Robust Adaptive Neural Fuzzy Controller Based on Computed Torque Control for Manipulators

Meysar Zeinali
Department of Mechanical Engineering
Queen's University
Kingston, Ontario, Canada
e-mail: zeinali@me.queensu.ca

Leila Notash
Department of Mechanical Engineering
Queen's University
Kingston, Ontario, Canada
e-mail: notash@me.queensu.ca

Abstract

In this article a robust controller for manipulators is proposed. The proposed controller applies adaptive fuzzy model output as a feedforward computed torque, decentralized PID controller as a feedback stabilizer, and multi-layer perceptron as a learning tool. The structure of the fuzzy model can be built either using the numerical solution of the direct dynamics of manipulator, which is fed to neural network to find inverse dynamics, or by any other input-output data that represents the inverse dynamics of manipulator in design stage. Then, to modify the fuzzy model structure, on-line learning is conducted via neural network to capture the inverse dynamics and uncertainties such as unmodelled dynamics and disturbances. The stability and robustness of the proposed controller are established using Lyapunov approach. The proposed controller is suitable for real time application.

1. Introduction

Since robots work in place of human in many fields, it might be natural to design a robot, which has a controller similar to human brain with learning, thinking, calculation, and evaluation capabilities. Design and implementation of such a controller is one of the most challenging tasks, especially when parallel manipulators with flexible links are required to maneuver very quickly and accurately under external disturbance and model uncertainty. In the last decade, much research effort has been devoted to the design of intelligent controller using fuzzy logic and neural network. Fuzzy logic provides human reasoning capabilities to capture uncertainties, which cannot be described by precise mathematical models. Neural networks offer exciting advantages such as adaptive learning, parallelism, fault tolerance, and generalization. Classical and modern control theory has been successful for systems that are well defined both in terms of deterministic and stochastic descriptions. In robotics, similar to many engineering applications, it is impossible or very difficult to obtain an accurate model of rigid or flexible manipulator to be controlled, due to the lack of detailed a priori information, complex dynamics, large dynamic coupling between different links, nonlinearity and time varying characteristics of the robot. However, robot model can at best be approximation of the real robot, as a consequence modeling error exists. This is even more so when one deals with linear time invariant model [1, 2]. To accommodate the system uncertainty, time variation of parameters and disturbance, learning, thinking and classical techniques must be incorporated.

Conventional adaptive controllers based on nonlinear control laws can achieve fine control and compensate the structured uncertainties or unknown parameters of manipulator dynamics. However, they often suffer from heavy computational burden and lack of dynamic model. As a result this hinders their real-time applications [3-8]. Although variable structure control strategy using sliding mode is an effective way to deal with uncertainties in the robotic system, the chattering phenomena due to switching operation will influence the accuracy of the tracking performance and trade-off between performance and chattering is needed [9-15]. Hence, there is a need for control strategies with learning, robustness and adaptive capability. In this regard, fuzzy logic and neural network have been proven to be very powerful techniques in the discipline of system control, especially when the controlled system is hard to model mathematically or when it has large uncertainties and strong nonlinearities. Therefore, fuzzy logic and neural networks have been widely adopted in model-free adaptive control of robot manipulators [16-26]. Furthermore, several hybrid techniques were applied to the adaptation of parameters in fuzzy or neural controllers, like genetic algorithm [27] and radial base function neural networks [28]. However, it turns out that only parameters adjustment will be insufficient in cases that they are using on-line model building method and parameter adjustment. For example, if the number of fuzzy rules, hidden layers and neurons is very large, real-time implementation will be difficult or impossible. More importantly, large number of the rules and hidden layers reduce the flexibility and numerical processing capability of the controller. As a result redundant or inefficient computation can be performed. In [29], fuzzy logic approach has been used to construct compact form of fuzzy model from crisp data, which can be utilized as a fuzzy controller. In [30], the structure of fuzzy rules was optimized by genetic algorithms. In [31], neuro-fuzzy controller was utilized to determine the inverse dynamics of robot. The number of fuzzy rules and neurons in hidden layers can be generated and deleted automatically. In [32], adaptive fuzzy compensator has been applied to the control of manipulator. These methods are successful when it is not necessary to determine the precise structure and parameters of the fuzzy or neural controllers in advance. However, in these approaches controller structure construction, totally are left to online operation of robot and will result heavy computational burden in each control loop. Moreover, on-line structure building controller suffers from the lack of systematic approach, large number of fuzzy rules, and training time. The robustness margin is also not clear enough.

This paper presents a new robust adaptive neural fuzzy controller (RANFC), which incorporated with classical PID to take advantages of classical control and neuro-fuzzy controller. The resulting intelligent controller investigates the systematic off-line fuzzy model construction and on-line modifications of fuzzy model to guarantee well defined robustness margin and fast on-line adaptability. This controller is built based on a neural network fuzzy (NNF) controller employing the generalized dynamic fuzzy neural networks (GD-FNN) learning algorithm [33]. The low-level learning and computational power of neural networks could be incorporated into the fuzzy logic system on the one hand and the high-level human-like thinking and reasoning of fuzzy logic systems could be used to build simple inverse dynamic model on the other hand. The main features of the proposed RANFC are summarized as follows:

- On-line learning. On-line learning will be used to fine-tune the controller and to cope with time varying dynamics of manipulator.
- Dynamic fuzzy structure. Fuzzy controller membership functions can be refined automatically according to their significance to the control system using numerical information via neural network.
- Fast learning speed. Weights of the NN are adjusted using modified back propagation (BP) iteration method.
- Fast convergence of tracking error. Manipulator joints can track the desired trajectory very quickly and accurately.
- Adaptive capability. Proposed controller applies a new adaptive law to update the output of fuzzy controller and structure of fuzzy model, using the tracking error in the state variables in the presence of disturbances and unmodelled dynamics to maintain the controller performance.
- Robustness. Asymptotic stability of the control system is established using the Lyapunov theorem and Barbalat's lemma.

This paper is organized as follows: Section 2 presents the general form of dynamic model of robot manipulator and its property, which will be used in the stability analysis. Section 3 introduces the details of proposed controller. Systematic fuzzy modeling is reviewed in Section 4. Robustness and global stability of the proposed controller are proven using the Lyapunov theory in Section 5. Section 6 presents the structure of neural network part and some simulation results of a two-link serial robot. Section 7 presents the manipulator that will be used as a test bed. Section 8 concludes the article.

2. Manipulator Dynamic Model

Dynamic model plays an important role in the design of control algorithms. If it was possible to derive perfect dynamic model of a system, it might need no feedback control or very little control effort to meet certain desired specifications. Therefore, it is necessary to establish the dynamic model before the control scheme is established.

Regardless of the method applied to derive the dynamic model, the equation of motion can be written in the following general form which represents the joint space dynamic model:

$$\tau = M(q,t)\ddot{q} + C(q,\dot{q},t)\dot{q} + F_f(\dot{q},t) + g(q,t) + T_d \quad (1)$$

$$\tau = M(q,t)\ddot{q} + n(q,\dot{q},t) \quad (2)$$

where $n(q,\dot{q},t) = C(q,\dot{q},t)\dot{q} + F_f(\dot{q},t) + g(q,t) + T_d$, $M(q)$ is an $n \times n$ manipulator inertia matrix (which is symmetric positive definite), $C(q,\dot{q})$ is an $n \times n$ matrix of centripetal and Coriolis terms, $g(q)$ is an $n \times 1$ vector of gravitational terms, and F_f is an $n \times 1$ vector denoting viscous and Coulomb friction coefficient, T_d is an $n \times 1$ vector arising from the unmodelled dynamics and external disturbances and τ is an $n \times 1$ vector of input generalized forces, which is generated by the active joint. In this work, it is assumed that the robot end-effector moves freely in the environment. There are two notable properties of dynamic model, which are useful for the stability check, dynamic model parameter identification and for deriving the control algorithm [34], given as:

- Skew-symmetry of matrix $(\dot{M}(q) - 2C)$.
- Linearity in dynamic parameters, which means that equation (1) can be written in compact form as follows;

$$\tau = Y(q, \dot{q}, \ddot{q})p \quad (3)$$

In the linearity equation p is a $\kappa \times 1$ vector of constant parameters (κ number of dynamic parameters such as link length and mass, moment of inertia and any constant related to the dynamic model). Y is an $n \times \kappa$ matrix, which is a function of joint position, velocity and acceleration [34, 35].

In the study of controller, it is relevant to find a solution for the *inverse dynamic* problem of manipulator. The inverse dynamic problem consists of determining the joint force/torque vector τ which is needed to generate the motion specified by the joint accelerations $\ddot{q}$, velocity $\dot{q}$, and position q. Once a joint desired trajectory is specified in terms of position, velocity and acceleration (typically as a result of an inverse kinematic procedure), inverse dynamic allows computation of the forces/torques needed to apply to joints to follow the desired trajectory. Inverse dynamic model can be used in two different ways, feedforward fashion or feedback fashion. In this work, fuzzy logic IF-THEN rules express the dynamic behavior of system. This "*knowledge base*" can be regarded as the fuzzy logic inverse dynamic model of robot that represents the interaction between the system states as well as the other complex phenomena such as flexibility and Coulomb friction in the robot. In this work, inverse dynamic model in feedforward fashion is used to compensate for the nonlinearity effects.

3. The Proposed Robust Adaptive Neuro-Fuzzy Controller

Figure 1 illustrates the structure of the proposed Robust Adaptive Neuro-Fuzzy Controller (RANFC), which consists of three parts namely, fuzzy controller, PID controller and learning algorithm. The fuzzy controller is connected in parallel with the PID controller to generate a control signal to approximate the manipulator inverse dynamics. The control law is given by:

$$\tau = \tau_{FL} + \tau_{PID} \tag{4}$$

and includes a feedforward term τ_{FL} which is the torque produced by the adaptive fuzzy logic controller, and τ_{PID} is the torque generated by the PID controller. The controller uses fuzzy model to calculate the join torques τ_{FL} , which is an estimate of the actual torque. Thus, the controller is based on the computed torque control (computed torque method uses exact dynamic model).

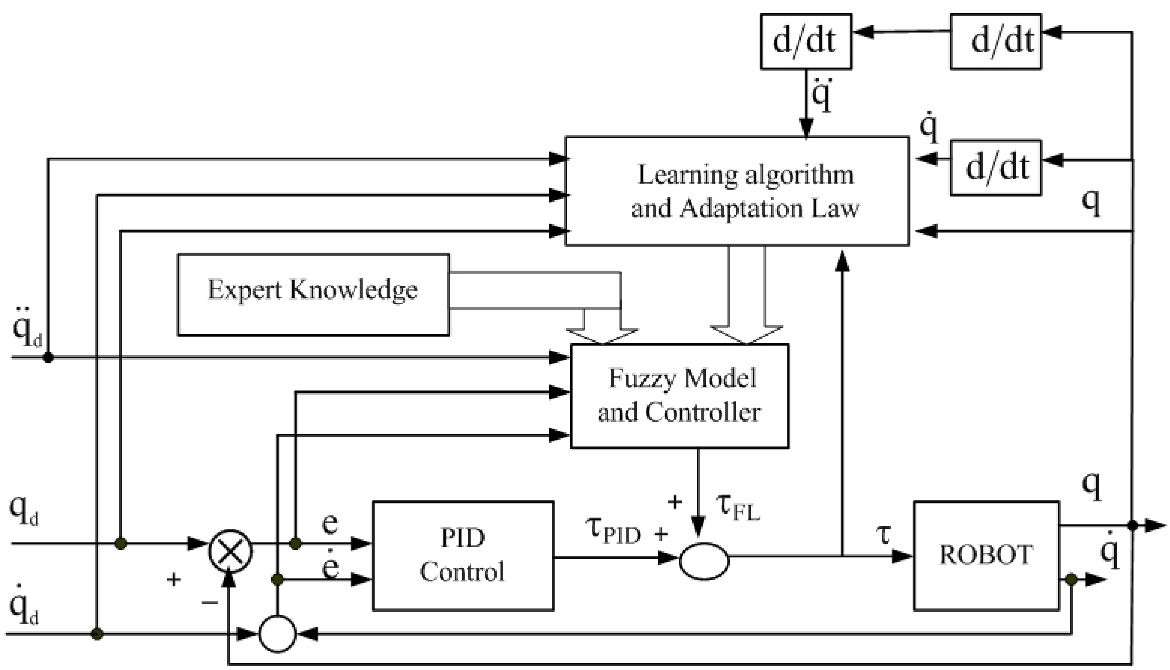

Figure 1 Proposed controller block diagram.

The robust control task is to follow a desired q_d and $\dot{q}_d$ in the presence of system parameter variations and uncertainties. The tracking error $e = q - q_d$ and rate of error $\dot{e} = \dot{q} - \dot{q}_d$ are to be observed. Based on these definitions, the generalized error vector can be considered as follows:

$$E(t) = \dot{e} + K_1 e + K_2 \int_0^t e dt \tag{5}$$

where K_1 and K_2 are $n \times n$ positive definite diagonal matrices. The integral of error is included in the generalized error to ensure zero offset error. If the dynamic model of the robot is exact, the perfect control torque based on the well known *computed torque* method can be designed as

$$\tau_d = M(q,t)(\ddot{q}_d - K_1\dot{e} - K_2 e) + n(q,\dot{q},t) \tag{6}$$

where $n(q,\dot{q},t) = C(q,\dot{q},t)\dot{q} + F_f(\dot{q},t) + g(q) + T_d$. Replacing τ_d of controller by τ in equation (1) yields:

$$\ddot{e} + K_1\dot{e} + K_2 e = 0 \tag{7}$$

which implies that the tracking error will converge to zero with proper choice of K_1 and K_2 [35, 36]. But the fact is that, the external disturbance and unmodelled dynamics represented by T_d are unknown in practice. Therefore, to compensate for uncertainties, the dynamic structure fuzzy controller is proposed to generate the optimal torque τ_{FL} to approximate the perfect control law. The systematic methodology of design and analysis of the proposed structure is presented by the following steps:

- *Development of fuzzy logic model*. The main knowledge about the system is encapsulated in fuzzy IF-THEN rules. This will be discussed in Section 5.
- *Proof of the stability and convergence.* For the proposed structure, the stability and robust performance of fuzzy and PID controller by defining a new Lyapunov function and using Slotine and Li [38] adaptive law is presented in Section 5.
- *Learning algorithm and adaptation law.* Learning process will be carried out in two phases, off-line and on-line. In off-line learning the input-output data from nominal dynamical model governed by equation (1) or from CAD system (VisualNastran or other numerical solution) is used to construct the fuzzy model based on the systematic method reviewed in Section 4. Then on-line learning can pursue to capture the uncertainties by actually operating the manipulator. More detail will be discussed in Section 6.

- *Designing and tuning of PID controller for each state independently*. For this part, the common existing methods are used.

4. A Review of Systematic Fuzzy Logic Modeling of Robot

Generally, the inverse dynamic model of *n* degree-of-freedom (DOF) manipulators with *m* actuated joints and using equations (1) through (3) can be represented as

$$\tau_i = F_i(q,\ \dot{q},\ \ddot{q},\ p) \qquad i = 1,2,3,...,m \tag{8}$$

And more generally, for time-variant-parameter manipulators equation (3) in vector form can be written as follows

$$\tau = F(q,\dot{q},\ddot{q},t) \tag{9}$$

From the fuzzy logic point of view, the encoded knowledge of the robot dynamics can be interpreted by *m* fuzzy models. Each model expresses the variation of one joint force/torque as a result of motion of all joints. In what follows the systematic methodology of fuzzy model construction will be reviewed.

Fuzzy modeling procedure can be formulated briefly and systematically, using three distinct steps as follows:

1. Linguistic variables in place of, or, in addition to numerical variables.
2. Simple relation between variables based on IF-THEN fuzzy rules.
3. Formulation of complex relations by fuzzy reasoning algorithms.

The first step in fuzzy modeling is the procedure of finding the significant input-output data. Available information or data can be found in the following classification of the sources.

- Conventional mathematical models
- Observation based on knowledge and/or experience
- Numerical data (from excitation of system or from numerical solution of direct dynamics)
- Image data
- Linguistic data

In this work, the qualitative model (linguistic model) of the system (robot) will be constructed based partly on the conventional mathematical model and the expert knowledge, and mainly on the numerical data measured by the joint sensors.

The second step of fuzzy modeling is rule extraction, which can be proceeded by three main types of fuzzy modeling namely, *Mamdani Fuzzy Models, Takagi-Sugenuo Fuzzy Models* and *Tsukamoto Fuzzy Models* that have been widely employed in various applications including robot control.

However, in the most general form, the encoded knowledge of multi-input-multi-output (MIMO) nonlinear system can be represented by fuzzy models consisting of IF-THEN rules with multi-antecedent and multi-consequent variables (with r antecedents, z consequent, and N rules).

$$\begin{array}{l} \text{IF } a_1 \text{ is } Q_{11} \text{ AND } a_2 \text{ is } Q_{12} \ldots \text{ AND } a_r \text{ is } Q_{1r} \text{ THEN } c_1 \text{ is } D_{11} \text{ AND } \ldots \text{ AND } c_z \text{ is } D_{1z} \\ \text{ALSO} \\ \vdots \\ \text{ALSO} \\ \text{IF } a_1 \text{ is } Q_{N1} \text{ AND } a_2 \text{ is } Q_{N2} \ldots \text{ AND } a_r \text{ is } Q_{N\mathrm{r}} \text{ THEN } c_1 \text{ is } D_{N1} \text{ AND } \ldots \text{ AND } c_z \text{ is } D_{Nz} \end{array} \tag{10}$$

where $a_1,a_2,...,a_r$ are input variables, and $c_1,c_2,...,c_z$ are output variables, Q_{1r} $Q_{2r},...,Q_{Nr}$ and $D_{1z},D_{12},...,D_{Nz}$ are the fuzzy sets of the universes of discourse which represent the input and output membership functions (MF), respectively [29].

Conceptually, a system with multiple independent output variables can be considered as a set of single output system. Consequently, the general structure of MIMO fuzzy system can also be considered as a collection of multi-input-single-output (MISO) fuzzy systems. Although for MISO fuzzy system the number of rules will be increased, modeling and inference will be more straightforward. That is why the literature concentrates on multi-input-single-output rules as generic presentation of fuzzy systems. Using MISO system for inverse dynamic problem of robots leads to the following form of rules for each active joint of a manipulator:

$$\begin{array}{l} \text{IF } q_1 \text{ is } Q_{m11} \text{ AND } q_2 \text{ is } Q_{m12} \ldots \text{ AND } q_r \text{ is } Q_{m1r} \text{ THEN } \tau_m \text{ is } D_{m1} \\ \text{ALSO} \\ \vdots \\ \text{ALSO} \\ \text{IF } q_1 \text{ is } Q_{mN1} \text{ AND } q_2 \text{ is } Q_{mN2} \ldots \text{ AND } q_r \text{ is } Q_{mNr} \text{ THEN } \tau_m \text{ is } D_{\mathrm{mN}} \end{array} \tag{11}$$

where m is the number of actuated joints; q_1, $q_2,\ldots,q_r$ are significant input variables for joint i (i=1,2,…,m) that were identified among the elements of the joint displacement, velocity and acceleration. τ_m is the output torque of joint m; $Q_{m1r},\ldots,Q_{mNr}$ and $D_{m1},\ldots,D_{mN}$ are the fuzzy sets representing the input and output MF, respectively [39].

The third step of fuzzy modeling is fuzzy reasoning, which is an inference procedure that derives conclusion from a set of fuzzy IF-THEN rule and known facts. One of the most applicable fuzzy reasoning, which has been widely used, is classical MAX/MIN composition method. This method considers the maximum of membership sets, which consists of the minimum of membership of antecedent part of different rules. For instance, for every similar D_{mN}:

$$\mu_{A-out}(\tau_m) = \text{MAX}\{\min\{Q_{m11},\ldots, Q_{m1r}\},\ldots,\min\{ Q_{mN1},\ldots,Q_{mNr}\}\} \tag{12}$$

where $\mu_{A-out}(\tau_m)$ is the aggregated membership function of the fired rules for specific output. By using the concept of fuzzy partitioning of the information (fuzzy partitioning of state variable domain), rules and fuzzy reasoning, decision-making can be accomplished.

The last step is defuzzification, which is the conversion of a fuzzy output value to an equivalent crisp value. In general there are five methods for defuzzyfying a fuzzy set, namely, *centroid-of-area* (COA), *mean of maximum* (MOM), *smallest of maximum* (SOM), *largest of maximum* (LOM) and *bisector of area* (BOA). One of the methods that widely have been used is the centroid-of-area method (COA) and can be expressed as follows:

- Multiply the membership degrees for each output variable by the singleton value of the output set.
- Add all of the preceding together and divide by the summation of output membership degrees.

$$COA = \frac{\int \mu_{A-out}(\tau_m)\tau_{s-m} d\tau}{\int \mu_{A-out}(\tau_m) d\tau} \tag{13}$$

where τ_{s-m} is the singleton value of output torque set, which is the output value with membership function one (100%) in case that there is one point with membership function one.

The key idea of this approach is to consider the fuzzy logic model (with crisp input and output) as a multi-dimensional nonlinear operator with upper and lower limits. The nonlinear characteristics of fuzzy logic model are due to its computational structure, fuzzification, inference and defuzzification. Formulation has been made based on two main assumptions:

- This approach to fuzzy logic modeling and control does not consider the internal parameters of system. Hence, the system model and control rules must be obtained from the input-output data, and to achieve this goal in the design and simulation phase, the input-output data either from CAD system (visualNastran) or approximate mathematical model will be fed to the neural network part of controller to capture an approximate inverse dynamics of the system.
- For simplicity, the control rules will be designed for each system state independently, despite the state interactions, while the stability and robustness of the entire system is guaranteed.

5. Stability and Robustness Analysis of Proposed Structure

Before the proof of stability of the closed loop controller, rewriting some of the equations and new arrangements are needed. Equating equations (2) and (9) results

$$\tau = M(q,t)\ddot{q} + n(q,\dot{q},t) = F(q,\dot{q},\ddot{q},t) \tag{14}$$

Taking derivative of $E(t)$ in equation (5) yields

$$\dot{E} = \ddot{e} + K_1\dot{e} + K_2 e \tag{15}$$

By rearranging and equating $K_1 = 2\Lambda$ and $K_2 = \Lambda^2$ an optimum response (critically damped) for each error state will be obtained as follows

$$\begin{aligned}\dot{E} &= \ddot{e} + 2\Lambda\dot{e} + \Lambda^2 e \\ &= (\ddot{q} - (\ddot{q}_d - 2\Lambda\dot{e} - \Lambda^2 e))\end{aligned} \tag{16}$$

The acceleration vector $\ddot{q}$ can be obtained from the system dynamics by rewriting equation (2) as

$$\ddot{q} = M^{-1}[\tau - n(q,\dot{q},t)] \tag{17}$$

Inserting (17) into (16) results in

$$\dot{E} = M^{-1}[\tau - (M(\ddot{q}_d - 2\Lambda\dot{e} - \Lambda^2 e) + n(q,\dot{q},t))] \tag{18}$$

In equation (18) the term $M(\ddot{q}_d - 2\Lambda\dot{e} - \Lambda^2 e) + n(q,\dot{q},t)$ is the system inverse dynamics with the input acceleration called "*reference*" acceleration and defined as follows

$$\ddot{q}_r = \ddot{q}_d - 2\Lambda\dot{e} - \Lambda^2 e \tag{19}$$

On this basis, the desired control input can be defined as

$$\tau_d = M(q,t)\ddot{q}_r + n(q,\dot{q},t) = F_d(q,\dot{q},\ddot{q}_r,t) \tag{20}$$

Because of the system uncertainty and variation, the inverse dynamic model (*in this case fuzzy logic model*) is an approximation of the real system. Hence

$$\hat{\tau} = \hat{M}(q,t)\ddot{q}_r + \hat{n}(q,\dot{q},t) = \hat{F}(q,\dot{q},\ddot{q}_r,t) \tag{21}$$

where $\hat{F}(q,\dot{q},\ddot{q}_r,t) = F(q,\dot{q},\ddot{q}_r,t) + \Delta F(q,\dot{q},\ddot{q}_r,t)$. At this point the uncertainty ΔF is assumed to be bounded

$$\left\|\Delta F(q,\dot{q},\ddot{q}_r,t)\right\| \le \rho(q,q,\ddot{q}_r,t) \tag{22}$$

Therefore the control input in equation (4) will be

$$\tau_d = \hat{\tau}_{FL} + \tau_{PID} \tag{23}$$

where the control term τ_{PID} is the stabilizing term and based on $e = q - q_d$ it can be defined as follows

$$\tau_{PID} = -K(\dot{e} + K_1 e + K_2 \int_0^t e\,dt) = -KE \tag{24}$$

where K is a strictly positive real number which represents the contribution of the PID controller, and $\hat{\tau}_{FL}$ is fuzzy controller output. $\hat{\tau}_{FL}$ will be updated in the presence of disturbance and uncertainties by on-line training and membership adjustment based on the following adaptation law

$$\dot{\hat{F}}(q,\dot{q},\ddot{q}_r,t) = -\Gamma E \tag{25}$$

where Γ is a symmetric positive definite matrix which defines rate of adaptation. Updating the fuzzy model output according to equation (25) will keep the output of fuzzy model as close as to the actual output, and then the PID controller can stabilize the system by little control effort. Now based on the established adaptation law and error vector the proof of *stability theorem* will be addressed.

Stability Theorem: *Consider the multi link parallel manipulator system represented by equations* (1) *and* (9), *if the robust control law of* (4) *and the adaptive law that will be derived later in this work are applied, asymptotic stability is guaranteed.*

Proof. By using equations (16) and (19), the derivative of generalize error vector can be defined as

$$\dot{E} = \ddot{q} - \ddot{q}_r \tag{26}$$

where $\ddot{q}_r = \ddot{q}_d - 2\Lambda\dot{e} - \Lambda^2 e$. A Lyapunov function candidate can be considered as follows

$$V = \frac{1}{2}\,(E^T M E + \tilde{F}^T \Gamma^{-1}\tilde{F}) \tag{27}$$

Taking the derivative of Lyapunov function and using definition $\tilde{F} = \hat{F}(q,\dot{q},\ddot{q}_r,t) - F(q,\dot{q},\ddot{q}_r,t) = \Delta F$, results

$$\dot{V} = \frac{1}{2}E^T\dot{M}E + E^T M\dot{E} + \dot{\hat{F}}^T\Gamma^{-1}\tilde{F} \tag{28}$$

Inserting (26) into (28) yields

$$\dot{V} = \frac{1}{2}E^T\dot{M}E + E^T M(\ddot{q} - \ddot{q}_r) + \dot{\hat{F}}^T\Gamma^{-1}\tilde{F} \tag{29}$$

By using system dynamic equation (1)

$$\dot{V} = \frac{1}{2}E^T\dot{M}E + E^T(\tau - (C(q,\dot{q},t)\dot{q} \ + \ F_f(\dot{q},t) \ + \ g(q,t) \ + T_d)) - E^T M\ddot{q}_r + \dot{\hat{F}}^T\Gamma^{-1}\tilde{F} \tag{30}$$

Replacing $\dot{q}$ by integrating the reference acceleration (19), $\dot{q} = E + \dot{q}_r$, in equation (30), yields

$$\dot{V} = \frac{1}{2}E^T\dot{M}E + E^T[\tau - (C(q,\dot{q},t)(E + \dot{q}_r) + F_f(\dot{q},t) + \ g(q,t) \ + T_d)] - E^T M\ddot{q}_r + \dot{\hat{F}}^T\Gamma^{-1}\tilde{F} \tag{31}$$

Rearranging equation (31)

$$\dot{V} = \frac{1}{2}E^T(\dot{M} - 2C)E + E^T[\tau - (M\ddot{q}_r + C(q,\dot{q},t)\dot{q}_r + F_f(\dot{q},t) + g(q,t) + T_d)] + \dot{\hat{F}}^T\Gamma^{-1}\tilde{F} \quad (32)$$

where the skew symmetry of $(\dot{M} - 2C)$ has been used to eliminate the term $\frac{1}{2}E^T(\dot{M} - 2C)E$. Then

$$\dot{V} = E^T[\tau - (M\ddot{q}_r + C(q,\dot{q},t)\dot{q}_r + F_f(\dot{q},t) + g(q,t) + T_d)] + \dot{\hat{F}}^T\Gamma^{-1}\tilde{F} \quad (33)$$

$$F(q,\dot{q},\ddot{q}_r,t) = M\ddot{q}_r + C(q,\dot{q},t)\dot{q}_r + F_f(\dot{q},t) + g(q,t) + T_d \quad (34)$$

$$\dot{V} = E^T(\tau - F) + \dot{\hat{F}}^T\Gamma^{-1}\tilde{F} \quad (35)$$

Taking the control law from equations (23) and (24) to be

$$\tau = \hat{\tau}_{FL} - KE = \hat{F} - KE \quad (36)$$

and inserting (36) into (35) yields

$$\dot{V} = E^T(\hat{F} - KE - F) + \dot{\hat{F}}^T\Gamma^{-1}\tilde{F} \quad (37)$$

Updating the fuzzy controller output according to update law $\dot{\hat{F}} = -\Gamma E$, yields

$$\dot{V} = E^T\tilde{F} - E^TKE - E^T\Gamma\Gamma^{-1}\tilde{F} \quad (38)$$

$$\dot{V} = -E^TKE \le 0 \quad (39)$$

Furthermore, from equation (27) $V(t) > 0$ is positive and from equation (39) $\dot{V}(t) \le 0$. These imply that $\dot{V}(t) = 0$ if and only if $\dot{E}(t) = 0$, and Barbalat's lemma [38] indicates that $\dot{V}$ tends to zero if it is uniformly continuous. Also it is possible to show that $E(t) \to 0$ *as* $t \to \infty$. As a result, the control system is asymptotically stable and the tracking error will converge to zero.

6. Neural Network Architecture

The proposed controller learning algorithm, depicted in Figure 2, has been constructed based on the Multi Layer Perceptron (MLP) network and uses enhanced Back Propagation (BP) algorithm as a learning method. This learning algorithm was developed recently by the author for the real-time application and is simple and efficient, and also faster than regular BP algorithm. As shown in Figure 2, the developed NN has three layers. The nodes in layer one are input nodes that represent numerical input data measured by the joint sensors, which are joint displacements, velocities and accelerations. Numbers of input nodes are at most $3m+1$, where m is the number of actuated joints of the manipulator. The nodes in layer two are processing nodes and are designed based on the trial and error method, which can approximate nonlinear functions in a wide range. The last layer is output layer and represents the joint torques. In this work, for each actuated joint of parallel manipulator individual NN with identical structure has been considered to capture the inverse dynamics of robot. To demonstrate the learning ability and fast convergence of algorithm, a two-link serial robot has been simulated. The inputs of joints one and two are defined as $q_1 = \pi/6(1-\cos(2\pi t))$, $q_2 = 2\sin(\pi t)$, respectively, and their first and second derivatives.

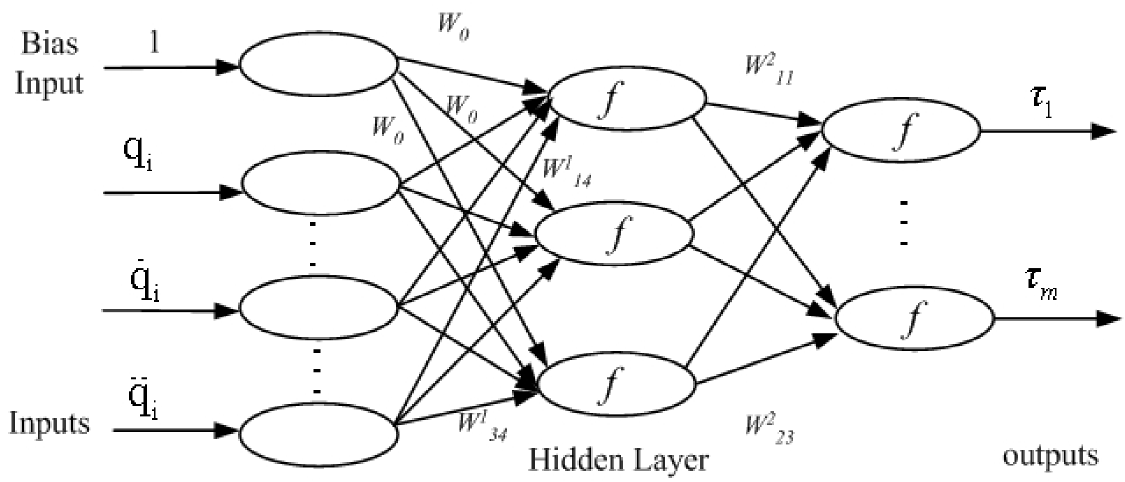

Figure 2 Three layer feedforward Neural Network.

Figure 3 represents the displacement, velocity and acceleration of joint one, and the output torque of this joint (desired output and NN output). As shown in Figure 3, the proposed neural network structure can learn up to the desired accuracy. In this analysis the nonlinear output torque with small number of observation of the training data can be approximated. Figure 4 represents the first 0.1 seconds of the first joint's torque (marked within a square in Figure 3) to show how fast neural network can follow the desired trajectory and approximate the inverse dynamics of the robot.

The robustness of the proposed NN structure in the presence of the disturbances has been investigated. Disturbance in the form of a sine function with different frequency and amplitude has been applied to the joint inputs at time 0.8 seconds. Figure 5 shows the disturbed inputs of joint one and its output. Figure 5 demonstrates how fast and accurate the proposed NN structure can accommodate the disturbance.

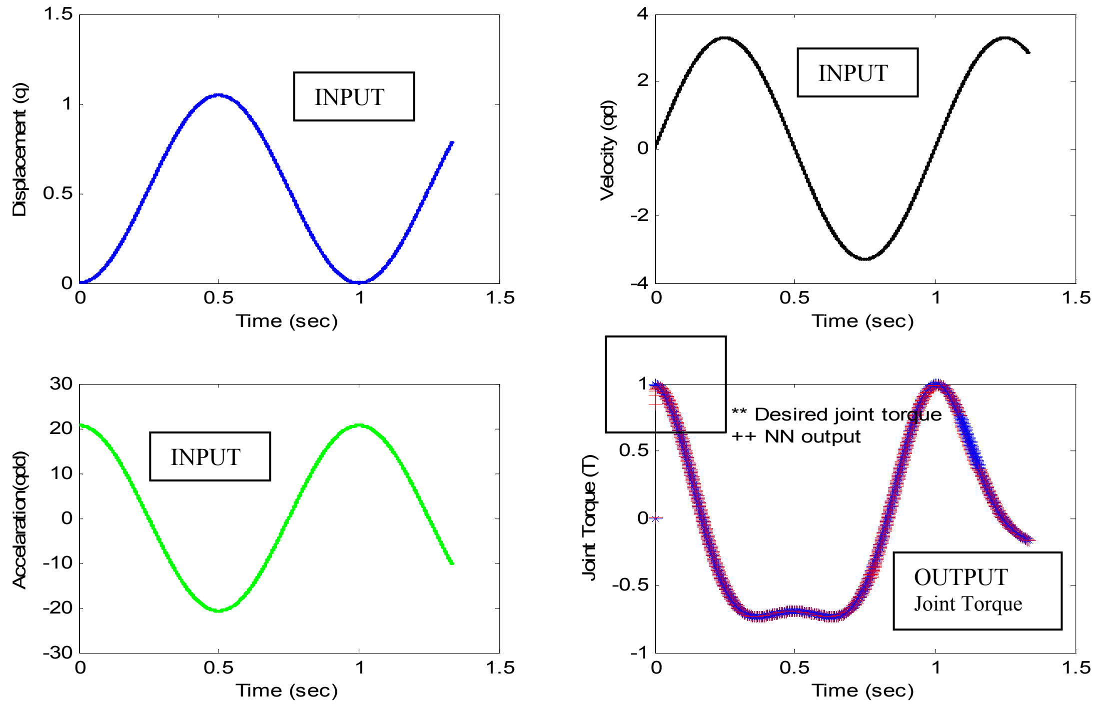

Figure 3 First joint inputs and torque approximation of a two-link serial robot by NN.

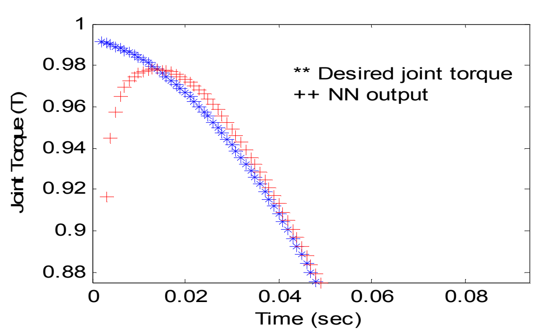

Figure 4 Joint torque approximation for the first 0.1 second of robot by NN.

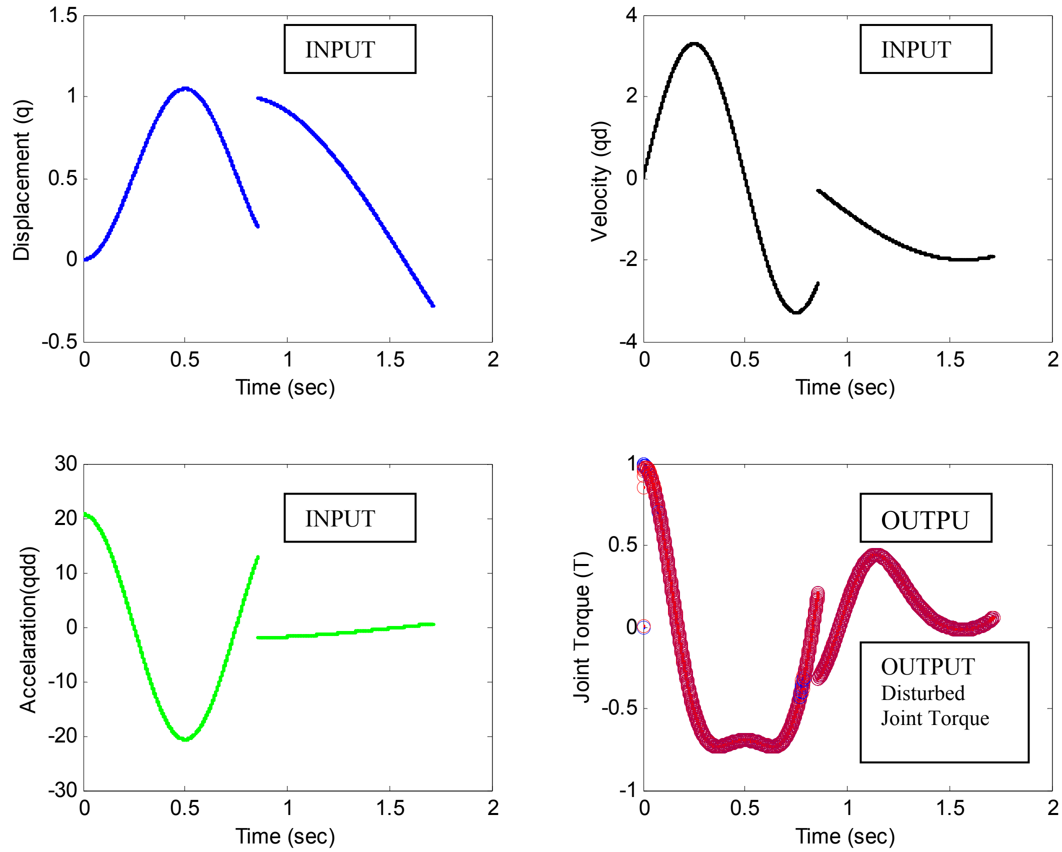

Figure 5 First joint inputs and torque approximation of robot by NN with disturbance.

7. Simulation

To demonstrate the improved performance, adaptive capability and robustness of the proposed controller, simulation of the controller has been conducted on the 3 DOF serial Phantom robot (model 1.5) [40].

The simulation was conducted in the presence of disturbance and payload change. Figures 7, 8 and 9 demonstrate the simulation results for three actuated the joints of Phantom robot. Results show that in the presence of 500% change in the entries of the mass matrix at time t = 1.5 seconds and 1000% change in the acceleration terms (**C** matrix), the controller performs very well. After small chattering in the joint torques, the tracking error is in the order of 0.001 mm, the convergence after disturbance to the desired trajectory is fast, and the stability is excellent.

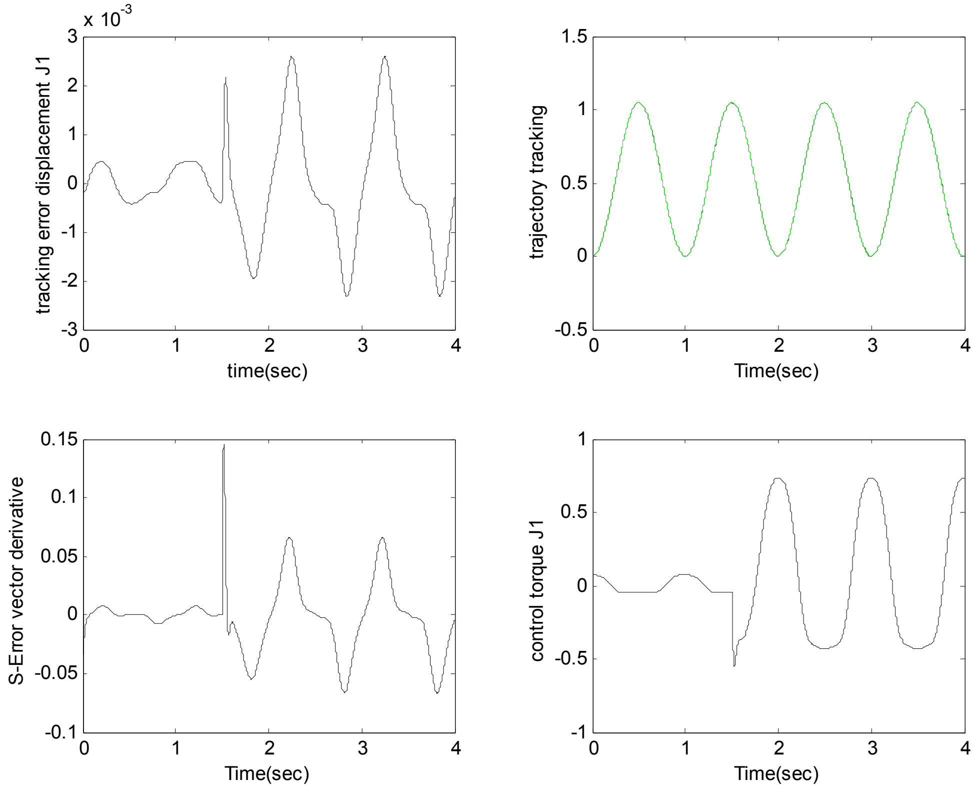

Figure 7 Simulation results for joint 1 of Phantom robot.

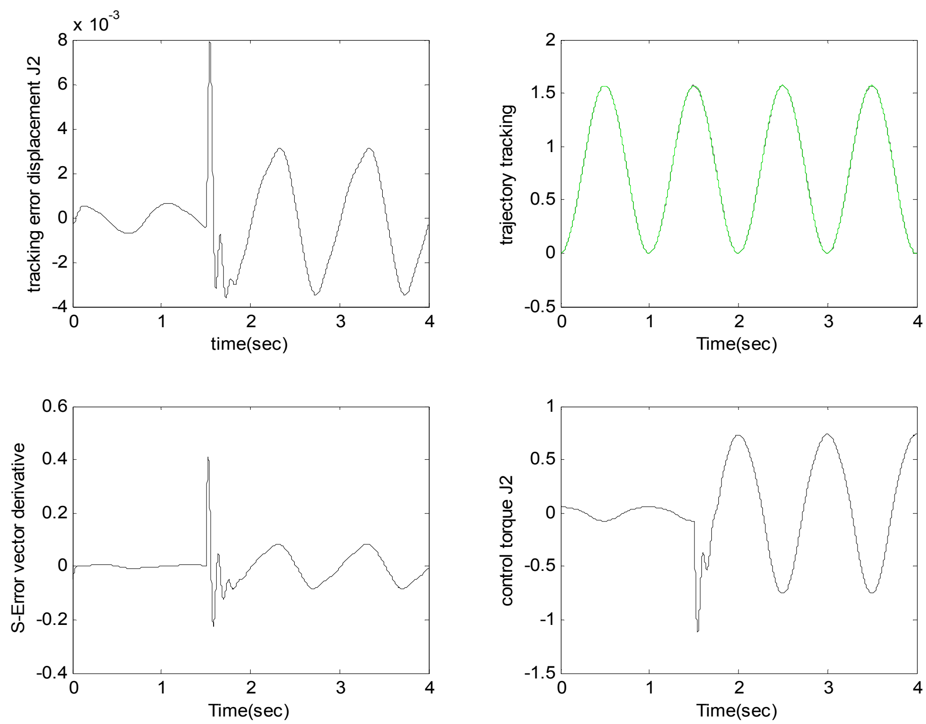

Figure 8 Simulation results for joint two of Phantom robot.

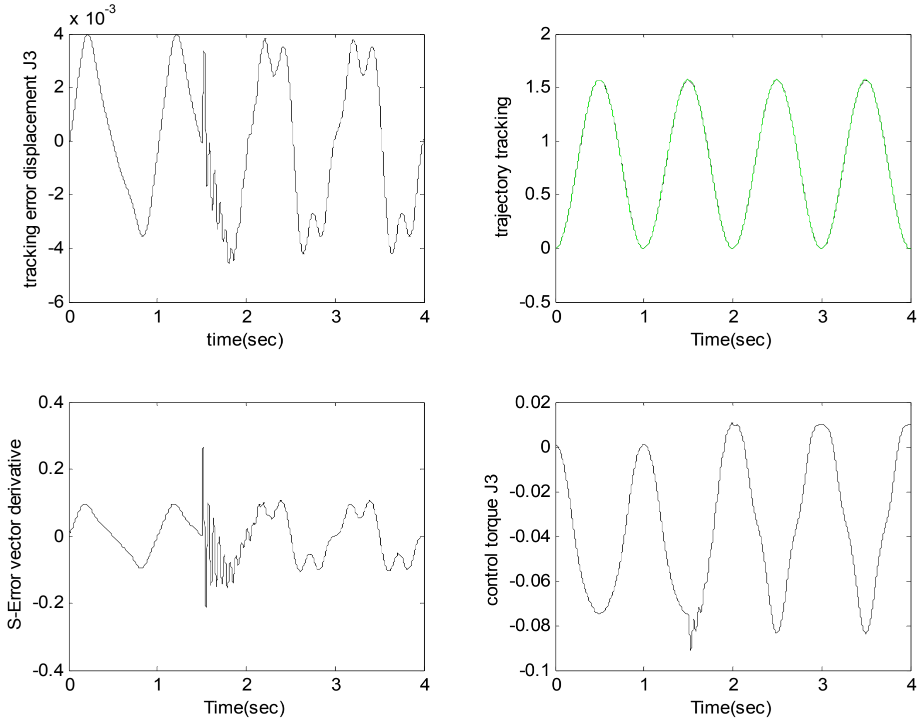

Figure 9 Simulation results for joint three of Phantom robot.

The proposed controller will be implemented on a 4 DOF cable-actuated manipulator for experimental evaluation. The robot has three translational DOF and one rotational DOF. The design of this cable-actuated manipulator has been reported in [41]. The developed manipulator consists of a central rigid linkage with two actuated joints and three cables that are used to control the end effector. The central linkage incorporates eight revolute joints to achieve 4 DOF motion.

8. Conclusion

A combined form of robust adaptive neural fuzzy controller based on the computed torque control theory along with the supervisory PID controller for real time applications is proposed. The asymptotic stability and convergence of controller is established using the Lyapunov approach. The control of manipulators can be initiated with or without simple fuzzy model, which is constructed from the numerical solution of direct dynamics. Then, the modification of the fuzzy model can be accomplished using the adaptation law and on-line training with neural network algorithm developed in this work.

References

1. Chandrasekharan, P. C., Robust Control of Linear Dynamical Systems. Academic Press Limited, 1996.
2. Zhou, k., Doyle, C. J., and Glover K., Robust Optimal Control, Prentice-Hall, Inc. 1996.
3. Astrom, K. J., and Wittenmark, B., Adaptive Control, (New York, NY: Addison-Wesley, 1995.
4. Honegger, M., and Codourey, A., Adaptive control of the Hexagild, a six DOF parallel manipulator, IEEE Proc., Intl. Conf. on Robotic and Automation, 1997pp. 543-548.
5. Landau, Y., Adaptive control, The model reference approach, Marcel Dekker Inc. 1979.
6. Luh, J. Y. S., Conventional controller design for industrial robots – A Tutorial, IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-13, No. 3, 1982, pp. 298-316.
7. Nguyen, C., Antrazi, S. S., and Zhou, Z. L., Adaptive control of a Stewart platform-based manipulator, Journal of Robotics Systems, 10(5), 1993, pp. 657-687.
8. Craig, J. J., and Sastry, S. S., Adaptive control of mechanical manipulators, Intl. Journal of Robotics Research, Vol. 6, No. 2, 1987, pp. 16-28.
9. Utkin, V., variable structure systems with sliding modes, IEEE, Trans. Auto. Control, Vol. AC-22, No.2, pp 212-222.
10. Young, K. D., Utkin, V. I., and Ozguner, U., A control engineer's guide to sliding mode control, IEEE Trans. on Cont. Systems Technology, Vol. 7, No 3, 1999, pp. 328-342.
11. Slotine, J. J. E, and Sastry, S. S., Tracking control of non-linear systems using sliding surface, with application to robot manipulators, INT. J. Control, Vol. 38, No. 2, 1983, pp. 465-492.
12. Slotine, , J. J. E, and Coetsee, J. A., Adaptive sliding controller synthesis for non-linear systems, INT. J. Control, Vol. 43, No. 6, 1986, pp. 1631-1651.
13. Rg, B., and Hd, U., An adaptive fuzzy sliding-mode controller, IEEE Trans. on Industrial Electronics, Vol. 48, No. 1, 2001, pp. 18-31.
14. Begon, P., Pierrot, F., and Dauchez, P., Fuzzy sliding mode control of a fast parallel robot, Proc. IEEE Intl. Conf. on Robotics and Automation, Vol. 1, 1995, pp. 1178-1183.
15. Harashima, F., Xu, J. X., and Hashimoto H., Tracking control of robot manipulator using sliding mode, IEEE Trans. on Power Elec., 1987, pp. 169-176.
16. Benallegue, A., Meddah, D. Y., and Daachi, B., Stable adaptive controller using MLP neural networks for rigid robot manipulators, Intl. Journal of Robotics and Automation, Vol. 16, No. 3, 2001, pp. 124-130.
17. Sun, F.C., Sun, Z.Q., and Feng, G.., Design of adaptive fuzzy sliding mode controller for robot manipulators, IEEE Intl. Conf. on Fuzzy Systems, Vol. 1, 1996, pp. 62-67.

18. Guez, A., Elibert, J., and Kam, M., Neural network architecture for control, IEEE, Control System Mag. Vol. 8 n 2, 1988, pp. 22-25.
19. Hornik, K., Stinchcombe, M., and Wite, H., Multilayered feedforward network are universal approximator, Neural Network, Vol. 2, No. 5, 1989, pp. 359-366.
20. Huang, Y., and Yasunobu, S., A general practical design method for fuzzy PID control from conventional PID control, IEEE, 2000, pp. 969-972.
21. Hyung, S. and Chul, H., Estimation for forward kinematic solution of Stewart platform using neural network, IEEE, Intl. Conf. on Intelligent Robots and system, 1999, pp. 501-506
22. Mehrotra, K., Mohan, C. K., and Ranka, S., Elements of artificial neural networks, The MIT Press, 2000.
23. Nabhan, T. M., and Zomaya, A. Y., Toward generating neural network structures for function approximation, Neural Networks, Vol. 7, No. 1, 1994, pp. 89-99.
24. Nguyen, D. H., and Widrow, B., Neural Networks for self-learning control systems, IEEE Control Systems Mag., 1990, pp. 18-23.
25. Patino, H. D., Carelli, R., and Kuchen, B. R., Neural Networks for advanced control of robot manipulators, IEEE Trans. on Neural Networks, Vol. 13, No. 2, 2002, pp. 343-353.
26. Tao, J. M., and Luh, J. Y. S., Application of neural network with real-time training to robust position / Force Control of multiple robots, IEEE, 1994, pp. 142-148
27. Park, D., Kandel, A., and Longhols, G., Genetic based ne fuzzy reasoning models with application to fuzzy control, IEEE, Trans on System, man, and Cybernetics, 24, 1994, pp. 39-47.
28. Lin, F. J., Hwang, W. J., and Wai, R. J., A supervisory fuzzy neural network control system for tracking periodic input, IEEE, Trans. Fuzzy Systems, 7(1), 1999, pp. 41-52.
29. Sugeno, M., and Yasukawa T., A fuzzy logic based approach to qualitative modeling, IEEE Transaction on fuzzy system 1, 1993, pp 7-31.
30. Jin, Y., Decentralized adaptive fuzzy control of robot manipulators, IEEE, Trans on System, Man, and Cybernetics, 28 1998, pp. 47-58.
31. Gao, Y., and Er, M. J., Adaptive fuzzy neural control of multiple-link robot manipulators, Intl. Journal of Robotics and Automation, Vol. 16, No. 4, 2001, pp. 172-182.
32. Yoo, B. K., and Ham, W.C., Adaptive control of robot manipulators using fuzzy compensator, Part I, IEEE Intl. Conf., On Intelligent Robots and Systems, Vol. 1, 1999, pp. 35-40.
33. Jang, J. S. R., Sun, C. T., and Mizutani, E., Neuro-Fuzzy and Soft Computing, Prentice-Hall, Inc.1997.

34. Sciavicco, L., and Siciliano, B., Modeling and Control of Robot Manipulators, (2nd Ed.), Springer-Verlag London Limited, 2000.
35. Yiu, Y.K., Cheng H., Xiong, Z.H., Liu, G.F., and Li Z.X., On the dynamics of parallel manipulator, IEEE, Intl. Conf. on Robotic and Automation, 2001, pp 3766-3771.
36. Craig, J. J., Introduction to Robotics: Mechanics and Control, 2nd Ed., Addison-Wesly Publ. Inc.1989.
37. Spong, M.W., and Vidyasagar, M., Robot Dynamics and Control, John Wiley and Sons, Inc., 1989.
38. Slotin, J. J. E, and Li , W., Applied Nonlinear Control, Prentice-Hall, Inc. 1991.
39. Emami, M. R., Goldenberg A. A., and Turksen, I. B., Fuzzy-logic control of dynamic system: from modeling to design, Engineering Application of Artificial intelligence, Vol. 13, 2000, pp. 47-69.
40. Cavusoglu, M.C., Feygin, D., and Tendick, F., A critical study of the mechanical and electrical properties of the PHANToM haptic interface and improvements for high-performance control, Presence-Teleoperators and Virtual Environments, Vol. 11, No. 6, 2002, pp 555-568.
41. Mroz, G., and Notash, L., Design and prototype of parallel, cable-actuated robot, to be presented at 11th World Congress in Mechanism and Machine Science, 2004.