A NN Controller and Tracking Error Bound for Robotic Manipulators

Jinyu Li Danwei Wang
School of Electrical and Electronic Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798
Email: edwwang@ntu.edu.sg

Abstract

In this paper, a robust neural network control scheme is proposed for robot tracking tasks. The neural network is trained on-line and the weight tuning algorithm has a small dead zone to overcome bounded disturbances. Under this proposed control scheme, it is shown that tracking error bound is completely determined by neural network approximation error bound, disturbance bound, as well as a control design parameter. The tracking error bound does not depend on the weight estimation errors. A two-link manipulator is used to illustrate the performance of the control scheme.

1 Introduction

Robot manipulators are highly nonlinear and dynamically coupled multi-input and multi-output systems. To achieve high precision tracking control, the coupling between the joints, the unmodeled disturbances, and the payload changes must be taken into consideration. This is a challenging task in control theory. So far, a number of control schemes have been developed, such as computed torque control, adaptive control, and iterative learning control. Computed torque control, which relies on the exact cancelation of the nonlinear dynamics of the manipulator system, requires the exact dynamic model. The performance of this method is degraded by the unmodeled dynamics and payload variation. Adaptive control and iterative control strategies require no *a priori* knowledge of unknown parameters. Many good results have been reported using these two schemes [1, 2, 3]. But the former requires the system structure to be known in advance and the latter requires the operation to be repetitive.

In the past decade, control of manipulator using neural networks(NN) has inspired the interest of many researchers because NN has powerful learning and approximation ability. Albus [4] and Miller et al. [5] used the cerebellar model articulation controller(CMAC) to control manipulators and obtained satisfying results. Johnson et al. [6] proposed an adaptive model based neural network controller which was evaluated experimentally. Although there is a lack of firm theoretical foundation in neural control, many theoretical results have been examined in [7, 8, 9, 10, 11, 12, 13] for the NN tracking control which can guarantee the convergence of the robot manipulator tracking error. But some previous studies have certain limitations. For example in [9, 10], Lewis et al. reported that there was a trade-off between the magnitudes of tracking error and weight error. In the convergence proof of [8], first order approximation is applied to deal with the nonlinear activation function. This is not an adequate method to deal with the nonlinear neurons [13].

In this paper, we proposed a robust manipulator control scheme using a three-layer neural network. The weight tuning algorithm we proposed is on-line and has a dead zone which guarantees the convergence and robustness of the system in the presence of bounded disturbances. The tracking error bound is decided by the NN approximation error, the bounds of disturbances, as well as the minimum singular value of the conventional gain matrix. When compared with the result of Lewis et al. [10], our scheme offers the advantage that the tracking error bound does not depend on the weight estimation errors. In this paper, we use the mean value theory to effectively prove the convergence of tracking error. Finally the proposed NN controller is implemented on a two-link manipulator to demonstrate its performance.

2 Background

We denote R as real numbers, R^n as real n-vectors, and $R^{m\times n}$ as real $m \times n$ matrices. $X = (x_1,\ x_2,\ \dots,\ x_n)$ denotes the n-vector. The Euclidean norm (i.e. 2-norm) is defined as

$$\|X\|_2^2 = \sum x_i^2.$$

Given $A = [a_{ij}],\ B \in R^{m\times n}$, the Frobenius norm is defined as

$$\|A\|_F^2 = tr(A^T A) = \sum a_{ij}^2,$$

where $tr(.)$ is the trace of matrix. The associated inner product is $\langle A,\ B\rangle_F = tr(A^T B)$.

Throughout this paper, $\|.\|$ denotes the Euclidean norm of vector or the Frobenius norm of matrix.

2.1 Neural Networks

Diagram of a three-layer neural network is shown in Figure 1. The output of the neural network is given by

$$y_i = \sum_{j=1}^{N_h}[w_{ij}\sigma[\sum_{k=1}^{N_i} v_{jk}\varphi(x_k) + \theta_{vj}] + \theta_{wi}],\quad i = 1, ..., N_o, \tag{1}$$

where $\sigma(.)$ and $\varphi(.)$ are activation functions, v_{jk} are input-to-hidden layer interconnection weights, and w_{ij} are hidden-to-output layer interconnection weights. $\theta_{vm}, \theta_{wm},\ m = 1, 2, ...$ are threshold offsets. $N_i,\ N_h,\ N_o$ are neuron numbers of the input, hidden, and output layers.

The NN equation can be expressed in matrix format. Define $\varphi(x) = [1\ \varphi(x_1)\ \varphi(x_2)\ ...\ \varphi(x_{N_i})]^T$, $y = [y_1\ y_2\ ...\ y_{N_o}]^T$ and weight matrices $W^T = [w_{ij}]$, $V^T = [v_{jk}]$. Including "1" in $\varphi(x)$ allows one to include the threshold vector $[\theta_{v1}\ \theta_{v2}\ ...\ \theta_{vN_h}]^T$ as the first column of V^T, so V^T contains both the weights and thresholds of the input-to-hidden layer connections. If $z = [z_1\ z_2\ ...]^T$ is a vector we define $\sigma(z) = [\sigma(z_1)\ \sigma(z_2)\ ...]^T$. Including "1" as the first term in the vector $\sigma(V^T\varphi(x))$ allows one to incorporate the thresholds θ_{wi} as the first column of W^T. Then we have

$$y = W^T\sigma(V^T\varphi(x)) \tag{2}$$

Note that any tuning of W and V includes tuning of the thresholds as well.

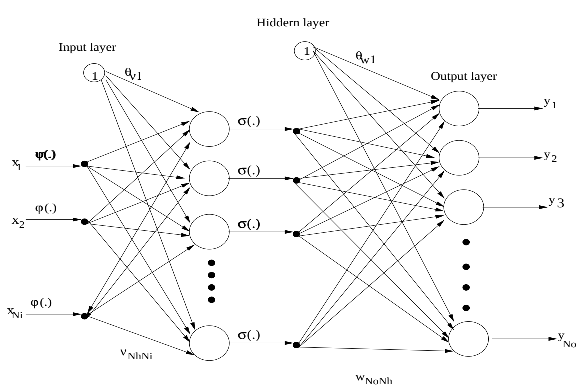

Figure 1: Three-layer Neural Network

For control purpose, NN must satisfy the function approximation property. It has been proved that given a sufficient number of hidden layer neurons, the multilayer feedforward neural networks can represent arbitrary mapping [14]. This means that any smooth nonlinear function can be reconstructed using neural networks. Let $f(x)$ be a general smooth function from R^n to R^m. For some number of hidden layer neurons N_h, there exist constant weights and thresholds such that

$$f(x) = W^T\sigma(V^T\varphi(x)) + \epsilon(x), \tag{3}$$

where $N_i = n,\ N_o = m$, and $\epsilon(x)$ is a NN functional reconstruction error vector which is bounded, i.e. $\|\epsilon\| < \epsilon_m$. ϵ_m can be sufficiently small by selecting N_h to be large enough.

The estimate of $f(x)$ is given by

$$\hat{f}(x) = \hat{W}^T\sigma(\hat{V}^T\varphi(x)), \tag{4}$$

where $\hat{W},\ \hat{V}$ are estimates of the ideal neural network weights W, V. We can get $\hat{W}$ and $\hat{V}$ through the weight tuning algorithms.

The activation function can be chosen as sigmoid, i.e.,

$$\sigma(z) = \varphi(z) = \frac{1}{1 + e^{-\alpha z}}. \tag{5}$$

The derivative of the sigmoid activation function is

$$\dot{\sigma}(z) = \alpha\sigma(z)(1 - \sigma(z)). \tag{6}$$

Clearly, all the activation function and its derivative are bounded.

2.2 Manipulator dynamics

The dynamics of an n-link manipulator expressed in the Lagrange form[15] is

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau, \tag{7}$$

where $q(t)$ is the joint variable vector, $D(q)$ is the inertia matrix, $C(q,\dot{q})$ is the coriolis/centripetal matrix, $G(q)$ is the gravity vector, and $F(\dot{q})$ is the friction. The unknown bounded disturbances are τ_d and the control input torque is $\tau(t)$.

Given a desired trajectory $q_d(t)$ the tracking error is

$$e(t) = q_d(t) - q(t). \tag{8}$$

The filtered tracking error is

$$r = \dot{e} + \Lambda e, \tag{9}$$

where $\Lambda = \Lambda^T$ is a positive definite matrix.

Let $q_r = \dot{q}_d + \Lambda e$. Differentiating $r(t)$ and using equation (7), we have

$$D\dot{r} = -Cr - \tau + f + \tau_d, \tag{10}$$

where the manipulator function is

$$f(x) = D(q)(\dot{q_r}) + C(q, \dot{q})q_r + G(q) + F(\dot{q}). \tag{11}$$

Manipulator function input x can be selected as: $x = [\dot{q_r}^T \ q_r^T \ q^T \ \dot{q}^T \ sgn(\dot{q})^T]^T$, where the signum function reflects the friction terms. Then the manipulator function can be approximated using neural networks.

The following properties of manipulator dynamics will be used later in this paper.
Property 1: D(q) is a positive definite symmetric matrix bounded by $d_1 I \leq D(q) \leq d_2 I$, where d_1 and d_2 are known positive constants, I denotes the identity matrix.
Property 2: The matrix $\dot{D} - 2C$ is skew-symmetric.
Property 3: The unknown disturbance satisfies $\|\tau_d\| < b_{\tau d}$, $b_{\tau d}$ is a known positive constant.

3 NN control design

In this section we present the NN controller with control structure as shown in Figure 2. This NN controller is different from that by Lewis et al. [10] in two ways. One is that the proposed scheme uses input activation function in the NN and the other is that a different weight tuning algorithm is used in our NN learning. With these differences, we are able to show that the tracking error bound depends on only the neural network approximation error bound, the system disturbance bound, as well as a control design parameter. We also use the mean value theory, which is firstly used by Song et al. [13] to tackle the nonlinear problem of the activation function, to effectively show the convergence of the tracking error in the robotic system.

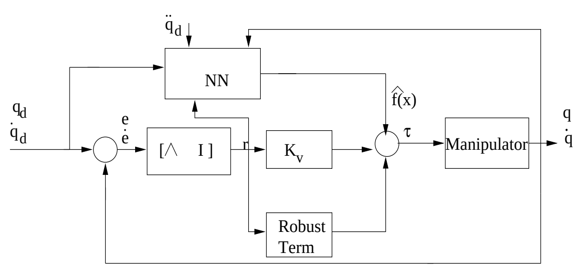

Figure 2: NN control robotic system structure

Some assumptions are used in the derivation.
Assumption 1: The ideal weights of NN are bounded as $\|V\|_F \leq V_m$ and $\|W\|_F \leq W_m$ with V_m and W_m known positive values.
Assumption 2: The desired position, velocity, and acceleration trajectories, i.e., $q_d, \ \dot{q}_d, \ \ddot{q}_d$, are bounded.

We propose the following control input as

$$\tau = \hat{f}(x) + K_v r - v. \tag{12}$$

In the above equation $\hat{f}(x)$ is the estimate of the manipulator function (11) using NN, $K_v r$ is the conventional proportional-plus-derivative(PD) term, where $K_v = K_v^T > 0$ is the gain matrix, and $v(t)$ is the function that provides robustness as shown in equation (16).

The closed-loop filtered error dynamics become

$$\begin{aligned} D\dot{r} = &- (K_v + C)r + W^T \sigma(V^T \varphi(x)) \\ &- \hat{W}^T \sigma(\hat{V}^T \varphi(x)) + \epsilon + \tau_d + v. \end{aligned} \tag{13}$$

Define the weight errors as $\tilde{W} = W - \hat{W}, \ \ \tilde{V} = V - \hat{V}$ and let $\hat{\sigma} = \sigma(\hat{V}^T \varphi(x)), \ \ \tilde{\sigma} = \sigma - \hat{\sigma}$.
Adding and subtracting $W^T \hat{\sigma}$ in equation (13), we have

$$D\dot{r} = -(K_v + C)r + \tilde{W}^T \hat{\sigma} + W^T \tilde{\sigma} + \epsilon + \tau_d + v. \tag{14}$$

Let

$$b_r = \frac{\epsilon_m + b_{\tau d}}{K_{vmin}} \tag{15}$$

where K_{vmin} is the minimum singular value of K_v. Now we can state our main results in the following theorem.

***Theorem* 1** *Consider the robotic system* (7) *under control* (12) *with the robustifying term*

$$\begin{aligned} v(t) = &- K_1(\|\hat{W}\| + W_m) sgn(r_\Delta) \\ &- K_2(\|\hat{V}\| + V_m) sgn(r_\Delta), \end{aligned} \tag{16}$$

where K_1,K_2 are positive constants, and

$$r_\Delta = \begin{cases} 0 & \|r\| \leq b_r \\ r & \|r\| \ > b_r, \end{cases} \tag{17}$$

Let NN weight tuning algorithm be given as

$$\dot{\hat{W}} = Q_1 \hat{\sigma} r_\Delta^T - Q_1 \dot{\hat{\sigma}} \hat{V}^T \varphi(x) r_\Delta^T \tag{18}$$

$$\dot{\hat{V}} = Q_2 \varphi(x) (\dot{\hat{\sigma}}^T \hat{W} r_\Delta)^T \tag{19}$$

where Q_1 and Q_2 are learning gain matrices which can be any constant positive definite matrices. If the assumptions 1 and 2 are valid, then the tracking error $r(t)$ will converge into the region $\|r\| \leq b_r$ and the weight estimates $\hat{W}$, $\hat{V}$ are uniformly bounded.

Proof:
The Lyapunov function is selected as

$$J = \frac{1}{2} r_\Delta^T D r_\Delta + \frac{1}{2} tr(\tilde{W}^T Q_1^{-1} \tilde{W}) + \frac{1}{2} tr(\tilde{V}^T Q_2^{-1} \tilde{V}). \tag{20}$$

When $\|r\| \le b_r$, $\dot{J} = 0$.

Otherwise, $r_\Delta = r$ and

$$\dot{J} = r^T D\dot{r} + \frac{1}{2} r^T \dot{D} r + tr(\tilde{W}^T Q_1^{-1} \dot{\tilde{W}}) + tr(\tilde{V}^T Q_2^{-1} \dot{\tilde{V}}). \tag{21}$$

Substituting the error system (14) will yield

$$\begin{aligned} \dot{J} = & - r^T K_v r + \frac{1}{2} r^T (\dot{D} - 2C) r + tr(\tilde{W}^T Q_1^{-1} \dot{\tilde{W}}) \\ & + tr(\tilde{V}^T Q_2^{-1} \dot{\tilde{V}}) + r^T (\tilde{W}^T \hat{\sigma} + W^T \tilde{\sigma} + \epsilon + \tau_d + v). \end{aligned} \tag{22}$$

The disposal of the term $W^T \tilde{\sigma}$ is a key point in the proof. Here we use the mean value theory to tackle this problem. Since the activation function $\sigma(.)$ in equation (5) is nondecreasing, there exists a series positive numbers $\mu_1(.),\ \mu_2(.), ..., \mu_{N_h}(.)$ with $0 \le \mu_i \le \alpha/4$, where $\alpha/4$ is the maximum value of the derivative of the activation function, such that the following equation holds

$$\tilde{\sigma} = \sigma(V^T \varphi(x)) - \sigma(\hat{V}^T \varphi(x)) = \mu \tilde{V}^T \varphi(x), \tag{23}$$

where $\mu(.) = diag\{\mu_1(.),\ \mu_2(.), ..., \mu_{N_h}(.)\}$.

From equation(23), we have

$$\begin{aligned} W^T \tilde{\sigma} = & W^T \mu \tilde{V}^T \varphi(x) \\ = & \hat{W}^T \dot{\hat{\sigma}} \tilde{V}^T \varphi(x) + \tilde{W}^T \dot{\hat{\sigma}} \tilde{V}^T \varphi(x) \\ & + W^T (\mu - \dot{\hat{\sigma}}) \tilde{V}^T \varphi(x) \\ = & \hat{W}^T \dot{\hat{\sigma}} \tilde{V}^T \varphi(x) + \tilde{W}^T \dot{\hat{\sigma}} V^T \varphi(x) \\ & - \tilde{W}^T \dot{\hat{\sigma}} \hat{V}^T \varphi(x) + W^T (\mu - \dot{\hat{\sigma}}) \tilde{V}^T \varphi(x). \end{aligned} \tag{24}$$

Note that $\dot{D} - 2C$ is skew symmetry and using equation (24) and the NN weight tuning algorithm(18)(19), we have

$$\begin{aligned} \dot{J} = & - r^T K_v r + tr(\tilde{W}^T [Q_1^{-1} \dot{\tilde{W}} + \hat{\sigma} r^T - \dot{\hat{\sigma}} \hat{V}^T \varphi r^T]) \\ & + tr(\tilde{V}^T [Q_2^{-1} \dot{\tilde{V}} + \varphi r^T \hat{W}^T \dot{\hat{\sigma}}]) + r^T (\epsilon + \tau_d) + r^T v \\ & + r^T [\tilde{W}^T \dot{\hat{\sigma}} V^T \varphi + W^T (\mu - \dot{\hat{\sigma}}) \tilde{V}^T \varphi] \\ = & - r^T K_v r + r^T (\epsilon + \tau_d) + r^T v \\ & + r^T [\tilde{W}^T \dot{\hat{\sigma}} V^T \varphi + W^T (\mu - \dot{\hat{\sigma}}) \tilde{V}^T \varphi] \\ \le & - K_{vmin} \|r\|^2 + \|r\| (\epsilon_m + b_{\tau d}) \\ & - \|r\| [K_1 (\|\hat{W}\| + W_m) + K_2 (\|\hat{V}\| + V_m)] \\ & + \|r\| (c_1 V_m \|\tilde{W}\| + c_2 W_m \|\tilde{V}\|), \end{aligned} \tag{25}$$

where $c_1,\ c_2$ are positive constants and $\|\dot{\hat{\sigma}}\| \|\varphi\| \le c_1$,$\|\mu - \dot{\hat{\sigma}}\| \|\varphi\| \le c_2$.

Choose $K_1 > c_1 V_m$ and $K_2 > c_2 W_m$ and we have

$$\dot{J} \le -\|r\| [K_{vmin} \|r\| - (\epsilon_m + b_{\tau d})]. \tag{26}$$

From the above equation, we can conclude that when $\|r\| > b_r$, $\dot{J} < 0$. Finally, the above results show that r_Δ, $\tilde{W}$, and $\tilde{V}$ are uniformly bounded, and tracking error $r(t)$ will converge into the region $\|r\| \le b_r$. □

Remark 1: Using the proposed control scheme, the bound b_r of the filtered tracking error r is completely determined by the NN approximation error bound, the system disturbance bound, and the minimum singular value of gain matrix K_v. The error bound b_r can be reduced when control gain matrix K_v is "increased".

Remark 2: The filtered tracking error bound b_r is not related to the weight estimation error. This is a major improvement from the results in [10] where there is a trade-off between the relative eventual magnitudes of the tracking error and the weight estimation error.

4 Simulation Results

A two-link manipulator is used to verify the proposed control scheme. Dynamics of the manipulator has the form of (7) and the details are given in [15]. Manipulator parameters are selected as $m_1 = 1\ kg$, $m_2 = 2.3\ kg$, $l_1 = 1\ m$, and $m_1 = 1\ m$. The desired trajectories are given as $q_{1d}(t) = sin(t)$, $q_{2d}(t) = cos(t)$. Initial values of $q_1,\ q_2,\ \dot{q}_1,\ \dot{q}_2$ are selected as zeros. The uncertain part of the manipulator, i.e., the friction and disturbance are given as $F(\dot{q}) = [\ 0.1sgn(\dot{q}_1)\ \ 0.1sgn(\dot{q}_2)]^T$, $\tau_d = [\ 0.2sin(q_1 + q_2)\ \ 0.2cos(q_1 + q_2)]^T$.

Controller (12) with parameters $K_v = diag\{25,\ 25\}$, $\Lambda = diag\{6,\ 6\}$, $Q_1 = diag\{15, ..., 15\}$, $Q_2 = diag\{10, ..., 10\}$. Neural network input x is selected as $x = [\dot{q}_r{}^T\ q_r^T\ q^T\ \dot{q}^T\ sgn(\dot{q})^T]^T$. So the neural network we used has ten inputs and two outputs. The number of hidden layer nuerons is selected 12. In the first control period, only the PD controller is used, so we set the initial values of NN weights and thresholds as zeros.

Simulation results are shown in Figures $3-5$. Figure 3 and Figure 4 show the position trajectories using the back propagation and our proposed weight tuning algorithms respectively. The tracking response of our method shows an improvement compared with that of the back propagation algorithm. After a short time's on-line training, the NN can work well as the manipulator function. For comparison, Figure 5 shows the position trajectories using PD controller, which can ensure the actual output bounded but with large error.

5 Conclusion

In this paper, a robust neural network control scheme is proposed for the trajectory tracking for robotic manip-

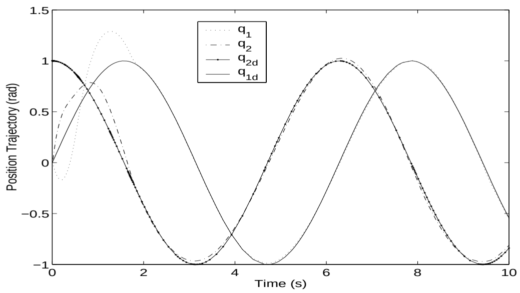

Figure 3: Desired and actual position trajectories using PD and NN controller with BP weight tuning algorithm

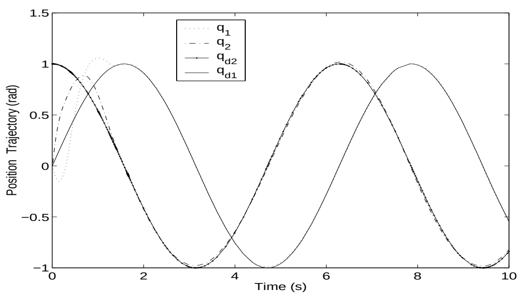

Figure 4: Desired and actual position trajectories using PD and NN controller with proposed weight tuning algorithm

ulators. The neural network is trained on-line without any *a priori* knowledge of the manipulator dynamics. The weight tuning algorithm in this paper has a small dead zone which guarantees the convergence and robustness of the system in the presence of bounded disturbances. The proposed control scheme shows that the bound of the filtered tracking error depends on only the NN approximation error bound, the disturbance bound, as well as the minimum singular value of gain matrix and that the filtered tracking error does not have to be related to the weight estimate errors. Finally the simulation results illustrate the effective performance of the proposed NN controller using a two-link manipulator.

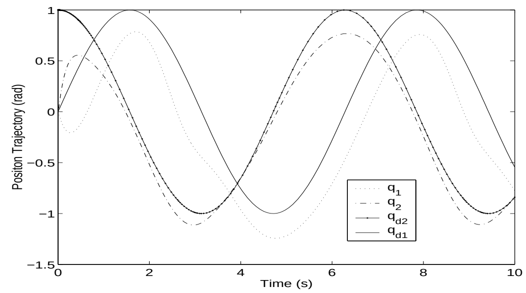

Figure 5: Desired and actual position trajectories only using PD controller.

References

[1] J.J. Craig, P. Hsu and S.S. Satry, "Adaptive Control of Mehcanical Manipulators," Int. J. Robotics Res., vol. 6, no. 3, pp.16-28, 1987.

[2] J.-J.E. Slotine and W. Li, "On the Adaptive Control of Robot Manipulators," Int. J. Robotics Res., vol. 6, no. 3, pp.49-59, 1987.

[3] S. Arimoto, "Learning Control Theroy for Robot Motion," Int. J. Adapt. Control Signal Process., vol. 4, no.6, pp. 534-564. 1990.

[4] J. S. Albus, "Data storage in the cerebellar model articulation controller(cmac)," J. Dyn. Sys. Meas. Control Trans., Vol. ASME 97, pp. 228-233, 1975.

[5] W. T. Miller, R. P. Hewes, F.H. Glanz and L.G. Kraft, "Real-time dynamic control of an industrial manipulator using a neural network-based learning control," IEEE Trans. Robotics and Automat., Vol. 6, pp. 1-9, 1990.

[6] M. Johnson and M. Leahy, "Adaptive model based neural network control," IEEE Int. Conf. on Robotics and Automation, pp1704-1709, Cincinatti, 1990.

[7] R. Carelli, E.F. Camacho, and D. Patino, "A neural network based feedforward adaptive controller for robots," IEEE Trans. on system, Man, and Cybernetics, Vol. 25, No. 9, pp.1281-1288, 1995.

[8] F. C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," IEEE Trans. on Automatic Control, Vol. 40, pp. 791-801, 1995.

[9] F. L. Lewis, K. Liu and A. Yesildirek, "Neural Net Robot Controller with Guaranteed Tracking Performance," IEEE Trans. on Neural Networks, Vol. 6, pp. 703-715, 1995.

[10] F. L. Lewis, A. Yesildirek and K. Liu, "Multilayer Neural Net Robot Controller with Guaranteed Tracking Performance," IEEE Trans. on Neural Networks, Vol. 7, pp. 388-399, 1996.

[11] Y.Jin, T. Pipe, and A. Winfield, "Stable Neural Network Control for Manipulators," Intelligent Systems Engineering, pp. 213-222, 1993.

[12] S. S. Ge and C. C. Hang, "Direct Adaptive Neural Network Control of Robots," Int. J. Syst. Sci., vol. 27, no. 6, pp.533-542, 1996.

[13] Q. Song, J. Xiao, and Y.C. Soh,"Robust Backpropagation Training Algorithm for Multilayered Neural Tracking Controller," IEEE Trans. on Neural Networks, pp. 1133-1141, 1999.

[14] K. Hornik, M. Stinchombe and H. White, "Multilayer feedforward neural networks are universal approximators," Neural Networks, Vol. 2, pp. 359-366, 1989.

[15] J.J. Craig, Introduction to Robotics: Mechanics and Control, Addison-Wesley Publishing Company, 1986.