

VIBRATION CONTROL OF A FLEXIBLE MACRO-MICRO MANIPULATOR SYSTEM USING NEURAL NETWORKS¹

X.P. Cheng and R.V. Patel

*Department of Electrical and Computer Engineering
University of Western Ontario
London, Ontario, CANADA N6A 5B9*

Abstract: In this paper, we present a neural network based control scheme for end-effector path tracking control of a system consisting of a rigid micro manipulator attached at the end of a flexible macro manipulator. The objective is to suppress vibrations in the macro manipulator and at the same time achieve desired motions of the end-effector of the micro manipulator. A two-layer feedforward neural network is utilized to approximate the dynamic behavior of the macro-micro manipulator (M^3) system in real time, and the controller is developed without any need for prior knowledge of the dynamics. A weight-tuning algorithm for the neural network is derived using Lyapunov stability theory. It is shown that both the path tracking error and the damped vibrations are uniformly ultimately bounded under this new control scheme. Simulation results are presented and compared to those obtained using a PD joint controller.

Keywords: Neural networks, macro-micro manipulators, vibration control.

1. INTRODUCTION

Long-reach manipulators have been proposed for a range of applications that include Space Station maintenance and operation, and nuclear waste disposal. In such applications, the lightweight structure of long-reach manipulators allows the actuators to move faster and carry heavier loads than conventional rigid manipulators. However, the significant structural flexibility makes it difficult to control the position and force at the end-effector accurately and reliably. The incorporation of a small, rigid micro manipulator at the tip of a large, flexible macro manipulator has been proposed as a solution to achieve the desired accurate and robust performance. In order to utilize the macro-micro manipulator (M^3) system effectively,

one must address the problem of controlling and compensating for vibrations resulting from the flexibility in the macro-manipulator links.

In recent years considerable research work has been done to address different aspects of the M^3 system. Magee and Book (1995) considered command filtering to prevent excitation of the compliant macro manipulator dynamics while the micro manipulator performs a specified task. Chiang et al. (1991) showed that a proper mechanical design of the micro arm can avoid non-minimum phase zeros so that a stable, high-bandwidth controller can be implemented. Yoshikawa et al. (1994) utilized the redundancy of the micro manipulator to compensate for the tracking error resulting from the deformation of the macro manipulator. Lew and Trudnowski (1996) developed a flexible-motion compensator which uses feedback of strain gauge measurements mounted on the flexible arm. Torres et al. (1996) proposed

¹ This research was supported by Natural Sciences and Engineering Research Council (NSERC) of Canada under grant STPGP 215729-98.

the *Pseudo-Passive Energy Dissipation* (P-PED) method to dissipate the macro-manipulator vibrations. Moallem and Patel (1999) proposed a decoupling control scheme for rapid positioning of the end-effector of the micro-manipulator. However, all of these control methods require exact knowledge of the dynamics of the M^3 system. Since generally it is very difficult to establish an accurate dynamic model for an M^3 system, the performance of these methods may be unsatisfactory for accurate operation.

The objective of the control scheme described in this paper is to suppress vibrations in the macro manipulator while achieving stable desired motions of the end-effector of the micro manipulator. Unlike all of the above mentioned papers, a two-layer feedforward neural network is utilized in this paper to estimate the nonlinear dynamic behavior for the M^3 system, and the resulting estimates are used to develop controllers for the macro and micro manipulators without any need for prior knowledge of the dynamic model of the M^3 system. Under this neural network based control scheme, both the tracking error of the end-effector of the micro manipulator and the vibration in the macro manipulator are rapidly suppressed and constrained within an arbitrarily small vicinity of the origin, while the magnitudes of the joint torques are kept bounded. Stability of the control scheme is proved using Lyapunov stability theory. Simulation results are presented for a planar redundant M^3 system.

2. MODEL AND ERROR DYNAMICS OF AN M^3 SYSTEM

2.1 Model of an M^3 System

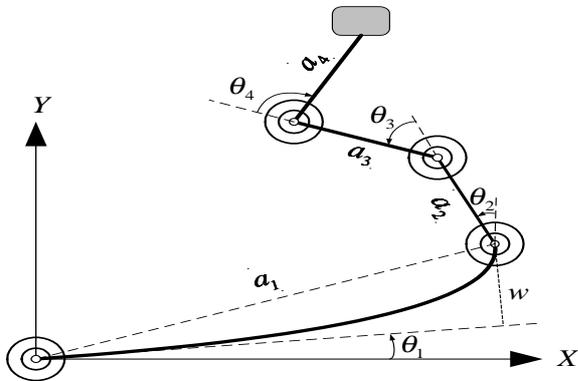


Fig. 1. Schematic of a macro-micro manipulator

Consider a system consisting of a macro manipulator with M flexible links and a micro manipulator with m rigid links (Figure 1). Let $\theta_M \in \mathbb{R}^M$ and $\theta_m \in \mathbb{R}^m$ be the joint variable vectors of the macro and micro manipulators, respectively, $\delta \in \mathbb{R}^e$ be the flexural displacement vector, and

$p \in \mathbb{R}^n$ be the end-effector position vector in the n -dimensional task space. We assume $m \geq n$ and $\text{rank}(J_{\theta_m}) = n$ for the micro manipulator. The end-point vector p is a nonlinear function of θ_M , θ_m and δ :

$$p = p(\theta_M, \theta_m, \delta) \quad (1)$$

and the small displacements \tilde{p} of the end-point vector can be expressed as

$$\tilde{p} = J_{\theta_M} \tilde{\theta}_M + J_{\theta_m} \tilde{\theta}_m + J_\delta \tilde{\delta} \quad (2)$$

where $J_{\theta_M} \in \mathbb{R}^{n \times M}$, $J_{\theta_m} \in \mathbb{R}^{n \times m}$ and $J_\delta \in \mathbb{R}^{n \times e}$ are Jacobian matrices of p with respect to θ_M , θ_m and δ respectively, and $\tilde{\theta}_M$, $\tilde{\theta}_m$ and $\tilde{\delta}$ denote small changes in θ_M , θ_m and δ respectively.

Denote $x = [\theta_M^T, \theta_m^T, \delta^T]^T$ and $\tau = [\tau_M^T, \tau_m^T, 0]^T$ with $\tau_M \in \mathbb{R}^M$ and $\tau_m \in \mathbb{R}^m$ being the control torques for the macro and micro manipulators, respectively. Using Lagrange's method, the equation of motion of the M^3 system can be written as follows (Moallem and Patel, 1999):

$$M(x)\ddot{x} + C(x, \dot{x})\dot{x} + G(x) = \tau \quad (3)$$

where M is the inertia matrix, C is the matrix of the Coriolis and centrifugal forces, G is the vector of gravity and elastic torques. Note that the dynamics have an important property which ensures that, in the parameterization, $\dot{M}(x) - 2C(x, \dot{x})$ is skew symmetric.

2.2 Tracking Error Dynamics

Denote the desired trajectories of θ_M , δ , θ_m generated from the desired end-effector trajectory p_d as

$$x_d = [\theta_{Md}^T, \theta_{md}^T, \delta_d^T]^T \quad (4)$$

The error vector is given by:

$$\tilde{x} \triangleq x - x_d = [\tilde{\theta}_M^T, \tilde{\theta}_m^T, \tilde{\delta}^T]^T \quad (5)$$

where $\tilde{\theta}_M = \theta_M - \theta_{Md}$, $\tilde{\delta} = \delta - \delta_d$, $\tilde{\theta}_m = \theta_m - \theta_{md}$. The filtered tracking error $r = [r_1^T, r_2^T, r_3^T]^T$ for the whole system is defined by

$$r_1 \triangleq \dot{\tilde{\theta}}_M + \Lambda_1 \tilde{\theta}_M + \kappa G \dot{\tilde{\delta}} + \Lambda_1 \kappa G \tilde{\delta} \quad (6)$$

$$r_2 \triangleq \dot{\tilde{p}} + \Lambda_2 \tilde{p} \quad (7)$$

$$r_3 \triangleq \dot{\tilde{\delta}} + \Lambda_3 \tilde{\delta} \quad (8)$$

where Λ_i , $i = 1, 2, 3$, are symmetric positive-definite matrices and $\kappa > 0$ is a design parameter. Substituting eqn.(2) into eqn. (7) yields

$$r_2 = J_{\theta_M} \dot{\tilde{\theta}}_M + J_{\theta_m} \dot{\tilde{\theta}}_m + J_\delta \dot{\tilde{\delta}} + \dot{J}_{\theta_M} \tilde{\theta}_M + \dot{J}_{\theta_m} \tilde{\theta}_m + \dot{J}_\delta \tilde{\delta} + \Lambda_2 (J_{\theta_M} \tilde{\theta}_M + J_{\theta_m} \tilde{\theta}_m + J_\delta \tilde{\delta}) \quad (9)$$

The vector r can be expressed in the form

$$r = \Gamma \dot{\tilde{x}} + \dot{\Gamma} \tilde{x} + \Lambda \Gamma \tilde{x} \quad (10)$$

where $\Lambda = \text{diag}\{\Lambda_1, \Lambda_2, \Lambda_3\}$ and

$$\Gamma = \begin{bmatrix} I_M & 0 & \kappa G \\ J_{\theta_M} & J_{\theta_m} & J_\delta \\ 0 & 0 & I_e \end{bmatrix} \quad (11)$$

with I_M and I_e the $M \times M$ and $e \times e$ identity matrices, respectively.

Differentiating r in (10) and utilizing (3), we can express the dynamics of the M^3 system in terms of the filtered tracking error r by

$$M\Upsilon\dot{r} + (C\Upsilon + M\dot{\Upsilon})r = f + \tau \quad (12)$$

where

$$\Upsilon \triangleq \begin{bmatrix} I_M & 0 & -\kappa G \\ -J_{\theta_m}^{-1}J_{\theta_M} & -J_{\theta_m}^{-1} & J_{\theta_m}^{-1}(J_\delta - \kappa J_{\theta_M}G) \\ 0 & 0 & I_e \end{bmatrix} \quad (13)$$

and f is an unknown nonlinear function of the robot parameters such as the inertia, centrifugal and Coriolis terms, DH parameters, etc., of the M^3 system, and is given by

$$f(u) = -G - M\ddot{x}_d - Cx_d + M\Upsilon(2\dot{\Gamma}\dot{\tilde{x}} + \ddot{\Gamma}\tilde{x} + \Lambda(\Gamma\dot{\tilde{x}} + \dot{\Gamma}\tilde{x})) + C\Upsilon(\dot{\Gamma}\tilde{x} + \Lambda\Gamma\tilde{x}) + M\dot{\Upsilon}r \quad (14)$$

where u can be chosen as

$$u \triangleq [\tilde{x}^T, \dot{\tilde{x}}^T, x_d^T, \dot{x}_d^T, \ddot{x}_d^T]^T \quad (15)$$

Expressing the error dynamics as (12) will enable us to derive a neural network based scheme in the next section.

3. NEURAL NETWORK CONTROL SCHEME

3.1 Controller Design

According to the universal approximation property of neural networks, there exists a two-layer neural network such that

$$f(u) = W_1^T \sigma(W_2^T u) + \varepsilon \quad (16)$$

where W_1 and W_2 are ideal target weights, σ is the vector of activation functions, and ε is the approximation error bounded on a compact set by ε_N , i.e., $\|\varepsilon\| < \varepsilon_N$. Define the matrix of all the weights in the neural network as

$$W \triangleq \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix} \quad (17)$$

We assume that on any compact subset of \mathbb{R}^n , the ideal neural network weights are bounded, i.e.,

$$\|W\|_F \leq b_W \quad (18)$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

Let an estimate of $f(u)$ using the neural network be given by

$$\hat{f}(u) = \hat{W}_1^T \sigma(\hat{W}_2^T u) \quad (19)$$

with \hat{W}_1 and \hat{W}_2 the actual values of the weights of the neural network determined by the learning

algorithm of the network. Note that \hat{f} can be partitioned into three blocks, $\hat{f} = [\hat{f}_1^T, \hat{f}_2^T, \hat{f}_3^T]^T$, with $\hat{f}_1 \in \mathbb{R}^M$, $\hat{f}_2 \in \mathbb{R}^m$ and $\hat{f}_3 \in \mathbb{R}^e$.

The structure of the error dynamics (12) and the subsequent stability analysis motivated us to design the neural network based controller for the M^3 system as follows:

$$\tau_M = -\hat{f}_1 + \xi_1 - \frac{\bar{r}_1 \bar{r}_3^T}{\|\bar{r}_1\|^2 + \epsilon} \cdot (\hat{f}_3 - \xi_3 + K_3 \bar{r}_3) - K_1 \bar{r}_1 \quad (20)$$

$$\tau_m = -\hat{f}_2 + \xi_2 - K_2 \bar{r}_2 \quad (21)$$

$$\xi_i = -k_W (\|\hat{W}\|_F + b_W) \bar{r}_i, \quad i = 1, 2, 3 \quad (22)$$

$$\bar{r} = [\bar{r}_1^T, \bar{r}_2^T, \bar{r}_3^T]^T = \Upsilon r \quad (23)$$

where K_i , $i = 1, 2, 3$, are positive definite matrices, $\epsilon > 0$ and $k_W > 0$ are positive design constants, and Υ is given by (13). Note that ξ_i , $i = 1, 2, 3$, are robustifying signals incorporated to compensate for approximation errors of the neural network and enhance the tracking performance of the M^3 system.

Based on the control laws defined by (20)-(23) and inspired by the modified Hebbing tuning rule for tracking control of rigid-link robotic systems (Yesildirek and Lewis, 1995), we propose a weight-tuning algorithm for the neural network that achieves stable tracking control of the flexible M^3 system. The algorithm is determined by the following equations:

$$\dot{\hat{W}}_1 = P\sigma(\hat{W}_2^T u)\bar{r}^T - \mu P\|\bar{r}\|\hat{W}_1 \quad (24)$$

$$\dot{\hat{W}}_2 = R\|\bar{r}\|u \cdot (\sigma(\hat{W}_2^T u))^T - \mu R\|\bar{r}\|\hat{W}_2 \quad (25)$$

$$\bar{r} = [\bar{r}_1^T, \bar{r}_2^T, \bar{r}_3^T]^T = \Upsilon r, \quad (26)$$

$$\bar{r}_1 \in \mathbb{R}^M, \quad \bar{r}_2 \in \mathbb{R}^m, \quad \bar{r}_3 \in \mathbb{R}^e$$

where $P = P^T$ and $R = R^T$ are any constant symmetric positive-definite matrices, $\mu > 0$ is a small scalar design parameter, and Υ is a nonlinear transformation given by (13). This algorithm is derived by an extension of Lyapunov theory, and it will be shown in the next section that under the control scheme and weight-tuning rule proposed above, the tracking error for the M^3 system can be uniformly driven to an arbitrarily small region around the desired trajectory. A block diagram of the M^3 control system is shown in Figure 2.

3.2 Stability Analysis

To analyze the stability of the proposed control scheme, choose the Lyapunov function candidate as

$$V(r, \tilde{W}_1, \tilde{W}_2, t) = \frac{1}{2}r^T \Upsilon^T M \Upsilon r + \frac{1}{2}tr\{\tilde{W}_1^T P \tilde{W}_1\} + \frac{1}{2}tr\{\tilde{W}_2^T R \tilde{W}_2\}$$

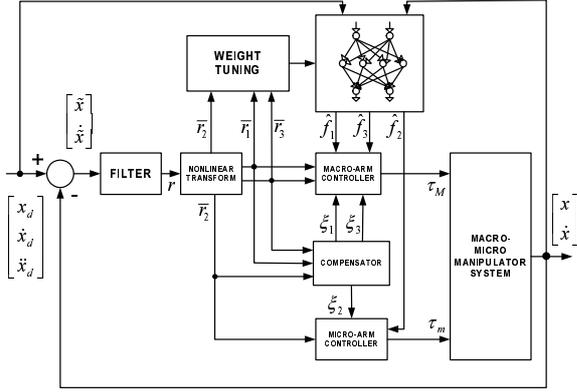


Fig. 2. Block diagram of the M^3 control system where $tr\{\cdot\}$ denotes the trace of a square matrix. Differentiating V along the solution of the error dynamics yields:

$$\begin{aligned} \dot{V} &= \frac{1}{2} r^T \Upsilon^T (\dot{M} - 2C) \Upsilon r + r^T \Upsilon^T (f + \tau) \\ &\quad + tr\{\tilde{W}_1^T P^{-1} \dot{\tilde{W}}_1\} + tr\{\tilde{W}_2^T R^{-1} \dot{\tilde{W}}_2\} \end{aligned}$$

Using the skew-symmetric property of $\dot{M} - 2C$ and the control laws defined by (20) and (23), we obtain

$$\begin{aligned} \dot{V} &= \tilde{r}_1^T (f_1 + \tau_M) + \tilde{r}_2^T (f_2 + \tau_m) + \tilde{r}_3^T f_3 \\ &\quad + tr\{\tilde{W}_1^T P^{-1} \dot{\tilde{W}}_1\} + tr\{\tilde{W}_2^T R^{-1} \dot{\tilde{W}}_2\} \\ &\leq -\tilde{r}^T K \tilde{r} + \tilde{r}^T \tilde{f} + \tilde{r}^T \xi \\ &\quad + tr\{\tilde{W}_1^T P^{-1} \dot{\tilde{W}}_1\} + tr\{\tilde{W}_2^T R^{-1} \dot{\tilde{W}}_2\} \end{aligned}$$

where $K = \text{diag}\{K_1, K_2, K_3\}$ and $\tilde{f} = f - \hat{f}$.

Substituting (14) and (19) into the above inequality and applying the trace property that $a^T b = tr\{ba^T\}$ yields

$$\begin{aligned} \dot{V} &\leq -\tilde{r}^T K \tilde{r} + tr\left\{\tilde{W}_1 \left(P^{-1} \dot{\tilde{W}}_1 + \sigma(\hat{W}_2^T u) \tilde{r}^T\right)\right\} \\ &\quad + tr\{\tilde{W}_2^T R^{-1} \dot{\tilde{W}}_2\} + \tilde{r}^T \xi + \tilde{r}^T W_1^T \tilde{\sigma} + \tilde{r}^T \varepsilon \end{aligned}$$

where we denote $\tilde{\sigma} \triangleq \sigma(W_2^T u) - \sigma(\hat{W}_2^T u)$.

Invoking the weight update laws and applying the trace property $tr\{A^T B\} \leq \|A\|_F \|B\|_F$, we obtain

$$\begin{aligned} \dot{V} &\leq -\tilde{r}^T K \tilde{r} + tr\{\mu \|\tilde{r}\| \tilde{W}_1^T \dot{\tilde{W}}_1\} \\ &\quad + tr\{\tilde{W}_2^T (\mu \|\tilde{r}\| \dot{\tilde{W}}_2 - \|\hat{r}\| u (\sigma(\hat{W}_2^T u))^T u)\} \\ &\quad + \tilde{r}^T \xi + \tilde{r}^T W_1^T \tilde{\sigma} + \tilde{r}^T \varepsilon \\ &\leq -\lambda_{\min}(K) \|\tilde{r}\|^2 + \mu \|\tilde{r}\| \cdot \|\tilde{W}\|_F (b_W - \|\tilde{W}\|_F) \\ &\quad + \|\tilde{r}\| \cdot \|\tilde{W}_2\|_F \cdot \|u\| \cdot (\sigma(\hat{W}_2^T u))^T u \\ &\quad + \tilde{r}^T \xi + \|\tilde{r}\| \cdot \|W_1\|_F \cdot \|\tilde{\sigma}\| + \|\tilde{r}\| \varepsilon_N \end{aligned}$$

where $\lambda_{\min}(K)$ is the minimum eigenvalue of K .

Applying $\|u\| \leq k_0 + k_1 \|r\|$ and substituting (22) into the above inequality yields

$$\begin{aligned} \dot{V} &\leq -\|\tilde{r}\| \{\lambda_{\min}(K) \|\tilde{r}\| - \mu \|\tilde{W}\|_F \cdot (b_W - \|\tilde{W}\|_F) \\ &\quad - \|\tilde{W}\|_F (k_0 + k_1 \|\tilde{r}\|) \sqrt{n_h}\} \\ &\quad + k_W (\|\tilde{W}\|_F + b_W) \|\tilde{r}\| - (\sqrt{n_h} b_W + \varepsilon_N) \|\tilde{r}\| \end{aligned}$$

where n_h is the number of the neurons in the hidden layer of the network, and the inequality $\|\sigma(\hat{W}_2^T u)\| < \sqrt{n_h}$ for the sigmoid activation has been used.

Choosing $k_W > k_1 \sqrt{n_h}$, we have

$$\begin{aligned} \dot{V} &\leq -\|\tilde{r}\| \{\lambda_{\min}(K) \|\tilde{r}\| - \mu \|\tilde{W}\|_F \cdot (b_W - \|\tilde{W}\|_F) \\ &\quad - k_0 \sqrt{n_h} \cdot \|\tilde{W}\|_F - (\sqrt{n_h} b_W + \varepsilon_N)\} \\ &\leq -\|\tilde{r}\| \{\lambda_{\min}(K) \|\tilde{r}\| + \mu (\|\tilde{W}\|_F - k_2)^2 - k_3\} \end{aligned}$$

where

$$\begin{aligned} k_2 &= \frac{\mu b_W + k_0 \sqrt{n_h}}{2\mu} = \frac{k_0 \sqrt{n_h}}{2\mu} + \frac{b_W}{2} \\ k_3 &= \mu k_2^2 + \sqrt{n_h} b_W + \varepsilon_N \end{aligned}$$

Therefore $\dot{V} < 0$ if $\|\tilde{r}\| > \frac{k_3}{\lambda_{\min}(K)}$, or equivalently,

$$\|r\| > \frac{k_3}{\lambda_{\min}(K) \|\Upsilon\|} \geq \frac{k_3}{\lambda_{\min}(K) b_\Upsilon} \triangleq b_r$$

or

$$\|\tilde{W}\|_F > k_2 + \sqrt{\frac{k_3}{\mu}} \triangleq b_{\tilde{W}}$$

Here b_r and $b_{\tilde{W}}$ denote the regions of convergence for the filtered tracking error and the weight estimation error respectively. \dot{V} becomes negative and V decreases outside the compact set defined by $\|r\| \leq b_r$ and $\|\tilde{W}\|_F \leq b_{\tilde{W}}$. According to the LaSalle extension of Lyapunov analysis, this implies that both r and \tilde{W} are uniformly ultimately bounded (UUB).

3.3 Bounds on Control Torques

It is pointed out in (Chiang et al., 1991; Moallem and Patel, 1999) that an M^3 system having a rigid micro manipulator attached at the end of a flexible macro manipulator can give nonminimum-phase characteristics under some configurations. For non-minimum-phase systems, perfect tracking or asymptotic tracking cannot be achieved by finite control inputs (Slotine and Li, 1991). Our control strategy for the M^3 system is to drive and maintain the tracking errors within an arbitrarily small vicinity around the nominal trajectory while keeping the magnitudes of the control input torques bounded. In the previous section, we have established the control torques and weight tuning rules for uniformly bounded tracking. It is then required to consider the boundedness of the control inputs as the filtered tracking error $r(t)$ becomes very small or as $r(t) \rightarrow 0$.

As shown in Section 3.2, both the target weights and their estimation errors are bounded. Therefore, the neural network estimate of the nonlinear function f is also bounded. Let $\|\hat{f}\| \leq b_f$ and $\|r\| \leq b_r$. By the Cauchy-Schwartz inequality we have

$$\begin{aligned}
\|\tau_M\| &\leq \|\hat{f}_1\| + \frac{\|\bar{r}_1\|\|\bar{r}_3\|}{\epsilon} (\|\hat{f}_3\| + \lambda_{max}(K_3)\|\bar{r}_3\|) \\
&\quad + \lambda_{max}(K_1)\|\bar{r}_1\| \\
&\leq b_f + \frac{b_r^2}{\epsilon} (b_f + \lambda_{max}(K_3)b_r) + \lambda_{max}(K_1)b_r \\
&\leq (1 + \frac{b_r^2}{\epsilon})^2 (b_f + \lambda_{max}(K)b_r) \triangleq b_M
\end{aligned}$$

and

$$\|\tau_m\| \leq \|\hat{f}_2\| + \lambda_{max}(K_2)b_r \leq b_f + \lambda_{max}(K)b_r \triangleq b_m$$

where $\lambda_{max}(K_i)$ is the maximum eigenvalue of K_i , $i = 1, 2, 3$.

4. SIMULATION RESULTS

To demonstrate the effectiveness of the proposed control scheme, we have applied it to a planar M^3 system with one flexible macro-arm and three rigid micro-arms as depicted in Figure 1. The flexible link has a thickness of 1.3 mm, width of 3.14 cm, length of 1 m, mass of 1 kg and flexural rigidity of 20 Nm². The rigid links have equal lengths of 0.2 m and equal masses of 0.2 kg.

The assumed-modes method with clamped-mass boundary conditions was used to model the M^3 system. Two orthonormal mode shapes were taken into account for simplifying the inertia and stiffness matrices. The resulting natural frequencies of vibration are 2.50 Hz and 15.6 Hz.

The desired trajectory $p_d = [x_d, y_d]^T$ for the end-effector is chosen as

$$\begin{aligned}
x_d &= 0.3 \sin(t) + 0.80 \\
y_d &= 0.3 \cos(t) + 0.39
\end{aligned} \tag{27}$$

which is a circle centered at (0.80, 0.39) with a radius of 0.3 m. The desired joint velocity $\dot{\theta}_d$ is given by

$$\dot{\theta}_d = J^+ \dot{p}_d + (I - J^+ J) \eta, \tag{28}$$

where J^+ is the pseudo-inverse of $[J_{\theta_M}, J_{\theta_m}]$ and η is an arbitrary 4×1 vector. The desired joint trajectory θ_d is obtained by integrating the above equation.

The neural network used for the simulation employs 30 input units for $u = [\hat{x}^T, \hat{x}^T, x_d^T, \dot{x}_d^T, \ddot{x}_d^T]^T$, 6 output units for $f = [f_1, f_2, f_3]^T$, and 10 hidden units with sigmoidal activation functions for achieving an adequate learning capability. This network has a total of 376 weights which require no prior knowledge of the parameters either of the M^3 system dynamics or of the control scheme, and were simply initialized at zero and updated on-line.

The parameters used in the simulation are listed as follows:

$$\begin{aligned}
K_i &= \text{diag}\{30, 10, 10, 10, 15, 15\}, \quad i = 1, 2, 3. \\
k_w &= \text{diag}\{5, 0.5, 2, 0.5, 10, 10\}, \quad P = 10I_{11},
\end{aligned}$$

$$\begin{aligned}
\Lambda &= \text{diag}\{10, 2, 2, 2, 15, 10\}, \quad R = 10I_{31}, \\
\mu &= 0.5, \quad b_w = 10, \quad \epsilon = 0.01, \quad \kappa = 1.0
\end{aligned}$$

Figures 3 to 6 compare the simulation results of tracking control of the M^3 system using the neural network based controller against those using a PD controller. The PD controller was implemented by shutting down the neural network in the outer loop of the system and using the state position and velocity feedback in the inner loop. It can be seen from Figures 3 to 5 that, under the proposed neural network based control scheme, the amplitudes of vibrations in the macro manipulator as well as the tip-position tracking errors of the micro manipulator rapidly converge to small regions around the origin while the magnitudes of the applied torques are kept bounded. In particular, the average amplitudes of the first two flexural modes are reduced by nearly 95% over the conventional PD joint controller. Note that there are additional vibrations in the macro manipulator induced by acceleration of the micro manipulator during 4.5 - 6.5 seconds. These vibrations are well suppressed by the neural network controller, though there remain small residual vibrations. In contrast, the PD controller is not capable of damping out these vibrations. Moreover, it can be observed from Figure 6 that there are significant oscillations in the control torques under the PD control due to its high-gain feature. It is clearly shown from the simulation results that use of the neural network significantly improves the tracking performance as the network learns more about the dynamics of the M^3 system.

5. CONCLUSIONS

In this paper, we have developed a novel control scheme for time-varying trajectory tracking control of a macro-micro manipulator (M^3) system based on neural networks. The control scheme allows us to constrain the tracking errors of the micro manipulator in the presence of vibrations due to the flexibility of the macro manipulator links within an arbitrarily small region around the origin by applying bounded control torques at the joints of the M^3 system. A neural network is designed to perform the learning and control tasks online simultaneously and no off-line training procedure is required for the neural network to identify the dynamic model of the M^3 system. The stability and convergence properties of the control scheme provide assurances of the reliability needed to make the controller feasible in practical real-time control. The performance of the control scheme is tested and compared to that of a PD controller by simulations on a three-link rigid micro manipulator attached at the tip of an one-link flexible manipulator. Future work

in this area will include implementation of the neural-network based strategy to a prototype M^3 system, currently under construction, consisting of a 4 degree-of-freedom macro manipulator and a 7 degree-of-freedom micro manipulator.

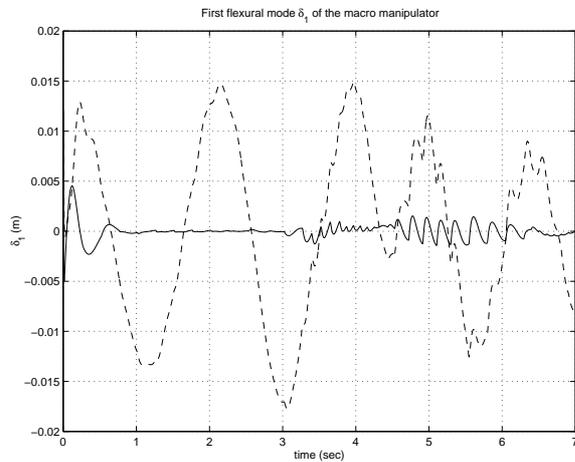


Fig. 3. First flexural mode of the macro manipulator (neural net: —, PD: - -)

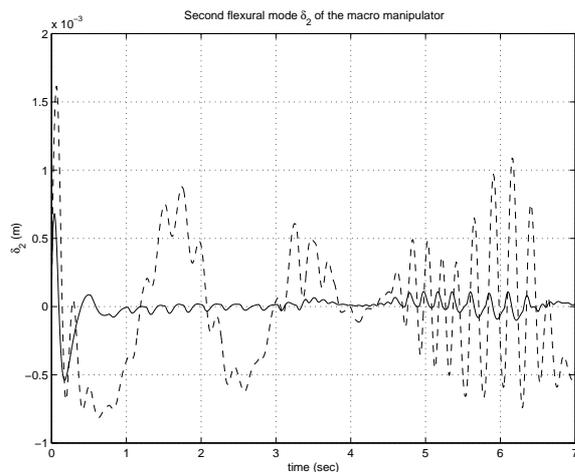


Fig. 4. Second flexural mode of the macro manipulator (neural net: —, PD: - -)

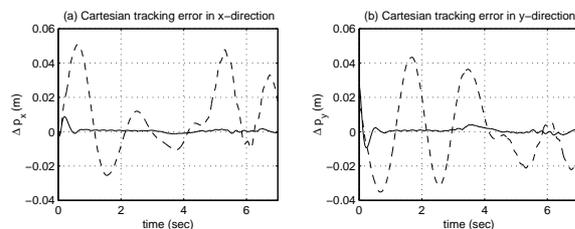


Fig. 5. Cartesian tracking errors (neural net: —, PD: - -): (a) in the x direction, (b) in the y direction

References

W.W. Chiang, R. Kraft, and R.H. Cannon. Design and experimental demonstration of rapid,

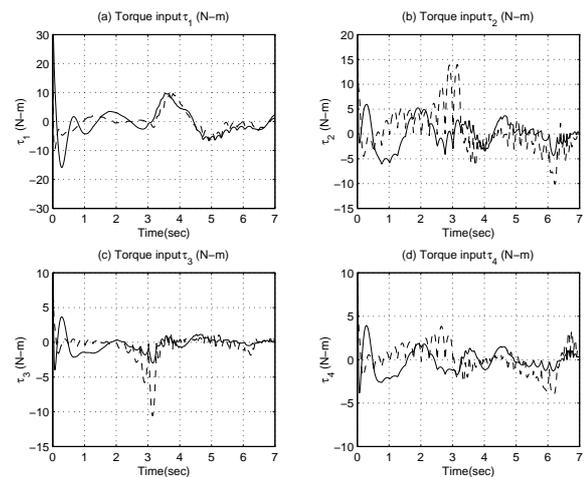


Fig. 6. Control torques vs. time (neural net: —, PD: - -): (a) τ_1 (Nm) (for macro (flexible link)), (b) τ_2 (Nm), (c) τ_3 (Nm), (d) τ_4 (Nm).

precise end-point control of a wrist carried by a very flexible manipulator. *Int. J. Robotics Research*, 10:30–40, 1991.

Z.H. Jiang and A.A. Goldenberg. Dynamic end-effector trajectory control for flexible micro-macro manipulators using an ideal manifold. *Japanese Society of Mechanical Engineering (JSME) Int. J. Series C*, 41:269–277, 1998.

J.Y. Lew and D.J. Trudnowski. Vibration control of a micro/macro-manipulator system. *IEEE Control System Magazine*, 16:26–31, 1996.

D.P. Magee and W.J. Book. Filtering micro-manipulators wrist commands to prevent flexible base motion. In *Proc. American Control Conference*, pages 1836–1842, Seattle, WA, 1995.

M. Moallem and R.V. Patel. A vibration control strategy for a boom-mounted manipulator system for high-speed positioning. In *Proc. IEEE Int. Conf. Intelligent Robotics and Systems*, pages 299–304, Takamatsu, Japan, 1999.

J.J.E Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, Englewood-Cliffs, NJ, 1991.

M.A. Torres, S. Dubowsky, and A.C. Pisoni. Vibration control of deployment structures' long-reach space manipulators: The P-PED method. In *Proc. IEEE Int. Conf. on Robotics Automation*, pages 2498–2504, Minneapolis, MN, 1996.

A. Yesildirek and F.L. Lewis. A neural net controller for robots with Hebbian tuning and guaranteed stability. In *Proceedings of American Control Conference*, pages 2784–2789, Seattle, WA, June 1995.

T. Yoshikawa, K. Hosoda, T. Doi, and H. Murakami. Dynamic trajectory tracking control of flexible manipulator by a macro-micro manipulator system. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1804–1809, San Diego, CA, June 1994.