Neural Network Impedance Force Control of Robot Manipulator

Seul Jung*and T.C. Hsia
Robotics Research Laboratory
Department of Electrical and Computer Engineering
University of California, Davis
Davis, CA 95616
e-mail:hsia@ece.ucdavis.edu and jung@hanbat.chungnam.ac.kr

Abstract

Performance of impedance controller for robot force tracking is affected by the uncertainties in both the robot dynamic model and environment stiffness. The purpose of this paper is to improve the controller robustness by applying the neural network(NN) technique to compensate for the uncertainties in the robot model. NN control techniques are applied to two impedance control methods : torque-based and position-based impedance control, which are distinguished by the way of the impedance functions being implemented. A novel error signal is proposed for the neural network training. In addition, a trajectory modification algorithm is developed to determine the reference trajectory when the environment stiffness is unknown. The robustness analysis of this algorithm to force sensor noise and inaccurate environment position measurement is also presented. The performances of the two NN impedance control schemes are compared by computer simulations. Simulation results based on a three degrees-of-freedom robot show that highly robust position/force tracking can be achieved under the presence of large uncertainties and force sensor noise.

1 Introduction

When the robot manipulator is in contact with a constraining surface(called the environment), force is generated between the end effector and the environment. The amount of forces exerted by the robot to the environment depends on how much the end-effector position is physically constrained by the environment to reach its goal position. Hence, the contact force can be regulated by appropriate control of the end-effector position. One most frequently used approach is to adjust the robot end-effector position in response to the sensed contact force in such a way that a target impedance relationship is satisfied. This is the well known impedance force control concept [1]. This approach is different from the hybrid force control technique which controls the position and the force separately in their own controllable directions [2].

There are two ways to implement the impedance force control concept. One is called torque-based impedance force control [1]. In this approach, the torque informations are assumed to be available so that the impedance function can implicitly be implemented inside the control loop. Another is called position-based impedance force control [3, 4]. In this approach, informations on control input torque are not available so that the impedance function is explicitly implemented outside the position control loop. Thus, the inner position loop is controlled by the trajectory command input that is adjusted by position correction output from impedance function of the outer loop. The impedance function is realized by the combination of inner loop and the outer loop control. The latter impedance control approach has the advantage of easy implementation on an existing position-controlled robot system [3]. However, the position controller has to be highly accurate in order to achieve accurate force control.

In both control methods, design of impedance control law is straightforward if the models of the robot manipulator and the environment are known with high precision [5]. Uncertainties in these models degrade the system performance. In practice, the complete dynamic model of a robot is not known exactly, and the environment stiffness is also approximately known. Accurate knowledge of the environment stiffness is very important in that it determines the robot target trajectory in order to attain the desired contact force.

In the literature, it has been shown that neural networks(NN) are very effective in compensating for trajectory tracking in free space [6, 7, 8, 9, 10, 11]. Here

ever, both studies did not address the issue of environment uncertainties.

Recently, there have been many studies made to solve the problem of environment stiffness uncertainty such as by employing separate trajectory modification control loop using integral control [14], or by generating reference position based on force tracking errors using adaptive control [15, 16] or by adjusting controller gains using adaptive control based on force errors [17]. Although these techniques do work, they tend to increase the complexity of the system dynamics which require special attention to system stability.

The objective of this paper is to present a solution to impedance force control assuming that there are uncertainties in robot dynamics and environment stiffness. The proposed solution employs the sensed contact force and neural network technique in the controller design. Based on the well known impedance control concept, a new unified training signal is derived for an NN controller with a two-layer feedforward architecture. This training signal makes simpler control structure than that in [12]. In addition, here we are proposing a simple technique for reference trajectory modification without using any informations about environment stiffness and introducing complex dynamics to the system [18]. Thus the NN impedance controllers are able to provide the robust force tracking for arbitrarily varying unknown environment stiffness by compensating for uncertainties both in robot dynamics and environment.

2 Dynamic Equation of Robot manipulators

The dynamic equation of an n degrees-of-freedom manipulator in joint space coordinates are given by :

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + \tau_f(\dot{q}) = \tau - \tau_e \qquad (1)$$

where the vectors $q, \dot{q}, \ddot{q}$ are the joint angle, the joint angular velocity, and the joint angular acceleration, respectively; $D(q)$ is the n x n symmetric positive definite inertia matrix; $C(q,\dot{q})\dot{q}$ is the n x 1 vector of Coriolis and centrifugal torques; $g(q)$ is the n x 1 gravitational torques; τ_f is the n x 1 vector of actuator joint friction forces; τ is the n x 1 vector of actuator joint torques; τ_e is the n x 1 vector of external disturbance joint torques. For simplicity, let us denote $h(q,\dot{q}) = C(q,\dot{q})\dot{q} + g(q)$ so that (1) can be rewritten as

$$D(q)\ddot{q} + h(q,\dot{q}) + \tau_f(\dot{q}) = \tau - \tau_e \qquad (2)$$

The relationship between joint velocity and Cartesian space velocity can be expressed as

$$\dot{X} = J(q)\dot{q} \qquad (3)$$

when the Jacobian matrix $J(q)$ represents the relationship between virtual end-effector displacements and virtual joint displacements. By differentiating (3) the Cartesian acceleration term can be found as

$$\ddot{X} = J(q)\ddot{q} + \dot{J}\dot{q} \qquad (4)$$

Then the equation of robot motion in joint space can also be represented in Cartesian space coordinates by the relationship

$$\ddot{q} = J(q)^{-1}(\ddot{X} - \dot{J}\dot{q}) \qquad (5)$$

Substituting (5) into (2) yields

$$D(q)J^{-1}(\ddot{X} - \dot{J}\dot{q}) + h(q,\dot{q}) + \tau_f(\dot{q}) = \tau - \tau_e \qquad (6)$$

The actuator forces are related to the joint torques of the actuators through the Jacobian of the mechanism, i.e.

$$\tau = J^T F \qquad (7)$$

The robot dynamic equation now becomes

$$J^{T^{-1}}DJ^{-1}(\ddot{X} - \dot{J}\dot{q}) + J^{T^{-1}}(h + \tau_f(\dot{q})) = F - F_e \qquad (8)$$

Finally, we have the robot dynamic equation model including disturbance force in Cartesian space as

$$D^*\ddot{X} + h^* + F_f^* = F - F_e \qquad (9)$$

where $D^* = J^{T^{-1}}DJ^{-1}$, $h^* = J^{T^{-1}}h - D^*\dot{J}J^{-1}\dot{X}$ and $F_f^* = J^{T^{-1}}\tau_f$. The robot dynamic equation (9) represents a highly nonlinear, coupled, and multi-input multi-output system. In most practical cases, the model is not exactly known. Thus only nominal estimates of the model are available for controller design.

3 Torque-Based Impedance Force Control

Direct application of position controllers designed for free space robot motion usually results in instability when robot hand is in contact with a constraining environment because these controllers ignore the fundamental dynamic relationship of interaction between the robot and the environment. Impedance control allows us to specify the robot stiffness for a given task under contact with the environment. The interaction between the environment and the end-point

of the robot in terms of a desired mechanical stiffness, called impedance is regulated. Thus the impedance control regulates the relationship between force and position by setting suitable gains of impedance parameters. Figure 1 shows the torque-based impedance control structure. The torque-based impedance control law F is

$$F = \hat{D}^* U + \hat{h}^* + F_e \tag{10}$$

where $\hat{D}^*, \hat{h}^*$ are estimates of D^*, h^* and F_e is the exerted force on the environment. And U is given by

$$U = \ddot{X}_r + M^{-1}[B(\dot{X}_r - \dot{X}) + K(X_r - X) - F_e] \tag{11}$$

where M B and K are n x n symmetric positive definite desired inertia, damping, and stiffness gain matrices, respectively and X_r is the reference end-point trajectory. Desired contact force F_e can be achieved by appropriately designing X_r as shown in (46). The control system is depicted in Figure 1. Combining (9),(10), and (11) yields the closed loop tracking error dynamic equation

$$\ddot{E} + M^{-1}(B\dot{E} + KE - F_e) = \hat{D^*}^{-1}(\Delta D^* \ddot{X} + \Delta h^* + F_f^*) \tag{12}$$

where $\Delta D^* = D^* - \hat{D^*}, \Delta h = h^* - \hat{h^*}$, and $E = (X_r - X)$. In the ideal case where $\Delta D^* = \Delta h^* = 0$, and $F_f^* = 0$, the closed loop robot behavior satisfies the target impedance relationships

$$F_e = M\ddot{E} + B\dot{E} + KE \tag{13}$$

4 Position-Based Impedance Control

In position-based force control, the impedance controller is added to form an additional control loop around the position controlled manipulator as shown in Figure 2. The position command X_c is determined by $X_c = X_r - X_a$ where X_a is related to the sensed contact force F_e. The desired relationship between X_a and F_e can be expressed as the impedance function

$$F_e = M\ddot{X}_a + B\dot{X}_a + KX_a \tag{14}$$

In frequency domain the impedance function is represented by

$$X_a(s) = [Ms^2 + Bs + K]^{-1} F_e(s) \tag{15}$$

The reference trajectory X_r and the robot position command X_c are related by

$$X_r = X_c + X_a \tag{16}$$

Thus, the position command X_c is modified by X_a from impedance relationship with F_e. In order to regulate F_e to track a desired force F_d, the reference trajectory X_r should be appropriately designed [5].

The control law for implementing the above impedance control concept is

$$F = \hat{D}^* U + \hat{h}^* \tag{17}$$

and

$$U = \ddot{X}_c + K_D(\dot{X}_c - \dot{X}) + K_P(X_c - X) \tag{18}$$

where $\hat{D}^*$ and $\hat{h}^*$ are nominal values for D^* and h^*, K_D and K_P are the desired positive definite diagonal matrices for damping and stiffness gains. Combining (9),(17), and (18) yields the closed loop position tracking error equation

$$\ddot{E} + K_D\dot{E} + K_P E = \hat{D^*}^{-1}[\Delta D^* \ddot{X} + \Delta h^* + F_f^* + F_e] + \ddot{X}_a + K_D\dot{X}_a + K_P X \tag{19}$$

where $\Delta D^* = D^* - \hat{D^*}$, and $\Delta h = h^* - \hat{h^*}$. The term $\hat{D^*}^{-1}[\Delta D^* \ddot{X} + \Delta h^* + F_f^* + F_e]$ is the robot dynamic uncertainty. In the ideal case where $\Delta D^* = \Delta h^* = F_f^* = 0$, the closed loop robot motion satisfies

$$\ddot{E} + K_D\dot{E} + K_P E = \ddot{X}_a + K_D\dot{X}_a + K_P X_a \tag{20}$$

By multiplying M in (20) and defining $B = MK_D$ and $K = MK_P$ the desired impedance relation (20) can be satisfied by

$$F_e = M\ddot{E} + B\dot{E} + KE \tag{21}$$

which is same as (13). Letting $F_e = K_e(X - X_e)$ where K_e is the environment stiffness constant and X_e is the environment position, we have

$$M\ddot{E} + B\dot{E} + KE = K_e(X - X_e) \tag{22}$$

Replacing $X - X_e$ with $X_r - X_e - X_r + X$ in (22) yields

$$M\ddot{E} + B\dot{E} + (K + K_e)E = K_e(X_r - X_e) \tag{23}$$

We note that the damping condition of (23) varies for different environment stiffnesses. In order to assure that the system behaves over-damped, impedance gains M, B and K should be adjusted in accordance with environment stiffness K_e. For example, the damping gain B should be increased to suppress the overshoot at initial contact of the robot with the environment if the environment is stiff. We show in the simulations that different impedance gains for different environment stiffnesses can be appropriately selected.

Since there are always uncertainties in the robot dynamic model such that $\Delta D^*, \Delta h^*$ and F_f^* are not identically zero, the ideal target impedance relationships (13) and (21) can not be achieved in general. The actual system performance is governed by (12) or (19) which has an unpredictable dynamics behavior. Thus force control is not robust. To improve robustness, NN controller can be introduced to generate additional signals to compensate for the disturbances due to model uncertainties.

5 Torque-Based NN Impedance Controller Scheme

In this section, we present torque-based NN impedance controller(TBNNIC) design to achieve disturbance rejection for an impedance force control system. The idea shown in Figure 3 is that the NN output Φ_τ cancels out the uncertainties caused by inaccurate robot model in the inverse dynamic model controller. The dimension of NN output depends upon the number of robot joints.

A new control law can be given by adding compensating signals to the control input signal

$$F(t) = \hat{D}^*(U + \Phi_\tau) + \hat{h^*} + F_e \tag{24}$$

where U is same as (11). Combining (24) with (9) yields the corresponding closed loop error system as

$$\ddot{E} + M^{-1}(B\dot{E} + KE - F_e) = \hat{D^*}^{-1}(\Delta D^* \ddot{X} + \Delta h^* + F_f^*) - \Phi_\tau \tag{25}$$

Since the control objective is to generate Φ_τ to reduce v to zero in (25), we therefore propose here to use v

$$v = \ddot{E} + M^{-1}(B\dot{E} + KE - F_e) \tag{26}$$

as the new error signal for training the NN. The ideal value of Φ_τ at $v = 0$ then is same as the uncertainties

$$\Phi_\tau = \hat{D^*}^{-1}(\Delta D^* \ddot{X} + \Delta h^* + F_f^*) \tag{27}$$

Clearly minimizing the error signal v allows us to achieve ideal force control directly. When $v = 0$, we achieve the ideal impedance function (13). However, the use of error signal as suggested in [7] requires on-line computation of the nominal robot model $F_m = \hat{D}^* \ddot{X} + \hat{h}^*$ which is far more time consuming than computing v in (26) as proposed. The training signal proposed in [12] does not satisfy both position and force tracking simultaneously since the error for training is formed in either position error or in force error.

6 Position-Based NN Impedance Control Scheme

Here we present the position-based NN impedance control(PBNNIC) scheme shown in Figure 4 that is implemented based on the position-controlled robot system. The NN has the same goal of canceling those uncertainties by compensating at trajectory not at the control input U. In the paper [19] the authors have shown the comparison study of compensating at different locations and demonstrated that compensation at the reference trajectory is better than that at other locations in free space motion of the robot manipulators. Same behaviors are expected to happen in force control.

The proposed control law for F is

$$F = \hat{D}^* U + \hat{h}^* \tag{28}$$

and U is

$$U = \ddot{X}_c + K_D(\dot{X}_c - \dot{X}) + K_P(X_c - X) - M^{-1}F_e \tag{29}$$

We note that F_e is introduced in U so that the impedance controller is outside the robot position control system.

The idea here is to move compensating location from the control input signal to reference trajectory so that compensations can be done by modifying the reference trajectory indirectly instead of modifying the control input signal directly. In this way, the robot is less sensitive to the disturbance injection. Substituting $X_c = X_r + \phi_p, \dot{X}_c = \dot{X}_r + \phi_d, \ddot{X}_a = \ddot{X}_r + \phi_a$ into U in (29) yields

$$U = \ddot{X}_r + \phi_a + K_D(\dot{X}_r - \dot{X} + \phi_d) + K_P(X_r - X + \phi_p) - M^{-1}F_e \tag{30}$$

Combining (28)(29)(30) with (1) yields the corresponding closed loop error system as

$$M\ddot{E} + B\dot{E} + KE - F_e = M(\hat{D^*}^{-1}(\Delta D^* \ddot{X} + \Delta h^* + F_e + F_f^*) - \Phi_P) \tag{31}$$

where $\Phi_P = \phi_a + K_D\phi_d + K_P\phi_p$. In view of (31), the target impedance

$$F_e = M\ddot{E} + B\dot{E} + KE \tag{32}$$

can be realized if Φ_P is properly defined which is same as (21). Thus the NN controller design objective is to generate ϕ_a, ϕ_d, ϕ_p such that Φ_P satisfies $\hat{D^*}^{-1}(\Delta D^* \ddot{X} + \Delta h^* + F_e + F_f^*)$. We therefore propose here an NN training signal v to be

$$v = \ddot{E} + M^{-1}(B\dot{E} + KE - F_e) \tag{33}$$

which is same as that of the torque-based. It is clear that the ideal value of Φ_P at $v = 0$ is

$$\Phi_P = \hat{D^*}^{-1}(\Delta D^* \ddot{X} + \Delta h^* + F_e + F_f^*) \tag{34}$$

which is slightly different from Φ_τ. We note that there are more uncertainties in Φ_P since $\Phi_P = \Phi_\tau + \hat{D^*}^{-1} F_e$. The term $\hat{D^*}^{-1} F_e$ comes from the control law (28) by not canceling F_e since we assume that the position-based impedance control is developed from the position controlled robot system. Then, minimizing v allows us to achieve the ideal force control objective (15).

7 Neural Network Compensator Design

The two-layer feedforward neural network structure shown in Figure 5 is used as the compensator. It is composed of an input buffer, a non-linear hidden layer, and a linear output layer. The inputs $X = [X_r^T(t)\ X_r^T(t-1)\ X_r^T(t-2)]^T$ for TBNNIC or $X = [X_r^T(t)\ X_r^T(t-1)\ X_r^T(t-2)\ F_e^T(t)]^T$ for PBNNIC are multiplied by weights w_{ij}^1 and summed at each hidden node. Then the summed signal at a node activates a nonlinear function $f(\cdot)$, called a sigmoid function, which is bounded in magnitude between -1 and 1:

$$f(\cdot) = \frac{1 - exp(-(\cdot))}{1 + exp(-(\cdot))} \tag{35}$$

The outputs from all $f(\cdot)$ are weighted by w_{jk}^2 and summed at each output node. Thus, the output ϕ_k at a linear output node can be calculated from its inputs as follows:

$$\phi_k = [\sum_{j=1}^{n_H} w_{jk}^2 (\frac{1 - exp(-(\sum_{i=1}^{n_I} x_i w_{ij}^1 + b_j^1))}{1 + exp(-(\sum_{i=1}^{n_I} x_i w_{ij}^1 + b_j^1))})] + b_k^2 \tag{36}$$

where n_I is the number of inputs, n_H is the number of hidden neurons, x_i is the ith element of input of X, w_{ij}^1 is the first layer weight between ith input and jth hidden neurons, w_{jk}^2 is the second layer weight between jth hidden neuron and kth output neuron, b_j^1 is a biased weight for jth hidden neuron and b_k^2 is a biased weight for kth output neuron. If n is the number of output neurons, the total number of weights w_T is $w_T = (n_I + 1)n_H + (n_H + 1)n$. The total number of neurons is $n_T = n_H + n$. Thus, our structure of TBNNIC shown in Figure 3 has $n_I = 9, n = 3, n_H = 6$ and $w_T = 81$. For PBNNIC shown in Figure 4, $n_I = 12, n = 9, n_H = 6$ and $w_T = 141$.

The weight updating law minimizes the objective function $\mathcal{J}$ which is a quadratic function of the training signal v.

$$\mathcal{J} = \frac{1}{2} v^T v \tag{37}$$

Differentiating equation(37) and making use of (25) or (31) yields the gradient of $\mathcal{J}$ as follows:

$$\frac{\partial \mathcal{J}}{\partial w} = \frac{\partial v^T}{\partial w} v = -\frac{\partial \Phi^T}{\partial w} v \tag{38}$$

in which the fact $\frac{\partial v^T}{\partial w} = -\frac{\partial \Phi^T}{\partial w}$ is used. The back-propagation update rule for the weights with a momentum term is

$$\Delta w(t) = \eta \frac{\partial \Phi^T}{\partial w} v + \alpha \Delta w(t-1) \tag{39}$$

where η is the update rate and α is the momentum coefficient. Specifically,

$$\Delta w_{ij}^1(t) = \eta s_j^1 (1 - s_j^1) x_i [\sum_{k=1}^{n} v_k w_{jk}^2] + \alpha \Delta w_{ij}^1(t-1) \tag{40}$$

$$\Delta w_{jk}^2(t) = \eta v_k s_j^1 + \alpha \Delta w_{jk}^2(t-1) \tag{41}$$

$$\Delta b_j^1(t) = \eta s_j^1 (1 - s_j^1) [\sum_{k=1}^{n} v_k w_{jk}^2] + \alpha \Delta b_j^1(t-1) \tag{42}$$

$$\Delta b_k^2(t) = \eta v_k + \alpha \Delta b_k^2(t-1) \tag{43}$$

$$s_j^1 = \frac{1}{1 + exp(-(\sum_{i=1}^{n_I} x_i w_{ij}^1 + b_j^1))} \tag{44}$$

$$s_k^2 = \sum_{j=1}^{n_H} s_j^1 w_{jk}^2 + b_k^2 \tag{45}$$

where s_j^1 is the output of jth hidden neuron and s_k^2 is the output of kth output neuron.

8 Trajectory Modification Technique for Compensating Environment Stiffness

Regulating F_e to track a desired contact force F_d, X_r must be appropriately designed based on F_d, X_e, K, and K_e as follows:

$$x_r = x_e + \Delta x_r \qquad \Delta x_r = \frac{f_d}{k_{eff}} \tag{46}$$

where x_r, x_e, f_d represent elements of X_r, X_e, F_d respectively, and k_{eff} is the combined stiffness of the environment and the robot [5]

$$k_{eff} = \frac{k k_e}{k + k_e} \tag{47}$$

in which k and k_e are corresponding elements of K and K_e. So when the environment stiffness uncertainty k_e is known, it is easy to calculate the reference trajectory x_r from (46) and (47). In practical case where k_e can not be accurately measurable, x_r may be inaccurate resulting poor force control performance.

In this section, we are proposing a simple and effective technique to deal with the unknown environment stiffness. Our proposed technique is to identify the environment stiffness k_e through the relationship

$$f_e = k_e(x - x_e) \tag{48}$$

Solving for k_e yields

$$k_e = \frac{f_e}{x - x_e} \tag{49}$$

where f_e is the measured contact force obtained from a wrist force sensor. In practice (49) is only a close estimate of k_e as f_e may be subjected to measurement error. Substituting (49) into (47) yields

$$k_{eff} = \frac{k\frac{f_e}{(x-x_e)}}{k + \frac{f_e}{(x-x_e)}} = \frac{kf_e}{k(x-x_e) + f_e} \tag{50}$$

Combining (50) with (46) yields

$$x_r = x_e + f_d[\frac{k(x-x_e) + f_e}{kf_e}] \tag{51}$$

This is the desired result of designing x_r as shown in Figure 6, we see that the reference trajectory x_r is modified from Δx_r based on f_d, f_e, k, x, and x_e, all of which are assumed to be known at all times. We note however that (51) is invalid when $f_e = 0$. This case represents the moment just before the robot is exerting force on the environment and $x = x_e$. So we see that the trajectory modification term Δx_r

$$\Delta x_r = f_d[\frac{k(x-x_e) + f_e}{kf_e}] \tag{52}$$

is indefinite at $f_e = 0$ when $x - x_e = 0$. To take care of this case, we determine the limits of Δx_r for $f_e = 0$ at $x - x_e = 0$ using $L'Hopital$'s rule which yields $\Delta x_r = \frac{f_d}{k}$. Therefore (51) is modified as

$$x_r = \begin{cases} x_e + \frac{f_d}{k} & \text{if } f_e = 0 \\ x_e + f_d[\frac{k(x-x_e)+f_e}{kf_e}] & \text{if } f_e \neq 0 \end{cases} \tag{53}$$

This result is applicable to robot position control for both contact and non contact phase. Simulations are carried out to demonstrate the performance of the proposed design.

8.1 Force sensor noise analysis

It is important to examine the practical question as to how robust the proposed result (53) is when force sensor noise is present. Let n_f be the zero mean random force sensor noise. Adding n_f to (52) yields

$$\Delta x_r' = f_d\frac{k(x-x_e) + f_e + n_f}{k(f_e + n_f)} \tag{54}$$

Calculating the ratio $\frac{\Delta x_r}{\Delta x_r'}$ yields

$$\frac{\Delta x_r}{\Delta x_r'} = \frac{f_e + n_f}{f_e}(\frac{k(x-x_e) + f_e}{k(x-x_e) + f_e + n_f}) \tag{55}$$

Substituting the relationship $x - x_e = \frac{f_e}{k_e}$ in (48) into (55) yields

$$\frac{\Delta x_r}{\Delta x_r'} = \frac{f_e + n_f}{f_e}(\frac{1}{1 + \frac{n_f}{\frac{kf_e}{k_e} + f_e}}) \tag{56}$$

For simplicity, let $k' = \frac{k}{k_e}$ and $f' = \frac{n_f}{f_e}$. We can show that (56) becomes

$$\frac{\Delta x_r}{\Delta x_r'} = \frac{k'f'}{k' + f' + 1} + 1 \tag{57}$$

This result exhibits the difference between Δx_r and $\Delta x_r'$ due to force measurement noise n_f. The difference depends on the ratios k' and f'. For most practical cases where the noise n_f is small and environment is sufficiently stiff, both k' and f' are small so that $\Delta x_r' \approx \Delta x_r$ It also shows that $\frac{\Delta x_r}{\Delta x_r'} \to 1$ as $k_e \to \infty$ (a rigid environment). So in the case of $k_e >> k$, noise n_f has very little effect on the accuracy of $\Delta x_r'$, and force tracking is sufficiently accurate. In general we can calculate the ratio $\frac{\Delta x_r}{\Delta x_r'}$ for a given k_e and n_f. For example, if $k_e = 1000$ for a non-rigid environment, $k = 100$ and force sensor noise is limited to $\pm 10\%$ of the force measurement, then $k' = 0.1$ and $f' = \pm 0.1$, and the ratio $\frac{\Delta x_r}{\Delta x_r'} = 1.00833 \sim 0.99$. This represents an error of less than 1% in $\Delta x_r'$.

8.2 Environment Position inaccuracy effect

Let δx_e be the measurement error of the environment position. Adding δx_e to (52) yields (assume that the noise n_f is absent)

$$\Delta x_r' = f_d\frac{k(x + \delta x_e - x_e) + f_e}{kf_e} \tag{58}$$

Calculating the ratio $\frac{\Delta x_r}{\Delta x_r'}$ yields

$$\frac{\Delta x_r}{\Delta x_r'} = (\frac{k(x-x_e) + f_e}{k(x + \delta x_e - x_e) + f_e}) \tag{59}$$

Dividing the relationship by $k(x - x_e) + f_e$ yields

$$\frac{\Delta x_r}{\Delta x'_r} = \frac{1}{1 + \frac{k k_e \delta x_e}{(k+k_e) f_e}} \tag{60}$$

Making use of $k_{eff} = \frac{k k_e}{k+k_e}$, we can show that (60) becomes

$$\frac{\Delta x_r}{\Delta x'_r} = \frac{1}{1 + \frac{k_{eff} \delta x_e}{f_e}} \tag{61}$$

If the environment is stiff such that $k_e >> k$ then $k_{eff} \approx k$.

$$\frac{\Delta x_r}{\Delta x'_r} \cong \frac{1}{1 + \frac{k \delta x_e}{f_e}} \tag{62}$$

To have minimal effect on environment position error, the term $\frac{k \delta x_e}{f_e}$ is required to be close to zero. As an example, consider the case where position measurement error is $\delta x_e = 0.5mm$, the desired force is 5N and $k = 500$. Then $\frac{k \delta x_e}{f_e} = \frac{500 * 0.0005}{5} = 0.05$ and $\frac{\Delta x_r}{\Delta x'_r} = 0.9524$ which is about 5% error.

9 NN Controller Performance

A comprehensive simulation study has been carried out using a three link rotary robot manipulator whose parameters are taken from the first three links of PUMA 560 arm. The nominal system parameters are used as the basis in forming the robot model $\hat{D}(q)$ and $\hat{h}(q, \dot{q})$. Model uncertainties included a 10 Kg mechanical tool attached to the third link, Coulomb friction, viscous friction forces $\tau_f(\dot{q})$ added to each joint where $\tau_f(\dot{q}) = 0.5 sgn(\dot{q}) + 0.8(\dot{q})$.

The discontinuous stiffness profile shown in Figure 9 is

$$k_e = \begin{cases} 20000 & 0 \leq t < 4 \\ 60000 & 4 \leq t < 8 \\ 40000 & 8 \leq t \end{cases} \tag{63}$$

The continuous stiffness profile shown in Figure 12 is

$$k_e = \begin{cases} 10000 & 0 \leq t < 2 \\ 1000\ e^{0.80472(t-2)} & 2 \leq t < 4 \\ 50000 & 4 \leq t < 5 \\ 50000 + 5000\ sin(\frac{\pi(t-5)}{2}) & 5 \leq t < 11 \\ 50000 & 11 \leq t < 12 \\ 50000\ e^{0.80472(t-11)} & 12 \leq t < 14 \\ 10000 & 14 \leq t \end{cases} \tag{64}$$

The performances of the proposed schemes are tested by tracking two different tasks : flat sine wave tracking(called Task 1) with discontinuous environment stiffness profile and circular tracking(called Task 2) with continuous environment stiffness profile as shown in Figure 7 and 8, respectively. For the NN controller, we have chosen six hidden neurons ($n_H = 6$). The back-propagation algorithm parameters are: $w_{ij}^k(0)$ is randomly selected, $\alpha = 0.9$. The controller gains are selected as $K_D = diag[300, 20, 20]$ and $K_P = diag[900, 900, 900]$ for Task 1, and $K_D = diag[200, 200, 200]$ and $K_P = diag[500, 500, 500]$ for Task 2 which give critically damped or over-damped motions at the end-effector. For the Task 1, the robot manipulator is required to move up and down in z direction while holding the same amount of forces on x direction with discontinuous environment stiffness profile shown in Figure 9. The one cycle time is 4 secs. The update rate $\eta = 0.00008$ for PBNNIC and $\eta = 0.08$ for TBNNIC is used. Sample tracking results for Task 1 are plotted in Figures 10 and 11. The simulation data showed that the rate of convergence is very fast with a convergence time less than 0.5 second. Thus on-line trajectory control of the proposed NN scheme is completely acceptable in practice. As a base line comparison, we also plotted the performance of the uncompensated system under same condition in Figure 10 and 11. It is seen that the force and position tracking errors are very large, and improvement of using NN is clearly demonstrated. We also see that the performance of the PBNNIC is slightly better than that of the TBNNIC.

Next task is that the robot is required to move circular trajectory with continuous environment stiffness profile shown in Figure 12. Since the environment is tilted by 45 degrees as shown in Figure 7 the total force normal to the surface has three components:$f_{e_x}, f_{e_y}, f_{e_z}$ in x, y, z direction. The update rate η for this task is 0.05 for TBNNIC and 0.0001 for PBNNIC. The force tracking response is plotted in Figure 13. The force tracking response of the uncompensated control system is not plotted because the error is too large. The improvement achieved by the NN controller is again clearly demonstrated. The small force overshoots at initial impact are settled down within 0.5 sec. Figure 14 shows the end-point reference trajectories x_r and actual trajectories x of the circular command trajectory for TBNNIC. This plot demonstrates that position tracking under NN control is excellent. The plot also shows that the same distance between actual position and reference position is maintained. We also see that the performance of the PBNNIC is slightly better than that of the TBNNIC in this task.

In order to prove the robustness of the proposed NN schemes we test the Task 2 with PBNNIC scheme with the random force sensor noise added to force measurements. The sensor noise is uniformly distributed

within ±5% of the actual force measurement. We see the force tracking results in Figure 15. Actually the force error is less than ±5% of the actual force plotted in Figure 13 under the same condition without sensor noise. We notice that more force errors occur in higher stiffness region which do not follow the analysis in section 9. Even though the ratio $\frac{\Delta x_r}{\Delta x_r}$ is minimized, the force error becomes larger at higher stiffness environment because the actual force generated by $f_e = k_e(x - x_e)$ is very sensitive to environment stiffness.

As expected, the impedance controller is able to track effectively the desired contact force for a time varying stiffness profile. Thus the simulation studies show that the proposed NN force tracking impedance controller is able to compensate for not only uncertainties in robot dynamics but uncertainties in environment.

10 Conclusions

Two new robust neural network impedance control schemes of robot manipulator are presented in this paper. One is the torque-based NN impedance control and another is the position-based NN impedance control. The NN compensators serve as auxiliary controllers for both the impedance control structures to counteract the uncertainties in the robot dynamics and disturbances occurring from environment during application. Also introduced is a simple technique to design the reference trajectory using the sensed contact force instead of environment stiffness to compensate for the uncertainties in environment stiffness. A unified training signal for both torque-based and position based impedance control schemes is developed for training the back-propagation algorithm, and time delayed reference trajectories are used as inputs to the NN. Simulations based on a three link robot show that system performances of both NN impedance control schemes under these conditions are excellent and the convergence rates are under one second. Thus the proposed NN controllers are feasible for on-line robot control. The performance of PBNNIC is slightly better than that of TBNNIC with paying more weights in NN. In addition, the PBNNIC has a structural advantage so that the implementation of the robust force control from position-controlled robot system can be done without modifying inside control structure.

References

[1] N. Hogan, "Impedance control : An approach to manipulator, part i, ii, iii", *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 3, pp. 1–24, 1985.

[2] M. H. Raibert and J. J. Craig, "Hybrid position and force control of robot manipulators", *ASME Journal of Dynamic Systems, Measurement, and Control*, pp. 126–133, 1981.

[3] D. A. Lawrence, "Impedance control stability properties in common implementations", *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 1185–1190, 1988.

[4] S. Jung and T. C. Hsia, "On neural network controller design for position based force control of robot manipulators", *Proc. of IASTED International Conference on Robotics and Manufacturing*, pp. 159–162, Cancun, June, 1995.

[5] Mark W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, John Wiley and Sons, 1989.

[6] H. Miyamoto, M. Kawato, T. Setoyama, and R. Suzuki, "Feedback error learning neural network for trajectory control of a robotic manipulator", *IEEE Trans. on Neural Networks*, vol. 1, pp. 251–265, 1988.

[7] A. Ishiguro, T. Furuhashii, S. Okuma, and Y. Uchikawa, "A neural network compensator for uncertainties of robot manipulator", *IEEE Trans. on Industrial Electronics*, vol. 39, pp. 61–66, December, 1992.

[8] J. Yuh and R. Lakshmi, "An intelligent control system for remotely operated vehicles", *IEEE J of Ocean Engineering*, vol. 18, pp. 55–62, 1993.

[9] F.L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance", *IEEE Symposium on Intelligent Control*, pp. 225–231, 1993.

[10] S. Jung and T.C. Hsia, "A study on new neural network schemes for robot manipulator control", *Robotica*, vol. 14, pp. 7–16, 1996.

[11] S. Jung and T. C. Hsia, "A new neural network control technique for robot manipulators", *Robotica*, vol. 13, pp. 477–484, 1995.

[12] J. M. Tao and J. Y. S. Luh, "Application of neural network with real-time training to robust position/force control of multiple robots", *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 142–148, 1993.

[13] R. Colbaugh, H. Seraji, and K. Glass, "Direct adaptive impedance control of robot manipulators", *Journal of Robotic Systems*, vol. 10, pp. 217–248, 1993.

[14] T. Lasky and T.C. Hsia, "On force-tracking impedance control of robot manipulators", *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 274–280, 1991.

[15] H. Seraji and R. Colbaugh, "Force tracking in impedance control", *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 499–506, 1993.

[16] R. Colbaugh and A. Engelmann, "Adaptive compliant motion of manipulators : Theory and experiments", *Proc. of IEEE International Conference on Robotics and Automations*, pp. 2719–2726, 1994.

[17] H. Seraji, "Adaptive admittance control : An approach to explicit force control in compliant motion", *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 2705–2712, 1994.

[18] S. Jung, T. C. Hsia, and R. G. Bonitz, "On force tracking impedance control with unknown environment stiffness", *Proc. of IASTED International Conference on Robotics and Manufacturing*, pp. 181–184, Cancun, June, 1995.

[19] S. Jung and T. C. Hsia, "On reference trajectory modification approach for cartesian space neural network control of robot manipulators", *Proc. of IEEE International Conference on Robotics and Automation*, pp. 575–580, Nagoya, 1995.

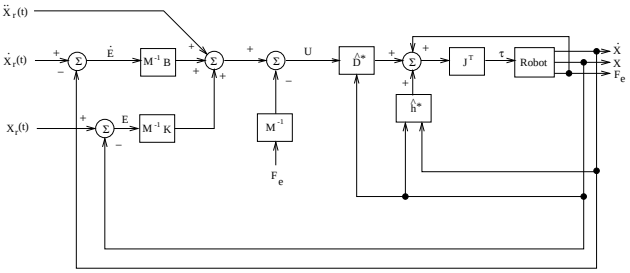

Figure 1: Torque-Based Force Control Structure

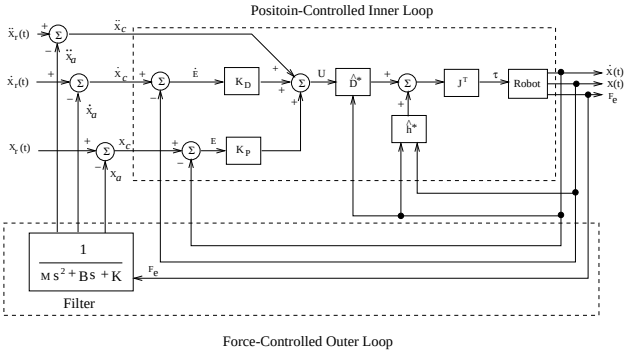

Figure 2: Position-Based Force Control Structure

Seul Jung(M'96) was born in Incheon, Korea, in 1964. He went to Han Yang University in Seoul, in 1983. He moved to United States in 1984 and received the B.S in Electrical Engineering from Wayne State University. He continued his studies and received M.S and Ph.D from The University of California, Davis, in 1991 and 1996, respectively. He worked as volunteers for the IEEE Conference on Robotics and Automations held in 1991 and 1995. He received two best student papers in separate international conferences. He has published 25 papers in the Robotics area. His research interests are robot control, intelligent control

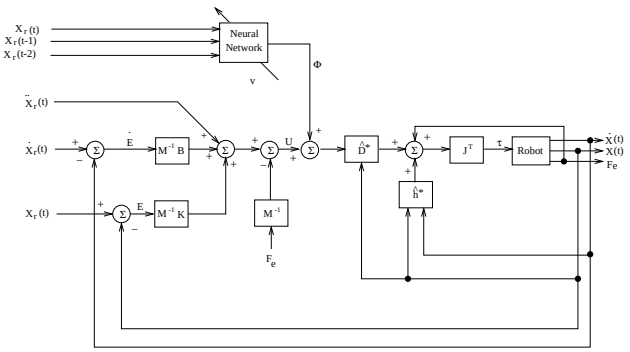

Figure 3: The proposed Torque-Based NN Force Control Structure

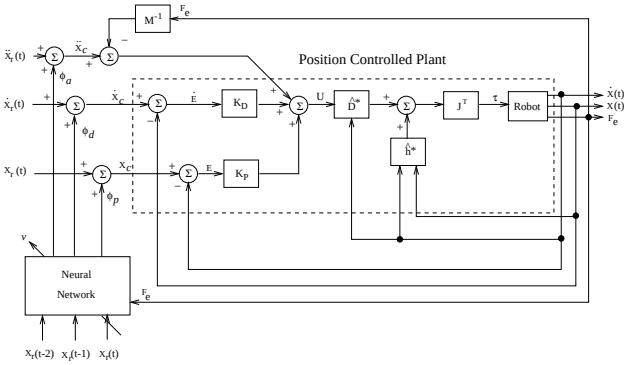

Figure 4: The proposed Position-Based NN Force Control Structure

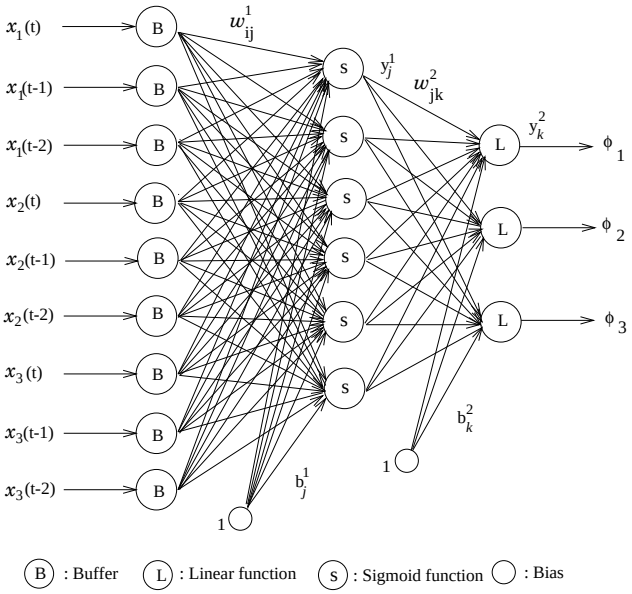

Figure 5: NN structure

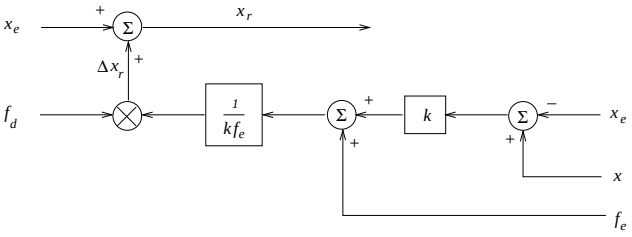

Figure 6: The Simple Trajectory Modification Loop

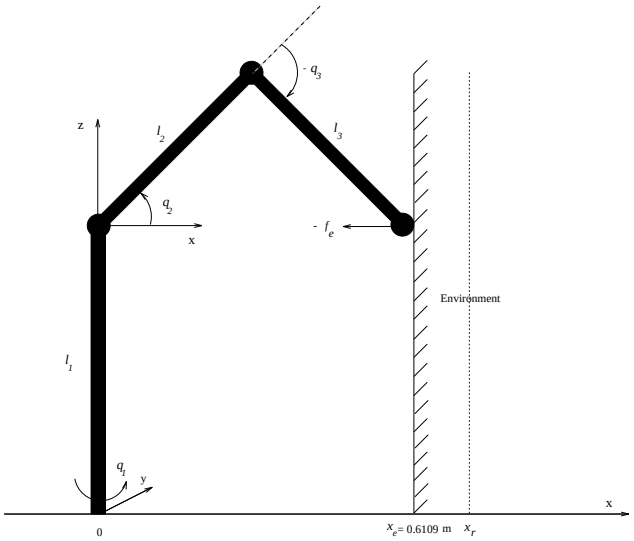

Figure 7: Task 1 : Sine Wave Tracking on Flat Surface Environment

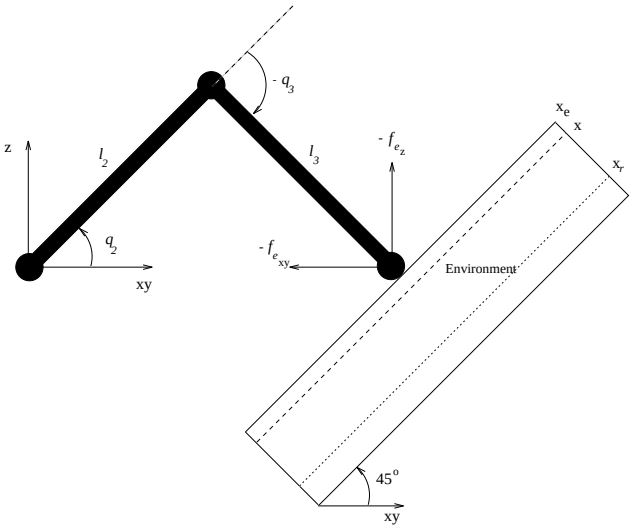

Figure 8: Task 2 : Circular Tracking on 45^o Tilted Surface Environment

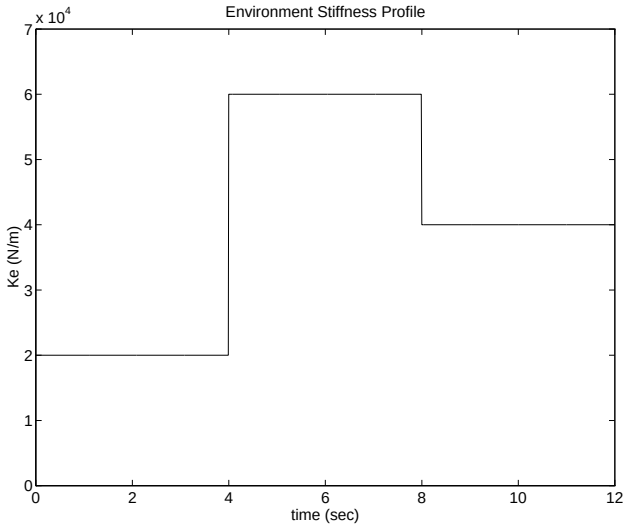

Figure 9: Time-Varying Discontinuous Environment Stiffness Profile I

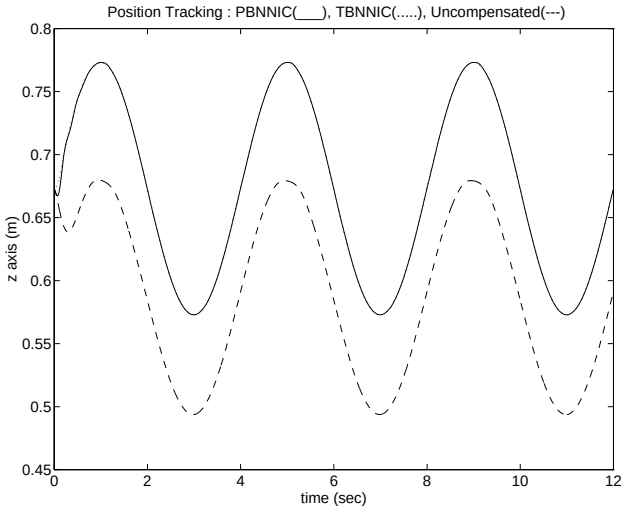

Figure 11: Task 1 (Flat Wall) : Position Tracking

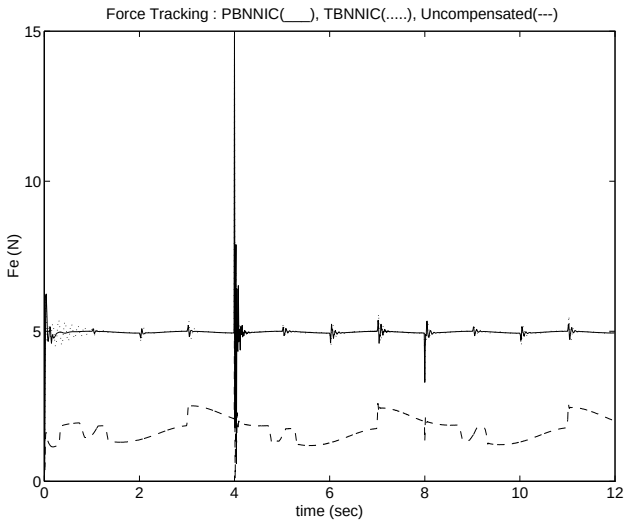

Figure 10: Task 1 (Flat Wall) : Force Tracking

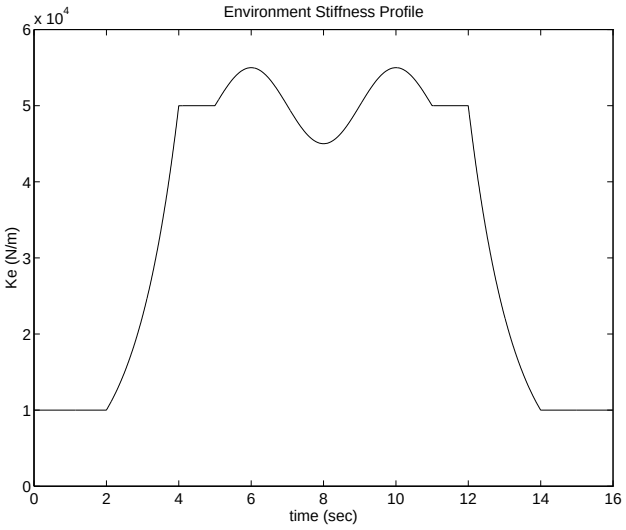

Figure 12: Time-Varying Continuous Environment Stiffness Profile II

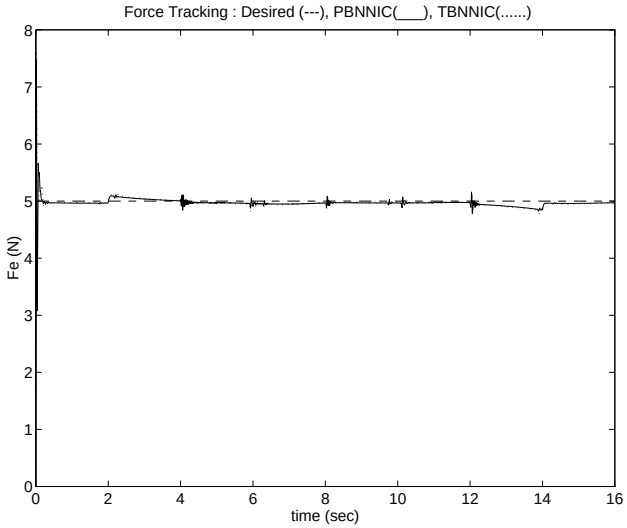

Figure 13: Task 2 (Circular Trajectory Tracking) : Force Tracking

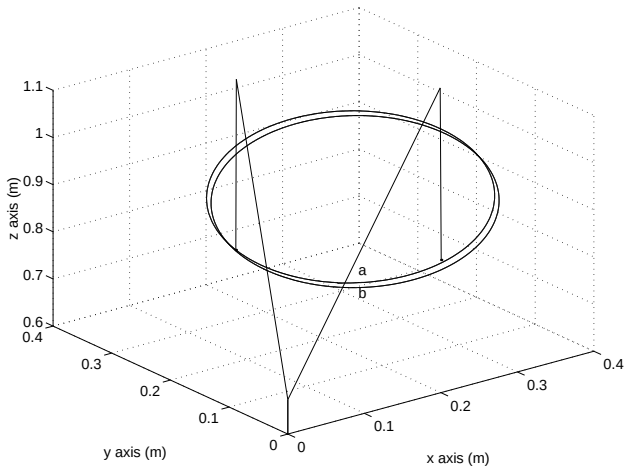

Figure 14: Task 2 : End Point Tracking of Circular Trajectory of TBNNIC : (a) Actual Tracking (b) Reference Trajectory

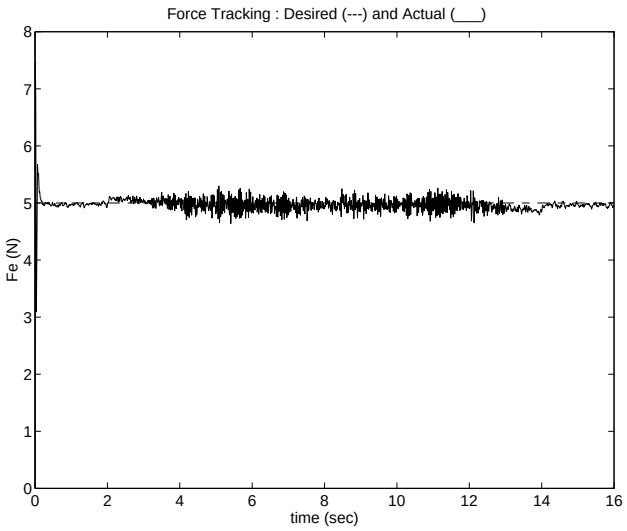

Figure 15: Task 2 : Force Tracking of Circular Trajectory of PBNNIC under ±5% Force Sensor Noise

and classical control design. Currently he is an assistant professor in Mechatronics Department at Chung Nam National University in Korea. He is now serving as a secretary of the IEEE International Symposium on Computational Intelligences in Robotics and Automations.

T. C. Hsia(Fellow) received the B.S. degree from the National Taiwan University and the PhD. degree from Purdue University, both in Electrical Engineering. He is a Professor in the Department of Electrical and Computer Engineering at the University of California at Davis, where he joined the faculty in 1965. He has been engaged in research and teaching in the areas of system identification, adaptive control, and adaptive signal processing. Since 1981, he has focused his research and teaching efforts in robotics and has established a Robotics Research Laboratory for experimental studies. He has published over 140 papers on these subjects. He is the author of the book, System Identification, published by Lexington Books in 1977. In 1986, he founded the International Journal of Robotics and Automation, published by the International Association of Science and Technology for Development (IASTED), and served and Editor-in-Chief until 1993.

Dr. Hsia is a Fellow of the IEEE. He was the General Chair of the 1991 International Conference on Robotics and Automation in Sacramento, CA; Program Chair of the First Chinese World Congress on Intelligent Control and Intelligent Automation in Beijing, China, 1993; and the General Chair of IASTED International Conference on Robotics and Manufac-

turing in Cancun, Mexico, 1995. He has been the Vice President of Finance of the IEEE Robotics and Automation Society since 1993.