An Approach to Uncertainty Compensation Using a Neural Network for Multi-Manipulator System Control

Peter C.Y. Chen* James K. Mills* Kenneth C. Smith*†‡

*Department of Mechanical Engineering, University of Toronto, 5 King's College Road, Toronto, Canada M5S 1A4

†Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, Canada M5S 1A4

‡Department of Library and Information Science, University of Toronto, 140 St. George Street, Toronto, Canada M5S 1A1

Abstract

An approach to uncertainty compensation using a multilayer feedforward neural network in multi-manipulator system control is proposed. The proposed approach is developed by formulating the dynamics of the multi-manipulator system in the constrained motion framework.

The error-backpropagation algorithm is employed for neural network learning. The teaching signal for neural network learning is derived by analyzing the stability of the closed-loop system. It is shown that if the neural network learns to generate the proper compensating signal, then the constrained motion of the multi-manipulator system tracks the desired motion asymptotically; as a consequence, the desired forces can be achieved.

Computer simulations are conducted to verify the proposed approach.

1 Introduction

A multi-manipulator system consists of two or more robots functioning in a coordinated fashion to accomplish a task. An application of this type of robotic system is in the area of fixtureless assembly, where, for instance, two mechanical manipulators hold two separate work pieces together while the pieces are being bonded [15]. Other applications include such tasks as space station assembly, maintenance, and servicing.

Significant research has been reported in the literature on the control of multi-manipulator systems. The two main approaches are the so-called master/slave formulation (e.g. [12, 1]) and the hybrid force/position control method (e.g. [8]). Often implicit in these approaches is the assumption that the dynamic parameters of the manipulators involved are known precisely. Such a restrictive assumption undermines the practicality of these approaches. To circumvent this difficulty, other control strategies have been proposed to deal with parameter uncertainty associated with the manipulators and to improve the robustness of the control system [10, 11, 5, 17].

Recently (artificial) neural networks have been employed in the area of robotic control. Application of neural networks to free motion control [16, 7, 3] and contact task control [6, 4] have been reported. An approach using a neural network for uncertainty compensation in the control of multi-manipulator system is proposed in [18], where the neural network is used in conjunction with a hybrid force/position control scheme.

In this paper, we proposed an approach for uncertainty compensation using a neural network in multi-manipulator system control. This proposed approach is formulated in the framework of constrained motion. By formulating the multi-manipulator dynamics within the constrained motion framework (as in [15]), the resulting dynamic equations of motion are expressed in the most natural form in a set of generalized coordinates, thus leading to a simplified framework in which the control law is derived and issue of stability examined.

Section 2 formulates the dynamics of the multi-manipulator system in the framework of constrained motion. Section 3 presents the proposed control law. Section 4 describes the neural network compensation scheme. Section 5 presents computer simulation results. Section 6 concludes the paper.

2 Multi-manipulator System Dynamics

We consider a robotic system consists of N cooperative manipulators handling a common object. The interaction between these manipulators is to be represented by the generalized forces exerted on the payload by the manipulators. The equations of motion of a manipulator i

$(i = 1, ..., N)$ with n_i joints can be expressed as

$$M_i(q_i)\ddot{q}_i + h_i(q_i, \dot{q}_i) = \tau_i - f_i \qquad (1)$$

where $q_i \in \mathcal{R}^{n_i}$, $\dot{q}_i \in \mathcal{R}^{n_i}$, and $\ddot{q}_i \in \mathcal{R}^{n_i}$ are respectively the joint position, joint velocity, and joint acceleration vectors, $M_i \in \mathcal{R}^{n_i \times n_i}$ is the inertia matrix, $h_i \in \mathcal{R}^{n_i}$ is a vector containing the Coriolis and gravitational terms, $\tau_i \in \mathcal{R}^{n_i}$ is the input torque vector, and $f_i \in \mathcal{R}^{n_i}$ is the generalized joint reaction due to the generalized forces exerted by the end-effector of the manipulator on the object.

Let

$$\begin{aligned} n &= \sum_{i=1}^{N} n_i \\ q &= (q_1, q_2, ..., q_i, ..., q_N)^T \in \mathcal{R}^{n \times 1} \\ M(q) &= diag(M_1, M_2, ..., M_i, ..., M_N) \in \mathcal{R}^{n \times n} \\ h(q, \dot{q}) &= (h_1, h_2, ..., h_i, ..., h_N)^T \in \mathcal{R}^{n \times 1} \\ \tau &= (\tau_1, \tau_2, ..., \tau_i, ..., \tau_N)^T \in \mathcal{R}^{n \times 1} \\ f &= (f_1, f_2, ..., f_i, ..., f_N)^T \in \mathcal{R}^{n \times 1} \end{aligned}$$

then (1) can be written in a compact form as

$$M(q)\ddot{q} + h(q, \dot{q}) = \tau - f. \qquad (2)$$

We make the following assumptions regarding the multi-manipulator system:

1. The mass and inertia of the object can be lumped into the last links of the manipulators.
2. The end-effector of each manipulator is in rigid contact with the object. This means that the entire system can be considered to be a closed-chain mechanism.
3. Each manipulator is completely rigid and non-redundant.

Since the manipulators are in rigid contact with the object, the motion of the manipulators are constrained in certain directions. The constraint may be expressed as

$$\varphi(q) = 0$$

where $\varphi(\cdot) \in \mathcal{R}^m$. It is assumed that $\varphi(\cdot)$ is continuous and twice differentiable.

We partition the vector q into two subvectors $\bar{q}_1$ and $\bar{q}_2$, i.e. $q = (\bar{q}_1^T, \bar{q}_1^T)^T$, where $\bar{q}_1 \in \mathcal{R}^m$, and $\bar{q}_2 \in \mathcal{R}^{n-m}$. We then assume that a function Ω exists such that the constraint can be expressed

$$\varphi(\Omega(\bar{q}_2), \bar{q}_2) = 0.$$

We next introduce a nonlinear coordinate transformation as described in [13]. Let

$$x = \begin{bmatrix} \bar{q}_1 - \Omega(\bar{q}_2) \\ \bar{q}_2 \end{bmatrix}.$$

Then

$$q = \begin{bmatrix} x_1 + \Omega(x_2) \\ x_2 \end{bmatrix} \equiv Q(x).$$

Since $\dot{q} = \frac{\partial Q(x)}{\partial x}\dot{x} \equiv T\dot{x}$, and $\ddot{q} = T\ddot{x} + \dot{T}\dot{x}$, Equation (2) becomes, with arguments suppressed

$$\bar{M}\ddot{x} + \bar{h} = T^T\tau - T^T f \qquad (3)$$

where $\bar{M} = T^T M T$, and $\bar{h} = T^T(M\dot{T}\dot{x} + h)$.

Note that this nonlinear transformation results in $x_1 = 0$. We partition the identity matrix $I_n \in \mathcal{R}^{n \times n}$ into two "sub-matrices" E_1 and E_2, i.e. $I_n = [E_1^T, E_2^T]$, where $E_1 \in \mathcal{R}^{m \times n}$ and $E_2 \in \mathcal{R}^{(n-m) \times n}$. Utilizing the relationships $x_1 = 0$ and $E_2 T^T f = 0$, we can express (3) in "reduced" form as

$$E_1\bar{M}E_2^T\ddot{x}_2 + E_1\bar{h} = E_1T^T\tau - E_1T^T f \qquad (4)$$

$$E_2\bar{M}E_2^T\ddot{x}_2 + E_2\bar{h} = E_2T^T\tau. \qquad (5)$$

3 Control

In practical robotic applications, it is usually the case that the parameters $\bar{M}$ and $\bar{h}$ are not known exactly. To construct a control law based on the dynamics model, we use the "nominal" (or estimated) values of these parameters, denoted by $\tilde{M}$ and $\tilde{h}$ respectively.

We now present the proposed control law for the system (3). The proposed control law is a modified computed-torque control as in [13], plus an additional compensating signal v, and is specified as

$$\begin{aligned} T^T\tau &= \tilde{M}(\ddot{x}^d + E_2^T K_v E_2(\dot{x}^d - \dot{x}) + E_2^T K_p E_2(x^d - x)) \\ &\quad \tilde{M}v + \tilde{h} + E_1^T K_f E_1 T^T(f^d - f) + T^T f^d \qquad (6) \end{aligned}$$

where $K_v \in \mathcal{R}^{(n-m)\times(n-m)}$, $K_p \in \mathcal{R}^{(n-m)\times(n-m)}$, and $K_f \in \mathcal{R}^{m \times m}$ are diagonal constant gain matrices, x^d and x are respectively the desired and actual position trajectory, f^d and f are respectively the actual and the desired constraint force.

Substituting (6) into (3) yields the closed-loop system dynamics

$$\ddot{e}_x + \bar{K}_v\dot{e}_x + \bar{K}_p e_x = \eta - v - \Delta f \qquad (7)$$

where

$$e_x = x^d - x, \quad \bar{K}_v = E_2^T K_v E_2, \quad \bar{K}_p = E_2^T K_p E_2$$

$$\eta = (\tilde{M}^{-1}\bar{M} - I)\ddot{x} + \tilde{M}^{-1}\Delta\bar{h}, \quad \Delta\bar{h} = \bar{h} - \tilde{h}$$

$$\Delta f = \tilde{M}^{-1}K_{\Delta_f}(f^d - f), \quad K_{\Delta_f} = \bar{K}_f + T^T$$

$$\bar{K}_f = E_1^T K_f E_1 T^T.$$

Let

$$e = \begin{bmatrix} e_x \\ \dot{e}_x \end{bmatrix}, \quad A = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

and $\Delta v = \eta - v$, then we can express (7) as

$$\dot{e} = Ae + B(\Delta v - \Delta f). \qquad (8)$$

Figure 1 schematically depicts the closed-loop system.

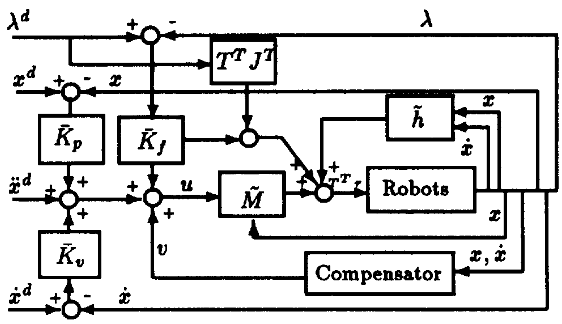

Figure 1 Multi-Manipulator System Control.

We next show that if the compensator is designed such that $\Delta v \to 0$, then the asymptotic motion of the manipulators can be achieved while the actual force tracks the desired force. Recall that

$$\eta = (\tilde{M}\bar{M} - I)\ddot{x} + \tilde{M}^{-1}\Delta\bar{h}.$$

We can express $\Delta v = 0$ as

$$(\bar{M} - \tilde{M})\ddot{x} + \Delta\bar{h} - \tilde{M}^{-1}v = 0$$

or in reduced form as

$$\begin{aligned} E_1(\bar{M} - \tilde{M})E_2^T\ddot{x}_2 + E_1\Delta\bar{h} - E_1\tilde{M}v &= 0 \qquad (9) \\ E_2(\bar{M} - \tilde{M})E_2^T\ddot{x}_2 + E_2\Delta\bar{h} - E_2\tilde{M}v &= 0. \qquad (10) \end{aligned}$$

Note that the proposed control law (6) can also be expressed in reduced form as

$$\begin{aligned} E_1 T^T \tau &= E_1\tilde{M}E_2^T(\ddot{x}_2^d + K_v(\dot{x}_2^d - \dot{x}_2) + K_p(x_2^d - x_2)) \\ &\quad + E_1\tilde{M}v + E_1\tilde{h} + E_1E_1^T K_f E_1 T^T(f^d - f) \\ &\quad + E_1 T^T f^d \qquad (11) \\ E_2 T^T \tau &= E_2\tilde{M}(\ddot{x}_2^d + K_v(\dot{x}_2^d - \dot{x}_2) + K_p(x_2^d - x_2)) \\ &\quad + E_2\tilde{M}v + E_2\tilde{h} + E_2E_1^T K_f E_1 T^T(f^d - f) \\ &\quad + E_2 T^T f^d. \qquad (12) \end{aligned}$$

Since $E_2T^Tf = 0$, $E_1E_1^T = I_m$, and $E_2E_1 = 0$, Equations (11) and (12) become

$$\begin{aligned} E_1 T^T \tau &= E_1\tilde{M}E_2^T(\ddot{x}_2^d + K_v(\dot{x}_2^d - \dot{x}_2) + K_p(x_2^d - x_2)) \\ &\quad + E_1\tilde{M}v + E_1\tilde{h} + K_f E_1 T^T(f^d - f) \\ &\quad + E_1 T^T f^d \qquad (13) \\ E_2 T^T \tau &= E_2\tilde{M}E_2^T(\ddot{x}_2^d + K_v(\dot{x}_2^d - \dot{x}_2) + K_p(x_2^d - x_2)) \\ &\quad + E_2\tilde{M}v + E_2\tilde{h}. \qquad (14) \end{aligned}$$

Substituting (13-14) into (4-5) yields the closed-loop system equations in reduced form

$$\begin{aligned} E_1\bar{M}E_2^T\ddot{x}_2 &+ E_1\bar{h} = E_1\tilde{M}E_2^T(\ddot{x}_2^d + K_v(\dot{x}_2^d - \dot{x}_2) \\ &+ K_p(x_2^d - x_2)) + E_1\tilde{M}v + E_1\tilde{h} \\ &+ K_f E_1 T^T(f^d - f) + E_1 T^T f^d \\ &- E_1 T^T f \qquad (15) \\ E_2\bar{M}E_2^T\ddot{x}_2 &+ E_2\bar{h} = E_2\tilde{M}E_2^T(\ddot{x}_2^d + K_v(\dot{x}_2^d - \dot{x}_2) \\ &+ K_p(x_2^d - x_2)) + E_2\tilde{M}v \\ &+ E_2\tilde{h} \qquad (16) \end{aligned}$$

or

$$\begin{aligned} E_1\tilde{M}E_2^T(\ddot{e}_2 &+ K_v\dot{e}_2 + K_pe_2) = E_1(\bar{M} - \tilde{M})E_2^T\ddot{x}_2 \\ &+ E_1\Delta\bar{h} - E_1\tilde{M}v + K_{\Delta_f}(f^d - f) \quad (17) \\ E_2\tilde{M}E_2^T(\ddot{e}_2 &+ K_v\dot{e}_2 + K_pe_2) = E_2(\bar{M} - \tilde{M})E_2^T\ddot{x}_2 \\ &+ E_2\Delta\bar{h} - E_2\tilde{M}v \qquad (18) \end{aligned}$$

where $e_2 = x_2^d - x_2$. Equation (18) characterizes the motion of the manipulators under the constraint $\varphi = 0$, while (17) is the equation of forces expressed in terms of the dynamics of the constrained motion. We "decompose" the overall dynamic equation in this reduced form for the reason that, put in such a form, the "motion aspect" and the "force aspect" of the manipulators can be more readily examined.

Now substituting (9-10) into (17-18), we obtain

$$\begin{aligned} E_1\tilde{M}E_2^T(\ddot{e}_2 + K_v\dot{e}_2 + K_pe_2) &= K_{\Delta_f}(f^d - f) \quad (19) \\ E_2\tilde{M}E_2^T(\ddot{e}_2 + K_v\dot{e}_2 + K_pe_2) &= 0. \qquad (20) \end{aligned}$$

We can see that equation (20) characterizes the desired asymptotic motion of the manipulators under the constraint $\varphi = 0$. With appropriate K_v and K_p, we can obtain $e_2 \to 0$ (and hence $q \to q^d$) as $t \to \infty$. Then from (19), it follows that $f \to f^d$.

Thus it can be concluded that if we design the compensator so as to achieve $\Delta v \to 0$, then we obtain asymptotic tracking of the desired motion of the manipulators, and consequently the desired constraint forces. Hence, we refer to Δv as the control error.

Note that the constraint force f can also be expressed in terms of the manipulator joint coordinates as

$$f = J^T\lambda$$

where $J = \frac{\partial\varphi}{\partial q}$, and $\lambda \in \mathcal{R}^m$ is a vector of Lagrange multipliers associated with the constraint $\varphi = 0$.

4 Uncertainty Compensation

Since $\Delta v = 0$ implies that $v = \eta(\cdot)$, although the structure of the function $\eta(\cdot)$ is known, the exact values of the parameters of this function are not explicitly known. An ideal compensator is a function whose output v exactly equals that of the function $\eta(\cdot)$ so that $\Delta v = 0$. Based on such a premise, the problem of designing can then be considered as a function approximation problem.

A multilayer feedforward neural network (with the error-backpropagation learning algorithm) represents an attractive mechanism for dealing with such a function approximation problem, mainly because of its ability to learn [9]. A multilayer feedforward neural network consists of a collection of processing elements (or units) arranged in a layer structure as shown in Figure 2.

For a neural network with two hidden layers, the network output is generated according to

$$v_i = g_1\left(\sum_{j=1}^{J_n} W_{ij} g_2\left(\sum_{k=1}^{K_n} R_{jk} g_3\left(\sum_{l=1}^{L_n} S_{kl} z_l\right)\right)\right)$$

where $g_m(x) = c_m \tanh(\gamma_m x)$, and c_m and γ_m are constants. For convenience we define a generalized weight vector Θ as

$$\Theta = [W_1, ..., W_{I_n}, R_1, ..., R_{J_n}, S_1, ..., S_{K_n}] \in R^{c_\theta}$$

where $(\cdot)_i$ represents the i^{th} row of the matrix $(\cdot)$, and

$$c_\theta = I_n \times J_n + J_n \times K_n + K_n \times L_n.$$

Then the mapping realized by the network can be compactly expressed as

$$v = g(Z, \Theta)$$

where Z is the input vector, i.e. $Z = [z_1, z_2, ...z_l, ...z_{L_n}]^T$.

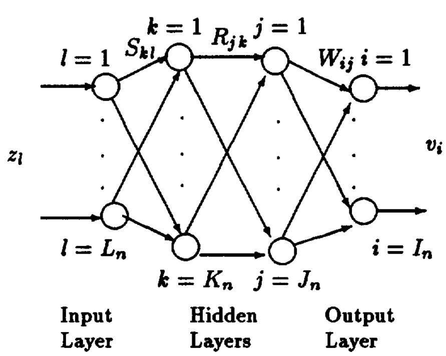

Figure 2 Neural Network Structure.

To approximate the function $\eta(\cdot)$, the neural network takes $x, \dot{x}$, and $\ddot{x}$ as its input, and produces an output v. Thus for a n-joint manipulator, the numbers of units in the input layer and the output layer are respectively $3n$ and n, i.e. $L_n = 3n, I_n = n$.

Note that the control error Δv represents the difference between the output of the uncertainty function $\eta(\cdot)$ and the output of the neural network v. The objective of neural network learning is then to effectively adjust the weights of the neural network to minimize the control error Δv. The error-backpropagation algorithm [14] is an effective algorithm for neural network learning.

Let the cost function to be minimized be

$$J_{\Delta v} = \frac{1}{2}\Delta v^T \Delta v.$$

Now proper application of the error-backpropagation algorithm yields the weight update rule

$$\dot{\Theta} = -\mu \Delta v^T \frac{\partial \Delta v}{\partial \Theta}$$

where μ is the learning rate. Since $\Delta v = v^d - v$ and $\frac{\partial v^d}{\partial \Theta} = 0$, where v^d is the desired output of the network, the update rule becomes

$$\dot{\Theta} = \mu \Delta v^T \frac{\partial v}{\partial \Theta}.$$

The error signal for neural network learning (i.e. the control error Δv) is constructed from (7)

$$\Delta v = \ddot{e}_x + \bar{K}_v \dot{e}_x + \bar{K}_p e_x + \Delta f. \tag{21}$$

Note that (21) implicitly contains the joint acceleration vector $\ddot{q}$, which can be estimated based on $\dot{q}$ using appropriate filtering techniques.

The closed-loop dynamics of the system with the neural network learning on-line is described by

$$\begin{cases} \dot{e} = Ae + B(\Delta v(x, \dot{x}, \ddot{x}, \Theta) - \Delta f) \\ \dot{\Theta} = -\mu \Delta v^T(x, \dot{x}, \ddot{x}, \Theta)\frac{\partial \Delta v(x, \dot{x}, \ddot{x}, \Theta)}{\partial \Theta}. \end{cases} \tag{22}$$

It can be proved that with a sufficiently small learning rate μ, the performance of the closed-loop system (22) improves as the learning process of the neural network iterates. This proof can be constructed in accordance with the methodology presented in [2].

5 Simulation

Computer simulations have been conducted to verify the proposed uncertainty compensation scheme. The multi-manipulator system consists of two planar robots as depicted schematically in Figure 3. One of the manipulators (robot 1 on the left) has three joints while the other (robot

2) has two. All links (except the last links) of the manipulators are of the length of $1m$; the last link of each robot has the length of $0.5m$.

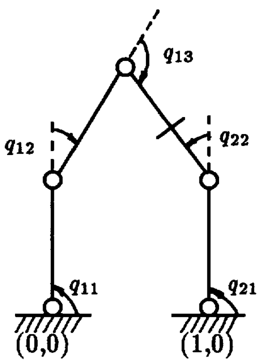

Figure 3   Two Planar Manipulators.

The constraints on the manipulators are as follows

$$\begin{aligned}\varphi_1 &= \cos q_{11} + \cos(q_{11}+q_{12}) + \frac{1}{2}\cos(q_{11}+q_{12}+q_{13}) \\ &\quad -1 - \cos q_{21} - \frac{1}{2}\cos(q_{21}+q_{22}) \\ \varphi_2 &= \sin q_{11} + \sin(q_{11}+q_{12}) + \frac{1}{2}\sin(q_{11}+q_{12}+q_{13}) \\ &\quad - \sin q_{21} - \frac{1}{2}\sin(q_{21}+q_{22}) \\ \varphi_3 &= q_{11}+q_{12}+q_{13}-q_{21}-q_{22}+\pi\end{aligned}$$

The dynamics of the system can be expressed as

$$M\ddot{q} + h = \tau - J^T\lambda$$

where

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad q_1 = \begin{bmatrix} q_{11} \\ q_{12} \\ q_{13} \end{bmatrix}, \quad q_2 = \begin{bmatrix} q_{21} \\ q_{22} \end{bmatrix}$$

$$M = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} \quad h = \begin{bmatrix} C_1\dot{q}_1 \\ C_2\dot{q}_2 \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

$$J = \frac{\partial\varphi}{\partial q}, \quad \varphi = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix}.$$

The elements of the matrices M_1, M_2, C_1, and C_2 are given in the Appendix. The partition of q is as follows

$$\bar{q}_1 = \begin{bmatrix} q_{11} \\ q_{12} \\ q_{13} \end{bmatrix}, \qquad \bar{q}_2 = \begin{bmatrix} q_{21} \\ q_{22} \end{bmatrix}.$$

To introduce parameter uncertainty into the system, the estimated values (instead of the "true" value) of the parameters (i.e. the c_i's as defined in the Appendix) were used in the control law (6). These parameter values are listed in Tables 1 and 2.

Parameter	*True*	*Estimated*
c_1	0.313	0.25
c_2	0.625	0.40
c_3	0.625	0.50
c_4	4.168	1.95
c_5	5.00	8.30
c_6	9.168	11.00

Table 1.   Parameter Values for Robot 1.

Parameter	*True*	*Estimated*
c_4	0.313	0.10
c_5	0.625	0.40
c_6	4.168	5.50

Table 2.   Parameter Values for Robot 2.

A neural network with six input units, twenty units in each of its two hidden layers, and five output units were used in the simulation. The input to the neural network were $q_{21}, q_{22}, \dot{q}_{21}, \dot{q}_2, \ddot{q}_{21}$, and $\ddot{q}_{22}$. The learning rate of the neural network was set at 0.00005. The initial weights of the network were set randomly at the order of 10^{-5}. The control gains were as follows

$$K_p = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}, \quad K_v = \begin{bmatrix} 10 & 0 \\ 0 & 20 \end{bmatrix}$$

$$K_f = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}.$$

The desired trajectories for q_{21}, q_{22}, and λ_1 were generated using a third order exponential function of the form

$$y(t) = y_0 + \delta - \delta(1 + \alpha t + \frac{\alpha^2}{2}t^2)e^{-\alpha t}$$

where α is a constant, and $\delta = y_f - y_0$, with y_0 and y_f being respectively the initial and final value of y. The values of λ_2 and λ_3 are to be regulated at zero.

The initial condition of the multi-manipulator system were set as $q_{11} = 90°$, $q_{12} = -30°$, $q_{13} = -120°$, $q_{21} = 90°$, $q_{22} = 30°$, $\lambda_1 = 0\,N$, $\lambda_2 = 0\,N$, and $\lambda_3 = 0\,Nm$. The final state of the system was specified to be $q_{21} = 70°$, $q_{22} = 60°$, $\lambda_1 = 5\,N$, $\lambda_2 = 0\,N$, and $\lambda_3 = 0\,Nm$.

To solve the set of differential-algebraic equations in (8), the second order back-difference formula were used in the numerical integration algorithm, with a step size fixed at $0.005s$. A series of 400 trials were conducted with the neural network learning on-line. For all the learning iterations, the constraints φ_1, φ_2 and φ_3 were checked and confirmed to be satisfied up to five significant digits.

Figure 4 shows the trajectory of the tip of each robot in task space. Figures 5, 6, and 7 show the force trajectories. Figures 8 through 12 show the control errors, which are seen to be significantly reduced by the neural network through learning. It is clear that, with the neural network as the uncertainty compensator, the performance of the system improves in the sense that significant reduction in the position and force errors are obtained.

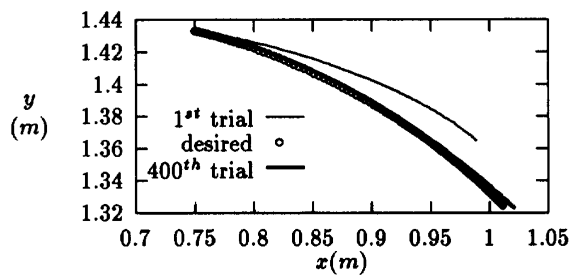

Figure 4 Position Trajectories in Task Coordinates.

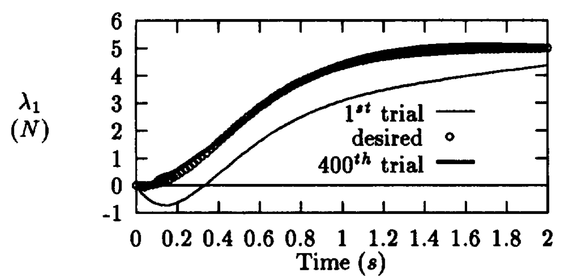

Figure 5 Force Trajectory λ_1.

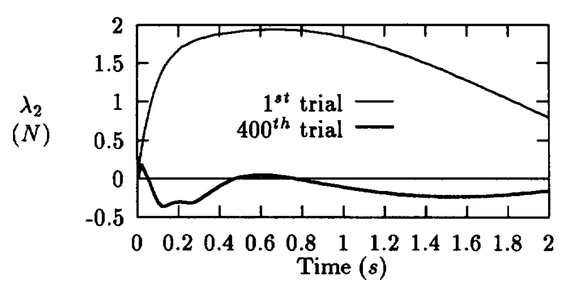

Figure 6 Force Trajectory λ_2.

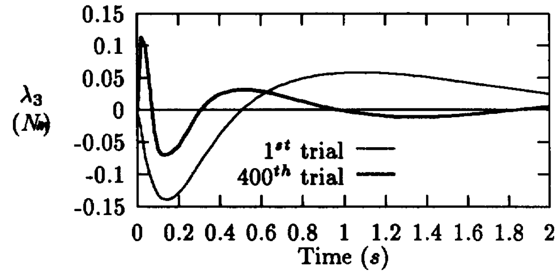

Figure 7 Force Trajectory λ_3.

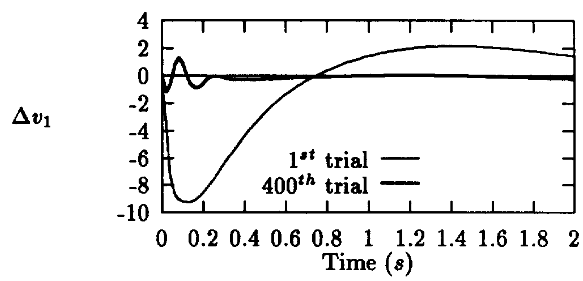

Figure 8 Control Error Δv_1.

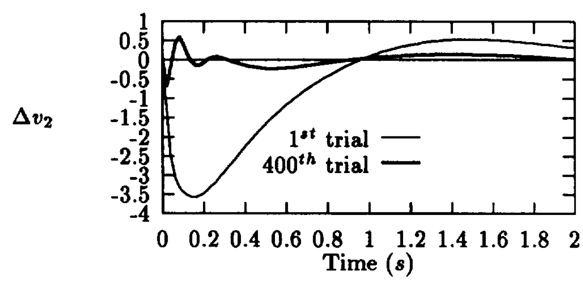

Figure 9 Control Error Δv_2.

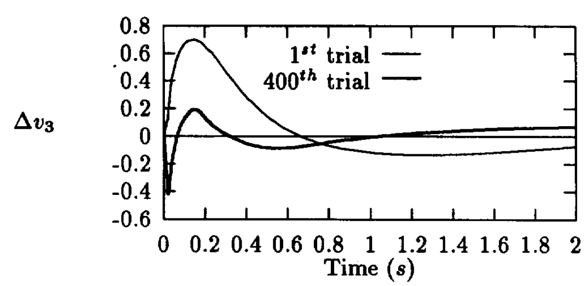

Figure 10 Control Error Δv_3.

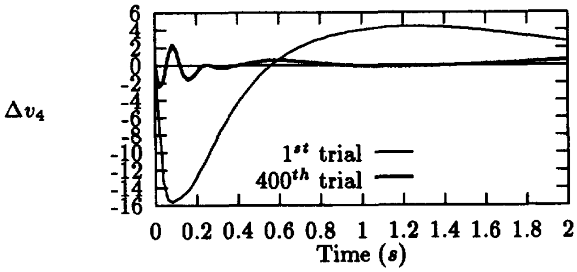

Figure 11 Control Error Δv_4.

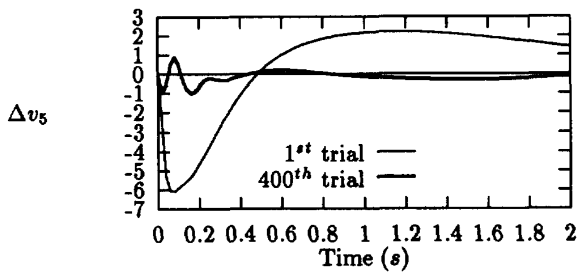

Figure 12 Control Error Δv_5.

6 SUMMARY

An approach to uncertainty compensation using a multi-layer feedforward neural network for the control of multi-manipulator system has been proposed. The proposed approach has been developed by formulating the dynamics of the multi-manipulator system in the constrained motion framework.

The error-backpropagation algorithm has been employed for neural network learning. The error signal for neural network learning has been derived by analyzing the stability of the closed-loop system. It has been shown that if the neural network learns to generate the proper compensating signal, then the constrained motion of the multi-manipulator system tracks the desired motion asymptotically; as a consequence, the desired forces can be achieved.

Computer simulations have been conducted to verify the proposed approach.

ACKNOWLEDGEMENT

This work is financially supported by IRIS, OGS, and NSERC. The authors wish to thank Professor B. Benhabib for use of the computing facility at the Computer Integrated Manufacturing Laboratory, Department of Mechanical Engineering, University of Toronto, and Dr. H. Krishnan for providing the FORTRAN code for solving differential-algebraic equations.

APPENDIX

The elements of the matrices M_1, M_2, C_1, and C_2 are as follows

$$
\begin{aligned}
M_1[1,1] &= c_1 + c_4 + c_6 + 2(c_2 + c_5)\cos q_{12} \\
&\quad + 2c_3\cos(q_{12} + q_{13}) \\
M_1[1,2] &= M_1[2,1] = c_1 + c_4 + c_5\cos q_{12} + 2c_2\cos q_{13} \\
&\quad + c_3\cos(q_{12} + q_{13}) \\
M_1[1,3] &= M_1[3,1] = c_1 + c_2\cos q_{13} + c_3\cos(q_{12} + q_{13}) \\
M_1[2,2] &= c_1 + c_4 + 2c_2\cos q_{13} \\
M_1[2,3] &= M_1[3,2] = c_1 + c_2\cos q_{13} \\
M_1[3,3] &= c_1 \\
M_2[1,1] &= c_4 + c_6 + 2c_5\cos q_{22} \\
M_2[1,2] &= M_2[2,1] = c_4 + c_5\cos q_{22} \\
M_2[2,2] &= c_4 \\
C_1[1,1] &= -c_3(\dot{q}_{12} + \dot{q}_{13}) \\
C_1[1,2] &= -c_3(\dot{q}_{11} + \dot{q}_{12} + \dot{q}_{13})\sin(q_{12} + q_{13}) \\
&\quad -(c_2 + c_5)\dot{q}_{11}\sin q_{12} - c_5\dot{q}_{12}\sin q_{12} \\
&\quad -c_2\dot{q}_{13}\sin q_{13} \\
C_1[1,3] &= -c_3(\dot{q}_{11} + \dot{q}_{12} + \dot{q}_{13})\sin(q_{12} + q_{13}) \\
&\quad -c_2(\dot{q}_{12} + \dot{q}_{13})\sin q_{13} \\
C_1[2,1] &= ((c_2 + c_5)\sin q_{12} + c_3\sin(q_{12} + q_{13}))\dot{q}_{11} \\
&\quad -c_2\dot{q}_{13}\sin q_{13} \\
C_1[2,2] &= -c_2\dot{q}_{13}\sin q_{13} \\
C_1[2,3] &= -c_2(\dot{q}_{11} + \dot{q}_{12} + \dot{q}_{13})\sin q_{13} \\
C_1[3,1] &= c_3\dot{q}_{11}\sin(q_{12} + q_{13}) \\
C_1[3,2] &= c_2\dot{q}_{12}\sin q_{13} \\
C_1[3,3] &= 0 \\
C_2[1,1] &= -c_5\dot{q}_{22}\sin q_{22} \\
C_2[1,2] &= -c_5(\dot{q}_{21} + \dot{q}_{22})\sin q_{22} \\
C_2[2,1] &= c_5\dot{q}_{21}\sin q_{22} \\
C_2[2,2] &= 0
\end{aligned}
$$

where

$$
\begin{aligned}
c_1 &= m_3 l_{c3}^2 + I_3 \\
c_2 &= m_3 l_2 l_{c3} \\
c_3 &= m_3 l_1 l_{c3} \\
c_4 &= m_2 l_{c2}^2 + m_3 l_2^2 + I_2 \\
c_5 &= m_2 l_1 l_{c2} + m_3 l_1 l_2 \\
c_6 &= m_1 l_{c1}^2 + m_2 l_1^2 + m_3 l_1^2 + I_1
\end{aligned}
$$

References

[1] Arimoto, S., F. Miyazaki, and S. Kawamura, Cooperative Motion Control of Multiple Robot Arms or Fingers. *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, 1407-1412, 1987.

[2] Chen, P.C.Y., J.K. Mills, and K.C. Smith, Uncertainty Compensation in Robot Trajectory Tracking: a Neural Network Approach. *Proceedings of the 1993 IEEE International Conference on Systems, Man and Cybernetics*, Le Touquet, France, October 17-20, 1993.

[3] Chen, C.Y.P., J.K. Mills, and K.C. Smith, Experimental Implementation of an Approach to Robot Control Using a Neural Network for Uncertainty Compensation. *Submitted to the 1994 IFAC Symposium on Robot Control*, Capri, Italy, September 19-21, 1994.

[4] Chen, C.Y.P., J.K. Mills, and K.C. Smith, Experimental Implementation of an Approach to Robot Compliant Motion Control Using a Neural Network. *Accepted for presentation at the Forth International Conference on Data and Knowledge Systems for Manufacturing and Engineering*, Hong Kong, May 2-4, 1994.

[5] Derventzis, C.A. and E.J. Davision, Robust Motion/Force Control of Cooperative Multi-Arm Systems. *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, 2230-2237, Nice, France, May, 1992.

[6] Fukuda, T. and T. Shibata, Neuromorphic Control for Robotic Manipulators: Position, Force and Impact Control. *Proceedings of the 5th IEEE International Symposium on Intelligent Control*, (Philadelphia, Pennsylvania), 310-315, 1990.

[7] Gomi, H. and M. Kawato, Learning Control for a Closed Loop System Using Feedback-Error-Learning. *Proceedings of the 29th Conference on Decision and Control*, (Honolulu, Hawaii), 3289-3294, 1990.

[8] Hayati, Samad A., Position and Force Control of Coordinated Multiple Arms. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 24, No. 5, 584-590, September 1988.

[9] Hertz, J., A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation*. Addison-Wesley: Redwood City, California, 1991.

[10] Hu, Y.R. and A.A. Goldenberg, An Adaptive Approach to Motion and Force Control of Multiple Coordinated Robot Arms. *Proceedings of the 1989 International Conference on Robotics and Automation*, 1091-1096, Scottsdale, AZ, May, 1989.

[11] Jean, Jong-Hann and Li-Chen Fu, An Adaptive Control Scheme for Coordinated Manipulator Systems. *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 2, 226-231, April 1993.

[12] Luh, J.Y.S., and Y.F.Zheng, Coordinated Relations between Two Coordinated Industrial Robots for Motion Control. *International Journal of Robotics Research*, Vol. 6, No. 3, 60-70, Fall, 1987.

[13] McClamroch, N.H. and Wang, D., Feedback Stabilization and Tracking of Constrained Robots. *IEEE Transactions on Automatic Control*, Vol. 33, pp. 419-426, May, 1988.

[14] McClelland, J.L. and D.E. Rumelhart (eds), *Parallel Distributed Processing*, Vol. 1, Cambridge, Massachusetts: MIT Press, 1986.

[15] Mills, J.K., Multi-Manipulator Control for Fixtureless Assembly of Elastically Deformable Parts. *Proceedings of Japan-USA Symposium on Flexible Automation*, (San Francisco, California), 1565-1572, 1992.

[16] Okuma, S. and A. Ishiguro, A Neural Network Compensator for Uncertainties of Robotic Manipulators. *Proceedings of the 29th Conference on Decision and Control*, (Honolulu, Hawaii), 3303-3307, 1990.

[17] Tao, Jian M. and J.Y.S. Luh, Robust Position and Force Control for System of Multiple Redundant-Robots, *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, 2211-2216, Nice, France, May, 1992.

[18] Tao, Jian M. and J.Y.S. Luh, Application of Neural Network with Real-Time Training to Robust Position/Force Control of Multiple Robots, *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, 144-148, 1993.