with elastic links has its own problems. In addition dynamical structure of such robots includes nonlinear and uncertainty factors in model. So, designing an adaptive controller that is able to provide the performance characteristics such as tracing the reference input, eliminating the disturbance and the conditions of response speed.

In this study, in order to provide above parameters a hybrid controller is used. The controller consists of three parts: linear feedback (PD), a nonlinear Sliding Mode Controller (SMC) and an adaptive Fuzzy-Neural Network (FNN) controller. The total control signal is computed as follows: where is the linear feedback control, is the sliding mode control and is the adaptive neural control.

Talebi *et al*. (1998) presented designed simulation and experimental results on the performance of neural network-based controllers for tip position tracking of flexible-link manipulators. The controllers are designed by utilizing the modified output re-definition approach. The performance of the four proposed neural network controllers is illustrated by simulation results for a two-link planar flexible manipulator and by experimental results for a single flexible-link test-bed. Tian and Collins (2005) described an adaptive neuro-fuzzy control system for controlling a flexible manipulator with variable payload. The controller proposed in this paper is comprised of a Fuzzy Logic Controller (FLC) in the feedback configuration and two dynamic recurrent neural networks in the forward path. A dynamic Recurrent Identification Network (RIN) is used to identify the output of the manipulator system and a dynamic Recurrent Learning Network (RLN) is employed to learn the weighting factor of the fuzzy logic. Simulations for determining the number of modes to describe the dynamics of the system and investigating the robustness of the control system are carried out. Yuangang (2006) focused on tracking control problem of flexible-link manipulators. In order to alleviate the effects of nonlinearities and uncertainties, a combined control strategy based on Neural Network (NN) and the concept of Sliding Mode Control (SMC) is proposed systematically. Chalhoub *et al*. (2006) designed a robust controllers and a nonlinear observer for the control of a single-link flexible robotic manipulator. The controllers consist of a Conventional Sliding Mode Controller (CSMC) and a Fuzzy-Sliding Mode Controller (FSMC). Moreover, the robust nonlinear observer has been designed based on the sliding mode methodology. Kim and Inman (2001) used a CSMC to damp out the vibrations of a flexible cantilevered beam with piezoelectric actuators/sensors. No rigid-body motion is considered in this study.

MATHEMATICAL MODEL OF THE SYSTEM

The physical system consists of a flexible link connected to a revolute joint (Fig. 1). The beam is made of aluminum and has an annular cross section. It is restricted in its motion to the horizontal plane. The stiffness of the beam in the longitudinal direction is much higher than in flexure. Therefore only the in plane transverse deflection

Corresponding Author: Mohsen Naeimi, Islamic Azad University, Arak Branch, Astaneh Campus, Astaneh, Iran
Tel: +98- 862 4434862 Fax: +98- 862 4434862

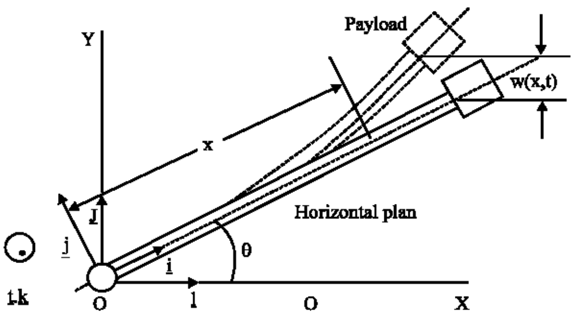

Fig. 1: Flexible link geometry and coordinates

of the beam, W(x,t) is considered in addition to its rigid-body motion. The payload consists of a lumped mass mounted at the free-end of the beam. The dynamic model retains all the coupling terms between the rigid and flexible motions of the beam. The position vector of an arbitrary point on the flexible link is given by:

$$\tilde{r}_B = x\tilde{i} + W(x,t)\tilde{j} \quad (1)$$

where, x is the time invariant since the longitudinal vibration is neglected (Fig. 1). The assumed modes method is implemented to approximate W(x,t), which is considered here to be dominated by the first two elastic modes.

It is written as a linear combination of admissible functions, Φ_i (x), of spatial coordinate and time-dependent generalized coordinates, $q_{2i}(t)$. The admissible functions are chosen to be the first two eigenfunctions of a clamped-free beam derived based on the Euler- Bernoulli beam assumption. Similarly, the position vector of the payload is determined by substituting x by L in Eq. 1. The velocity vector of an arbitrary point of the beam is given by:

$$T_t = \frac{1}{2}\int_{M_B} (\dot{\tilde{r}}_B.\dot{\tilde{r}}_B)dm + \frac{1}{2}m_p(\dot{\tilde{r}}_p.\dot{\tilde{r}}_p) \quad (2)$$

where, $\tilde{\omega}$ is equal to $\dot{\theta}_K$. The total kinetic energy of the system is written as:

$$T_t = \frac{1}{2}\int_{M_B} (\dot{\tilde{r}}_B.\dot{\tilde{r}}_B)dm + \frac{1}{2}m_p(\dot{\tilde{r}}_p.\dot{\tilde{r}}_p) \quad (3)$$

The strain energy stored in the system is expressed as:

$$E_t = \frac{1}{2}\int_0^L EI\left(\frac{\partial^2 W(x,t)}{\partial x^2}\right)^2 dx \quad (4)$$

The total virtual work, done on the system, is determined as follows:

$$\delta W_t = \tau_1\delta\theta + \frac{1}{2}\int_0^L P(\dot{\theta},x)\left(\frac{\partial W(x,t)}{\partial x}\right)^2 dx \quad (5)$$

where, τ_1 is the non-conservative generalized control torque applied at the base joint.

The second term reflects the stiffening effect of the beam induced by the centrifugal force $P(\dot{\theta},x)$, which can be expressed as:

$$P(\dot{\theta},x) = \frac{1}{2}\rho A\dot{\theta}^2 L^2(1 - \frac{x^2}{L^2}) + m_p L\dot{\theta}^2 \quad (6)$$

The variation of the inertial axial force, $P(\dot{\theta},x)$, due to the flexible motion is neglected in this formulation.

The equations governing the rigid and flexible motions of the beam are obtained by implementing the Lagrange principle. The resulting equations of motion are three highly nonlinear, coupled, stiff, second-order ordinary differential equations. These equations are then converted to a set of six first order ordinary differential equations that can be written as:

$$\begin{pmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\\ \dot{x}_3(t)\\ \dot{x}_4(t)\\ \dot{x}_5(t)\\ \dot{x}_6(t)\end{pmatrix} = \begin{pmatrix} f_1(\tilde{x}(t),\tau_1)\\ f_2(\tilde{x}(t),\tau_1)\\ f_3(\tilde{x}(t),\tau_1)\\ f_4(\tilde{x}(t),\tau_1)\\ f_5(\tilde{x}(t),\tau_1)\\ f_6(\tilde{x}(t),\tau_1\end{pmatrix} = \begin{pmatrix} x_4(t)\\ x_5(t)\\ x_6(t)\\ -M^{-1}(\tilde{x}(t))\begin{bmatrix}\bar{g}_1(\tilde{x}(t))+\tau_1\\ \bar{g}_2(\tilde{x}(t))\\ \bar{g}_3(\tilde{x}(t))\end{bmatrix}\end{pmatrix} \quad (7)$$

where, the state vector is defined to be

$$x(t) = \begin{bmatrix}\theta & q_{21} & q_{22} & \dot{\theta} & \dot{q}_{21} & \dot{q}_{22}\end{bmatrix}^T$$

The state equations are solved numerically by using the Gear's method, which is well suited for solving stiff systems. This model is used, in this study, as a test bed for assessing the combined performances of the controllers and the observer in the presence of both structured and unstructured uncertainties of the plant.

However, it should be emphasized that a simplified version of the model, obtained by ignoring the second elastic mode of the beam, has been used herein in the design of the controllers and the observer. Its equations can be expressed as:

$$\dot{x}_r = \begin{pmatrix} f_{r1}(\tilde{x}_r,\tau_1)\\ f_{r2}(\tilde{x}_r,\tau_1)\\ f_{r3}(\tilde{x}_r,\tau_1)\\ f_{r4}(\tilde{x}_r,\tau_1)\end{pmatrix} = \begin{pmatrix} x_{r3}\\ x_{r4}\\ -M_r^{-1}(\tilde{x}_r)\begin{bmatrix}\bar{g}_{r1}(\tilde{x}_r)+\tau_1\\ \bar{g}_{r2}(\tilde{x}_r)\end{bmatrix}\end{pmatrix} = \begin{pmatrix} x_{r3}\\ x_{r4}\\ g_{r1}(\tilde{x}_r)+b_1(\tilde{x}_r)\tau_1\\ g_{r2}(\tilde{x}_r)+b_2(\tilde{x}_r)\tau_1\end{pmatrix} \quad (8)$$

Where:

$$\tilde{x}_r = \left[\theta \quad q_{21} \quad \dot{\theta} \quad \dot{q}_{21}\right]^T$$

$M_r(\tilde{x}_r)$, $\bar{g}_{r1}(\tilde{x}_r)$ and $\bar{g}_{r2}(\tilde{x}_r)$ are obtained from $M(\tilde{x})$, $\tilde{g}_1(\tilde{x})$ and $\tilde{g}_2(\tilde{x})$ in Eq. 7 by deleting the entries and terms associated with $q_{22}(t)$ and its time derivative.

The expressions for f_{r3} and f_{r4} are:

$$\ddot{\theta} = f_{r3}(\tilde{x}_r, \tau_1) = g_{r1}(\tilde{x}_r) + b_1(\tilde{x}_r)\tau_1 = -\frac{x_{r2}(661.69x_{r3}x_{r4} - 66.58x_{r3}^2 - 772189)}{330.85x_{r2}^2 + 5.86} + \frac{18.19\tau_1}{330.85x_{r2}^2 + 5.86} \quad (9)$$

$$\ddot{q}_{21} = f_{r4}(\tilde{x}_r, \tau_1) = g_{r2}(\tilde{x}_r) + b_2(\tilde{x}_r)\tau_1 = -\frac{x_{r2}(930210 - 787.26x_{r3}x_{r4} + 80.53x_{r3}^2 + 649027x_{r3}^2 + 56.19x_{r2}^2x_{r3}^2}{330.85x_{r2}^2 + 5.86} - \frac{21.64\tau_1}{330.85x_{r2}^2 + 5.86} \quad (10)$$

Both controllers (FNN and SMC) are designed based on the following θ equation.

MATERIALS AND METHODS

We have studied linear feedback, a nonlinear sliding mode controller and adaptive neural network controllers which are considered in following parts.

Stable direct adaptive control: There have been several recent direct adaptive control techniques which have been designed to guarantee overall system stability. The method of uses Lyapunov stability theory in the design of the network learning rule, rather than a gradient descent algorithm like back propagation. The controller consists of three parts: linear feedback, a nonlinear sliding mode controller and an adaptive neural network controller (Fig. 3). The total control signal is computed as follows: where is the linear feedback control, is the sliding mode control and is the adaptive neural control. The function allows a smooth transition between the sliding and adaptive controllers, based on the location of the system state:

$$u(t) = u_{pd}(t) + (1 - m(t))u_{ad} + m(t)u_{sl}(t) \quad (11)$$

where, $u_{pd}(t)$ is the linear feedback control, $u_{sl}(t)$ is the sliding mode control and $u_{ad}(t)$ is the adaptive neural control. The function m(t) allows a smooth transition between the sliding and adaptive controllers, based on the location of the system state:

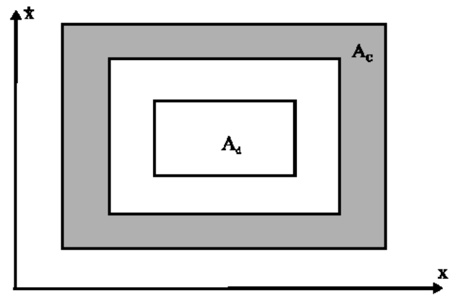

Fig. 2: Controller regions

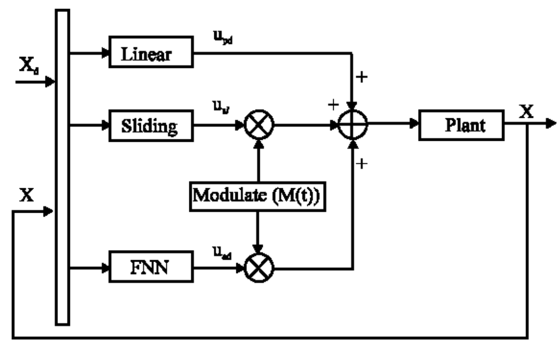

Fig. 3: Stable direct adaptive control

$$\begin{cases} m(t) = 0 & x(t) \in A_d \\ 0 < m(t) < 1 & \text{otherwise} \\ m(t) = 1 & x(t) \in A_c \end{cases} \quad (12)$$

where, the regions might be defined as in Fig. 2.

The sliding mode controller is used to keep the system state in a region where the neural network can be accurately trained to achieve optimal control. The sliding mode controller is turned on (and the neural controller is turned off) whenever the system drifts outside this region. The combination of controllers produces a stable system which adapts to optimize performance.

It should be noted that this neural controller uses the radial basis neural network. The radial basis output is a linear function of the network weights, which allows faster training and simpler analysis than is possible with multilayer networks. It has the disadvantage that it may require many neurons if the number of network inputs is large. It also requires that the centers and spread of the basis functions be selected before training.

Design of controller: It is discribed earlier that, the controller has four parts and we will design each of them separately. The resulted equations in the previous part for the robot with flexible link are all nonlinear in which indefinite terms are observed as well.

The inaccuracy in the model may be due to uncertainty in basic plan or because of ignoring dynamic parameters such as friction, coriolis acceleration etc. Therefore, designing a control with some robust parts seems inevitable. A SMC controller can be a good choice to protect against the above-mentioned conditions. In addition, a SMC can guarantee system stability because Lyapunov criterion. However, a linear controller (PD) is suitable for the system transient respond. Such parameters of rise time and overshoot can be improved by setting coefficients K_P, K_D. Using a fuzzy-neural controller, we consider the issue of the control's adaptation with plant. The adaptive controller FNN improves the system through minimizing the tracking error of the steady state response.

Design of nonlinear sliding mode controller: In order to design nonlinear sliding mode control (Tao *et al*., 2004), we should look at Eq. 9 and 10. In these equations, we choose the θ link angle with the horizon as the output of the system, whereas the applied torque to the joint τ_1 is considered as the input of the system, so that the outcome system is an SISO.

All controllers are designed based on the following τ equation:

$$\ddot{\theta} = f_{r3}(\tilde{x}_r, \tau_1) = g_{r1}(\tilde{x}_r) + b_1(\tilde{x}_r)\tau_1 \quad (13)$$

where, the term $b_1(\tilde{x}_r)$ is considered to be fully known. However, $g_{r1}(\tilde{x}_r)$ is treated as an unknown term. It has been approximated by the following nominal function $\hat{g}_{r1}$:

$$\hat{g}_{r1}(\hat{x}_r) = -\frac{x_{r2}(700x_{r3}x_{r4} - 70x_{r3}^2 - 770{,}000)}{300x_{r2}^2 + 6} \quad (14)$$

Only the upper bound of the model imprecision is assumed to be known. It is defined as:

$$|\Delta f_{r3}| = |f_{r3} - \hat{f}_{r3}| = |g_{r1} - \hat{g}_{r1}| \le F_3 \quad (15)$$

Since the task of the controller is to force θ to track the desired angular displacement θ_d, then the tracking error is defined to be:

$$e_1 = x_{r1} - x_{r1d} = \theta - \theta_d \quad (16)$$

Accordingly, the sliding surface is expressed as:

$$s(\tilde{e},t) = \dot{e}_1 + \lambda e_1 \quad (17)$$

Based on the nominal function $\hat{g}_{r1}$ of the system, the continuous control law τ_{1eq}, satisfying $\dot{s}(\tilde{e},t) = 0$, is expressed as:

$$\tau_{1eq} = b_1^{-1}\{-\hat{g}_{r1} + \ddot{x}_{1d} - \lambda\dot{e}_1\} \quad (18)$$

Once on the surface, the dynamic response of the system is governed by:

$$\left(\frac{d}{dt} + \lambda\right)e_1 = 0 \quad (19)$$

The tracking error will be driven to zero by selecting λ to be a strictly positive constant. To force the system trajectory to converge to the sliding surface in the presence of both model uncertainties and disturbances, the feedback control torque τ_1 is defined as:

$$\tau_1 = \tau_{1eq} - b_1^{-1}k\,\mathrm{sgn}(s) \quad (20)$$

where, k is determined by satisfying the following sliding condition:

$$\frac{d}{dt}V_1 = \frac{1}{2}\frac{d}{dt}s^2(\tilde{e},t) \le -\eta|s| \quad (21)$$

$V_1 = 1/2S^2$ is a positive definite function. It represents the squared distance between the sliding surface and any representative point of the system. The selection of η to be strictly positive will ensure that $\dot{V}_1$ is negative definite. Therefore, V_1 becomes a Lyapunov function that decreases along all trajectories of the system; thus, causing the sliding surface to become an invariant set. It can be easily proven that the above inequality is satisfied by selecting k to be:

$$k \ge \eta + F_3 \quad (22)$$

To alleviate the chattering problem induced by the switching term in the control signal, the sgn(s) term in Eq. 20 is often replaced by a saturation function as follows:

$$\tau_1 = \tau_{1eq} - b_1^{-1}k\,\mathrm{sgn}\left(\frac{s}{\Phi}\right) \quad (23)$$

where, Φ is the thickness of the boundary layer. It is considered herein to be time-variant.

Therefore, to ensure convergence of the system trajectory to the boundary layer, the sliding condition in Eq. 21 had to be modified to the following form:

$$\frac{1}{2}\frac{d}{dt}s^2(\tilde{e},t) \le (\dot{\Phi} - \eta)|s| \qquad (24)$$

The above condition can be satisfied by changing the expression of τ_1 as follows:

$$\tau_1 = \tau_{1eq} - b_1^{-1}\bar{k}\,\mathrm{sat}\left(\frac{s}{\Phi}\right) \qquad (25)$$

where, $\bar{k} = k - \dot{\Phi}$. The differential equation governing the behavior of Φ. It is given by:

$$\dot{\Phi} + \lambda\Phi = k(\theta_d) \qquad (26)$$

where, $k(\theta_d)$ is defined in Eq. 22.

Up to this stage, the formulation has only dealt with the design of the SMC.

Design of linear controller: To produce the control signal u_{pd} (linear control output), we consider a simple controller with the following transformation function:

$$u_{pd} = k_p e(t) + k_d \dot{e}(t) \qquad (27)$$

where, e is the system error which is gained via $e = \theta - \theta_d$. $\dot{e}$ is derivative of the system error.

Different methods are suggested for PID controller designing. All these methods, notice the process of choosing the control parameters in order to provide the desired operation characteristics. The method that we have used here is Ziegler-Nichols approximate method. Based on characteristics of transient response of the system under control, Ziegler-Nichols has proposed rules for determining proportional gain K_p and derivation time T_d.

Design of fuzzy neural network controller: Here, A feed forward four layers fuzzy neural network is constructed that is presented by Lee and Teng (2000). To implement the fuzzy control rules stated in Eq. 28 first layer accepts input variables. It nodes represent input linguistic variables second layer is used to calculate gaussian membership values nodes in this layer represent the terms of the receptive linguistic variables. Nodes at third layer represent fuzzy rules.

The links between third layer and fourth layer are connected by the weighing values w_p^i.

For a multi-input single output FNN system, let x be the input linguistic variable and α_j as the firing strength of rule j, which is obtained by the product of the grades of the membership function $\mu_{A_{ij}}(x)$ in the antecedent. If w_j represents the j-th consequence link weight, the inferred value y, is then obtained by taking the weighted sum of its input.

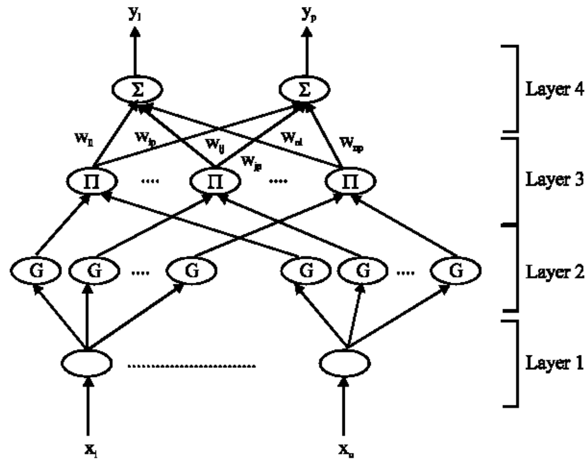

Fig. 4: The configuration of the proposed FNN

The proposed FNN (Fig. 4) realized the following fuzzy control rules:

$$R^j: \text{IF } u_{1j} \text{ is } A_{1j},\ldots,u_{nj} \text{ is } A_{nj}, \text{ then } y = w_j \qquad (28)$$

where, for i = 1,…, n, $u_{ij} = x_i$, A_{1j}, A_{nj} are fuzzy sets, w_j is a fuzzy singleton and n is the number of inputs.

Finally, the output of FNN is obtained:

$$y^* = \sum_{j=1}^{m} \alpha_j w_j \qquad (29)$$

Where:

$$\alpha_j = \prod_{i=1}^{n} \mu_{A_{ij}}(u_{ij})$$

Layered operation of the FNN: Next we shall indicate the signal propagation and operation functions of the nodes in each layer. In the following description U_i^k denotes the i-th input of a node in the k-th layer, O_j^k denotes the i-th node output in layer k.

- **Layer 1: Input layer:** The nodes in this layer only transmit input valve to the next layer directly:

$$o_i^1 = u_i^1 \qquad (30)$$

From this equation, the link weight at first layer w_i^1 is unity:

- **Layer 2: Membership layer:** In this layer, each node performs a membership function and acts as a unit of memory. The Gaussian function is adopted here as a membership function, thus we have:

$$o_{ij}^2 = \exp\left(-\frac{(u_{ij}^2 - m_{ij})}{(\sigma_{ij})^2}\right) \quad (31)$$

where, m_{ij} and σ_{ij} are the center (or mean) and width (or standard deviation-STD) of the Gaussian membership function. The subscript ij indicates the j-th term of the i-th input x_i

- **Layer 3: Rules layer:** The nodes in this layer one called rate nodes, the following AND operation is applied to each rule node to integrate these fan-in values:

$$o_i^3 = \prod_i u_i^3 = \exp\left\{-\left[D_i(u_i^2 - m_i)\right]^T\left[D_i(u_i^2 - m_i)\right]\right\} \quad (32)$$

Where:
D_i = Diag $[1/\sigma_{li}, 1/\sigma_{2i},...,1/\sigma_{ni}]$
u_i = $[u_{li}, u_{2i},...,u_{ni}]^T$
m_i = $[m_{li}, m_{2i},...,m_{ni}]^T$

The output o_i^3 of a rule node represents the "firing strength" of its corresponding rule.

- **Layer 4: Output layer:** Each node in this layer is called an output linguistic node. This layer performances the defuzzification operation. The node output is a linear combination of consequences obtained from each rule, That is:

$$y_i = o_j^4 = \sum_i^m u_{ij}^4 w_i^4 \quad (33)$$

Where, $u_i^4 = o_i^3$ and w_{ij}^4 (the link weight) is the output action strength of the j-th output associated with the i-th rule. The w_{ij}^4 are the tuning factors of this layer.

Finally, the overall representation of input x and the m-th output y is:

$$y_m(k) = o_m^4(k) = \sum_{j=1}^m w_{mj} \prod_{i=1}^n \exp\left[-\frac{\left[x_i(k) - m_{ij}\right]^2}{(\sigma_{ij})^2}\right] \quad (34)$$

where, m_{ij}, σ_{ij} and w_{mj} are tuning parameters.

Learning algorithm: Consider the single Output case for simplicity. Present goal is to minimize the following cost function:

$$E(k) = \frac{1}{2}(y(k) - \hat{y}(k))^2 = \frac{1}{2}\sum_j (y(k) - o^4(k))^2 \quad (35)$$

where, y(k) is the desired output and $\hat{y}(k) = O^4(k)$ is the current output for each discrete time k. In each training cycle starting at the input nodes in the current output $\hat{y}(k)$.

By using BP learning algorithm, the weighing vector of the FNN is adjusted such that the error defined in Eq. 35 is less than a designed threshold value after a given number of training cycles. The well-known algorithm may be written briefly as:

$$W(k+1) = W(k) + \Delta W(k) = W(k) + \eta\left(-\frac{\partial E(k)}{\partial W}\right) \quad (36)$$

where, in this case η and w represent the learning rate and tuning parameters of the FNN. Let $e(k) = y(k) - \hat{y}(k)$ and $w = [m, \sigma, w]^T$ be the training error and weighting vector of the FNN, then the gradient of error E(.) in Eq. 35 with respect to an arbitrary weighting vector w is:

$$\frac{\partial E(k)}{\partial W} = -e(k)\frac{\partial \hat{y}(k)}{\partial W} = -e(k)\frac{\partial o^4(k)}{\partial W} \quad (37)$$

By recursive application of the chain rule, the error term for each layer is first calculated, the parameters in the corresponding layers are adjusted, with the FNN Eq. 34 and cost function defined in Eq. 35, derive the update rule of w_{ij}:

$$w_{ij}(k+1) = w_{ij}(k) - \eta^w \frac{\partial E(k)}{\partial w_{ij}} \quad (38)$$

Where:

$$\frac{\partial E(k)}{\partial w_{ij}} = -e(k).o_i^3$$

Similarly, the update laws of m_{ij}, σ_{ij} are

$$m_{ij}(k+1) = m_{ij}(k) - \eta^m \frac{\partial E(k)}{\partial m_{ij}} \quad (39)$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \eta^\sigma \frac{\partial E(k)}{\partial \sigma_{ij}} \quad (40)$$

Where:

$$\frac{\partial E(k)}{\partial m_{ij}} = -\sum_k e(k) w_{ik}.o_k^3.\frac{2(x_i - m_{ij})^2}{(\sigma_{ij})^2} \quad (41)$$

$$\frac{\partial E(k)}{\partial \sigma_{ij}} = -\sum_k e(k) w_{ik}.o_k^3.\frac{2(x_i - m_{ij})^2}{(\sigma_{ij})^3} \quad (42)$$

The BP algorithm is a widely used algorithm for training multilayer network by means of error propagation via variation calculus. But its success depends upon the quality of the training data.

Designing the modulator: In order to design a control switch capable of designating the appropriate control for the plan according to control areas of signal, we have tried to use a smooth transition. In the simplest form of the switch, a sigmoid function was used (Fig. 5) in which the average and variance of the function are related to system error. That is, the system error determines the output coefficient by which the function is to switch the two adaptive and sliding mode controls.

RESULTS AND DISCUSSION

At this stage, the equations are simulated with numerical values of the system listed: Cross sectional area of the beam (A) is equal 7.2839×10^{-4} m^2, Outer radius of the beam (Ro) 0.0381 m , Inner radius of the beam (Ri) 0.0349 m, Length of the beam (L) 2.3 m, Mass of the beam (MB) 4.535 kg, Payload mass (mp) 3.405 kg and Density of aluminum (ρ) 270 kg m^{-3}.

The deigned controller is applied the plant and the results, are then analyzed.

In simulating, the input torque is considered as step input and calculated the step response of system. Figure 6 shows the system output. The system error tends toward zero and the response is with a desirable rise time

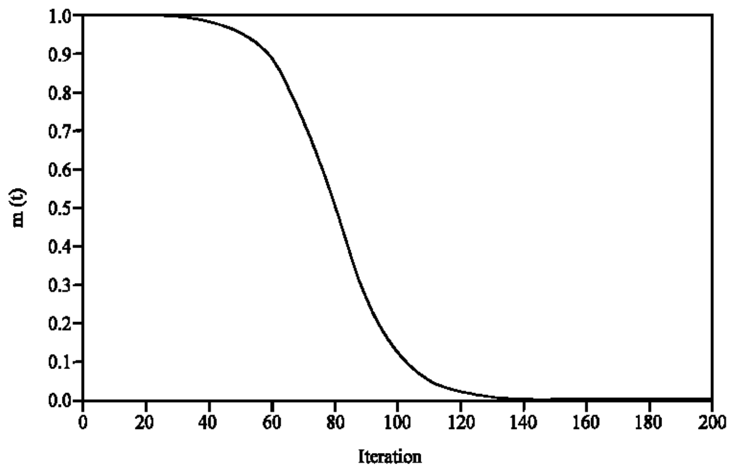

Fig. 5: Modulator switch

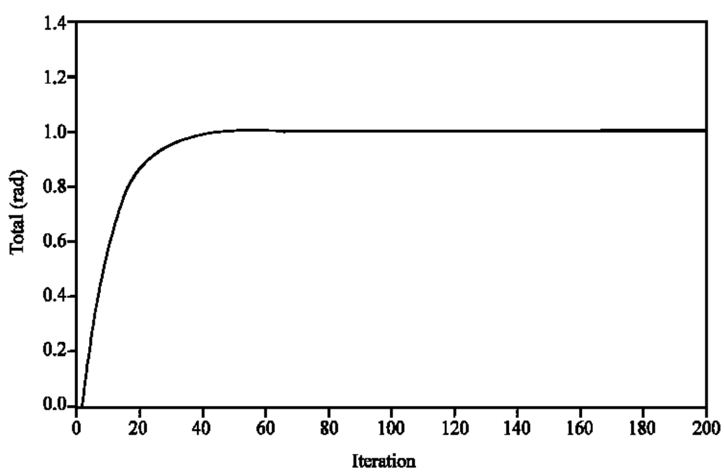

Fig. 6: Step response

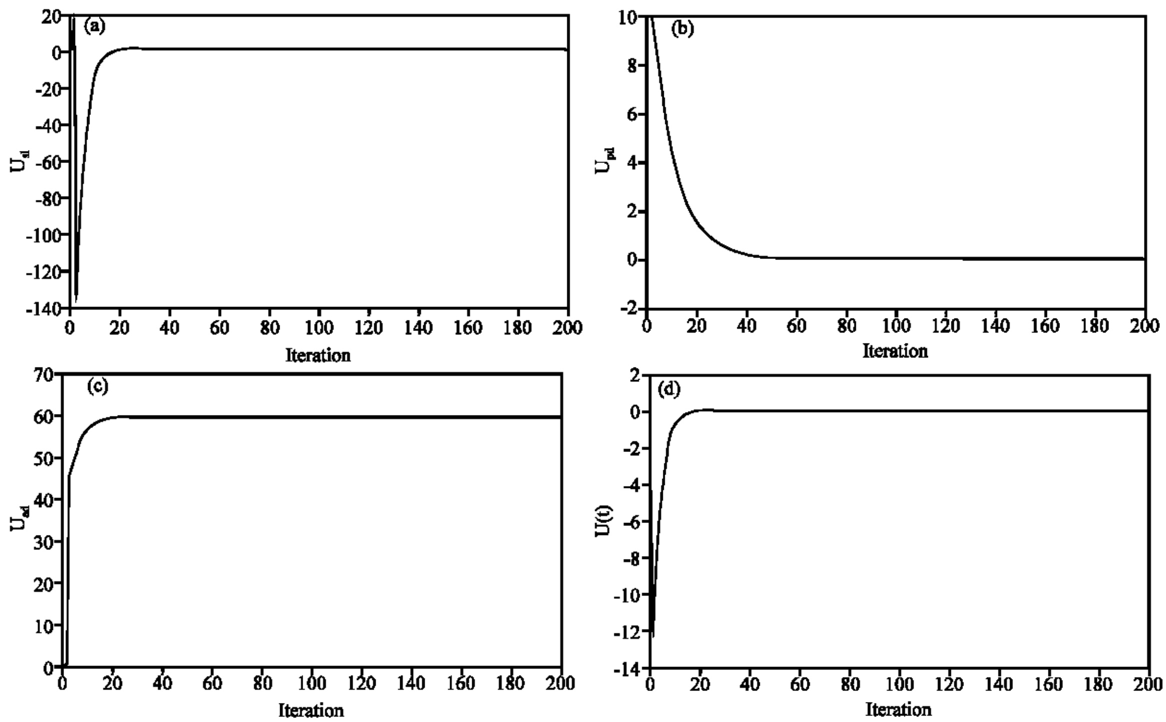

Fig. 7: Outputs of controllers, (a) SMC output, (b) linear controller, (c) Adaptive Controller and (d) Sum of output

and without overshoot. It is observed that from an iteration of about 80, that controller has switched to adaptive, system has suitable output and the tracking error is approximately zero. In Fig. 7a-d the control signal has also been looked at.

A control signal of SMC with a range of about 20 to -130 Nm without severe vibration has resulted. The control signal of PD controller with the range of 0 to 10 Nm at iteration 40 has become zero because the system error has also become zero at this iteration. FNN does not have a role in producing the total control signal applied to the plan before the iteration of 40. However, when the parameters of FNN network are formed based on error, it produces the required control signal and takes the control of the system.

CONCLUSION

In this study, a kind of nonlinear hybrid controller as the direct adaptive mode for controlling a robot with a flexible link was designed and simulated. There were indefinite and nonlinear terms in the dynamic equations of the robot. Nevertheless, with its robustness, the controller could well face these parameters. Through minimizing the tracking error, the control also showed an adaptive quality. Many studies have been carried out regarding the control of elastic-link robots; while few of them have pursued compound works. By choosing the above compound method, we tried to resolve the inefficiencies of conventional controllers. Comparing the results with Chalhoub's research (2006), one can see that the system response is more desirable. The applied range of torque to the plan is between -14 to 5 Nm, while in this reference with the given equations, it even reaches 305 Nm. We managed to get the steady stat error of system close to zero. That is to say, the control has eliminated all the vibrations of the robot's end effecter. The control signal applied to the plan has also the minimum vibrations; therefore, it can be easily produced and applied to the robot in the laboratory.

REFERENCES

Chalhoub, N.G., G.A. Kfoury and B.A. Bazzi, 2006. Design of robust controllers and a nonlinear observer for the control of a single-link flexible robotic manipulator. J. Sound Vibration, 291: 437-461.

Kim, M.H. and D.J. Inman, 2001. Reduction of observation spillover in vibration suppression using a sliding mode observer. J. Vibration Control, 7: 1087-1105.

Lee, C.H. and C.C. Teng, 2000. Identification and control of dynamic systems using recurrent fuzzy neural networks. IEEE Trans. Fuzzy Syst., 8: 349-366.

Talebi, H.A., K. Khorasani and R.V. Patel, 1998. Neural network based control schemes for flexible-link manipulators: Simulations and experiments. Neural Networks, 11: 1357-1377.

Tao, C.W., W.Y. Wang and M.L. Chan, 2004. Design of sliding mode controllers for bilinear systems with time varying uncertainties. IEEE Trans. Syst. Man Cybernet. Part B, 34: 639-645.

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

Yuangang, T., 2006. Neural network control of flexible-link manipulators using sliding mode. Neurocomputing, 70: 288-295.