Real-Time Vibration Control of an Industrial Manipulator Mounted on a Compliant Base

H. Bassan[1], H.A. Talebi[2], R.V. Patel[1] and M. Moallem[1]

[1] Department of Electrical & Computer Engineering, University of Western Ontario
London, Ontario, Canada, N6A 5B9
hsbassan@uwo.ca, rajni@eng.uwo.ca, mmoallem@engga.uwo.ca

[2] Department of Electrical Engineering, AmirKabir University of Technology
Tehran, Iran, 15914
alit@cic.aku.ac.ir

***Abstract*— Structural flexibility is an inherent characteristic of a class of macro-micro manipulators consisting of a rigid micro manipulator mounted on a long-reach (flexible) macro manipulator. Vibrations caused by flexibility make it difficult to achieve accurate control of the end-point of the micro manipulator. In this paper, we develop a control strategy that can be applied to such a system. An experimental test-bed has been developed in which a 6 DOF PUMA 560 manipulator is mounted on a compliant platform. The control strategy consists of a rigid body inverse dynamics controller together with a neural network based strategy for damping out the oscillations due to the flexible base. Experimental results obtained from the test-bed are presented to show the effectiveness of the proposed control scheme.**

Keywords: Vibration control, real-time, neural network based control, flexible systems

I. Introduction

Several applications of robotic systems require a manipulator to operate from a compliant base. Examples of such manipulators are long-reach robotic systems for nuclear environments, robots used in high voltage live-line operations, and small robots mounted on large robots called macro-micro manipulators. Macro-micro (abbreviated hence forth as M-m) manipulators were proposed for long reach tasks requiring speed and precision. As mentioned in [1], active vibration control by joint actuators of the macro part may reduce structural vibrations, but its performance may be limited by the usually small actuator bandwidths. Also higher modes can be excited in a complex multiple-link macro. Compliance can result in errors at the end effector of the manipulator and may destabilize the system because of the possibility of the non-minimum phase characteristics associated with flexible-link manipulators. Vibration control of M-m manipulators has been considered by a number of researchers in recent years, e.g., see [2]–[8].

Obtaining a desired performance for a M-m manipulator requires fairly accurate modeling of the flexible base and compensating for the deflection at the end point of the rigid robot. However, it is often difficult to obtain an accurate model of flexibility. Thus model-based methods may require extensive system identification and tuning in order to achieve a desired performance. The nonlinear mapping properties of neural networks, their adaptive nature, and their ability to deal with uncertainty make them a powerful tool for the control of nonlinear systems. During the last decade, several neural network approaches have been developed for control of flexible-link manipulators, e.g., see [9]. However, not much work has been reported on practical implementation of neural network control schemes especially for M-m manipulator systems (see [10]). In [11]–[14], several approaches have been developed for identification and control of nonlinear systems based on stability theory. Most of these results, however, are too complicated to be used in an experimental framework. Moreover, some of them assume that the nonlinear system is linear in parameters. Such an assumption is often not valid for manipulators with flexibility.

In this paper, we present a neural-network based controller for vibration control of a PUMA manipulator mounted on a compliant base. The control strategy consists of the manipulator rigid body inverse dynamics for tracking the joints and a neural network controller added to damp out the oscillations at the base of the manipulator. Experimental results are presented for a test bed consisting of a PUMA560 robot mounted on a compliant base as depicted in Figure 1.

II. Dynamic Modeling

In this section, a mathematical model of the system is given. A 6 DOF PUMA 560 manipulator is mounted on an X-shaped platform as shown in Figure 1. The platform is

This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grants STPGP 215729-98, RGPIN-1345 and RGPIN-227612-00

supported by 4 springs to emulate a manipulator system mounted on a flexible base. The motion of the links of the manipulator results in oscillations of the platform. The essential idea to model the oscillatory behavior of the base is to consider the base motion as a 3 DOF unactuated manipulator with zero link lengths. Towards this end, one prismatic joint (γ) and two revolute joints ($\alpha \& \beta$) are considered as shown in Figure 3. The prismatic joint emulates the translational motion in the vertical direction, and the two revolute joints emulate angular motions about two planar axes. The lateral motions of the platform (in the horizontal plane) were found to be significantly smaller than the vertical deflections and were therefore neglected. Flexible-link manipulators do exhibit 6 DOF motion in general, but the measurement and control of six elastic DOF's have complications of their own and future work will involve addressing those issues. Figure 3 also shows the frame assignments for the joints. Hence, the full inverse dynamics of the manipulator system can be obtained by using a standard algorithm such as the Newton-Euler approach. The dynamic equation of motion takes the general form for manipulators with flexibility, given by

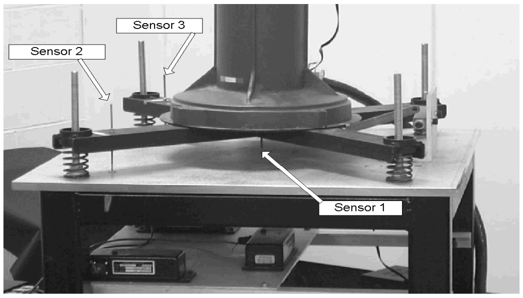

Fig. 1. Close up of the platform.

$$M(\delta,\theta)\left[\begin{array}{c}\ddot{\delta}\\ \ddot{\theta}\end{array}\right]+\left[\begin{array}{c}h_1(\theta,\dot{\theta},\delta,\dot{\delta})+K\delta\\ h_2(\theta,\dot{\theta},\delta,\dot{\delta})\end{array}\right]=\left[\begin{array}{c}0\\ u\end{array}\right], \qquad (1)$$

where θ is the 6×1 vector of the joint variables, $\delta = [\gamma\ \alpha\ \beta]^T$ is the 3×1 vector of the deflection variables, h_1 and h_2 are the terms due to gravity, Coriolis, and centrifugal forces, M is the positive–definite symmetric inertia matrix, and K is the positive-definite diagonal stiffness matrix.

III. Control Methodology

The objective in this section is to define a control scheme such that the joints of the micro manipulator track the desired trajectory while the oscillations of the base are damped out quickly. This objective is not easy to achieve since the system under control is an underactuated mechanical system. It is assumed that there exist uncertainties in the dynamic parameters of the manipulator which are to be compensated for by a neural network controller. The control structure is shown in Figure 2. As the figure shows, the control torque u has two components, $u = u_c + u_d$, where u_c is the rigid body controller for joint tracking and u_d is the torque required to damp out the vibrations. The rigid body control is the standard inverse dynamics (computed torque) controller given by

$$\begin{aligned}\acute{u} &= \ddot{\theta}_d + K_v\dot{E} + K_pE\\ u_c &= M(\theta)\acute{u} + h_2(\theta,\dot{\theta}) \qquad (2)\end{aligned}$$

where $E = \theta_d - \theta$ and $M(\theta),\ h_2(\theta,\dot{\theta})$ are obtained from (1) by considering only the joint motions.

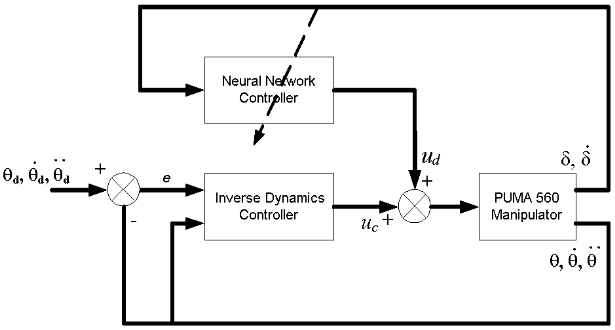
Neural Network Controller
Inverse Dynamics Controller
PUMA 560 Manipulator
$\theta_d, \dot{\theta}_d, \ddot{\theta}_d$
+
-
e
u_d
+
u_c +
$\delta, \dot{\delta}$
$\theta, \dot{\theta}, \ddot{\theta}$

Fig. 2. The schematic of the proposed neural network controller.

A. Deflection Control using a Neural Network

The neural network has been employed only for vibration suppression, as the rigid-body inverse dynamics controller for the PUMA 560 has been reported to yield good performance. Besides, implementation of a neural network to control the whole system's dynamics would result in significantly higher computational cost, thereby making real-time control harder to achieve.

In this section, the control structure specified in the previous section is modified to incorporate the base deflection in the control action. One of the main limitations of the computed torque control strategy is that there is no direct way to effectively damp out the elastic vibrations at the base of the manipulator. In other words, controlling the joints does not necessarily guarantee satisfactory behavior for the end-effector position. Our proposed approach attempts to overcome the above difficulty. To consider the effect of the base motion in the control input, first, a simple PD type controller is designed:

$$u_d = -K_{dp}\delta - K_{dv}\dot{\delta} \qquad (3)$$

However, there is no standard method to properly adjust the gains K_{dp} and K_{dv} even if the models of the base and the manipulator are accurately known; so an *ad-hoc* procedure is often used to tune these gains. In many cases, this does not result in satisfactory performance. Experimental results obtained using *the best possible* combination of PD gains reported in Section V clearly demonstrate this fact.

The nonlinear mapping properties of neural networks and their adaptive nature are central to their use in controller design. Figure 2 shows the schematic of the proposed controller. The key to designing this controller is to include the base elastic deflections directly in the objective function of the neural network. This amounts to defining the objective function of the neural network as $J = \frac{1}{2}(\delta^T K_1 \delta + \dot{\delta}^T K_2 \dot{\delta})$, where $\delta = [\gamma\ \alpha\ \beta]^T$. Consequently, direct control over the elastic vibrations of the flexible modes becomes feasible through K_1 and K_2. Experimental results shown in Section V reveal that good control over the elastic deflection can be obtained by this modification. The inputs to the network are δ and $\dot{\delta}$, and the output of the network is the control signal u_d. The weight adjustment mechanism is based on the steepest descent method, namely $\dot{\mathbf{w}} = -\eta(\frac{\partial J}{\partial \mathbf{w}})^T$, where $\mathbf{w}$ is the vector of the weights of the network and η is the learning rate. Now $\frac{\partial J}{\partial \mathbf{w}}$ is computed according to

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial \delta}\frac{\partial \delta}{\partial \mathbf{w}} + \frac{\partial J}{\partial \dot{\delta}}\frac{\partial \dot{\delta}}{\partial \mathbf{w}} = \delta^T K_1 \frac{\partial \delta}{\partial \mathbf{w}} + \dot{\delta}^T K_2 \frac{\partial \dot{\delta}}{\partial \mathbf{w}}.$$

By using $\frac{\partial \delta}{\partial \mathbf{w}} = \frac{\partial \delta}{\partial u}\frac{\partial u}{\partial \mathbf{w}}$, $\frac{\partial \dot{\delta}}{\partial \mathbf{w}} = \frac{\partial \dot{\delta}}{\partial u}\frac{\partial u}{\partial \mathbf{w}}$, and $u = u_c + u_d$, $u_d = \Phi(\delta, \dot{\delta})$, where $\Phi(.)$ is the mapping performed by the neural network, we can write $\frac{\partial J}{\partial \mathbf{w}}$ as

$$\frac{\partial J}{\partial \mathbf{w}} = (\delta^T K_1 \frac{\partial \delta}{\partial u} + \dot{\delta}^T K_2 \frac{\partial \dot{\delta}}{\partial u})\frac{\partial \Phi}{\partial \mathbf{w}},$$

Now, $\frac{\partial \Phi}{\partial \mathbf{w}}$ can be computed using the backpropagation method, and $\frac{\partial \delta}{\partial u}$ and $\frac{\partial \dot{\delta}}{\partial u}$ denote the gradient of the plant. However, since it is assumed that little knowledge about the flexible dynamics is available, it is difficult to obtain an analytical expression for the plant Jacobian. Saerens and Soquet [15] suggested the use of the sign of the Jacobian instead of its real value for the training of neural adaptive controllers. This is often available simply from qualitative knowledge of the system (see Section V for details). The plant backpropagation equation then becomes:

$$\frac{\partial J}{\partial \mathbf{w}} = (\delta^T K_1 SGN(\frac{\partial \delta}{\partial u}) + \dot{\delta}^T K_2 SGN(\frac{\partial \dot{\delta}}{\partial u}))\frac{\partial \Phi}{\partial \mathbf{w}},$$

where SGN(.) denotes the sign of (.).

A two-layer backpropagation neural network was used for the experiment with 6 neurons in the hidden layer and 2 neurons in the output layer. The hidden layer neurons have sigmoidal transfer functions and the output neurons use linear activation functions. Note that the neural network is trained and employed as an online controller and no off-line training is required.

IV. Experimental Setup

In this section, the test-bed used in this research is described in some details. As mentioned in Section II, the platform was modeled as three unactuated joints- one prismatic and two revolute.

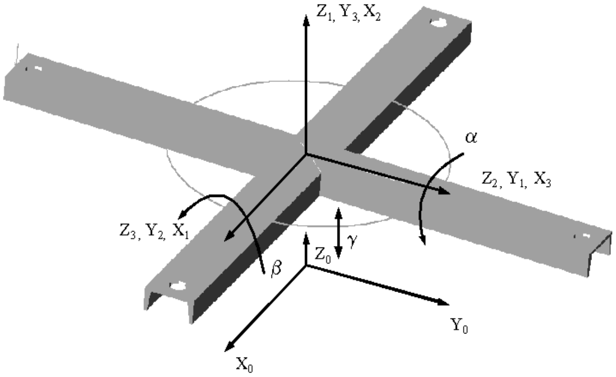

Fig. 3. Frame assignment for the manipulator base.

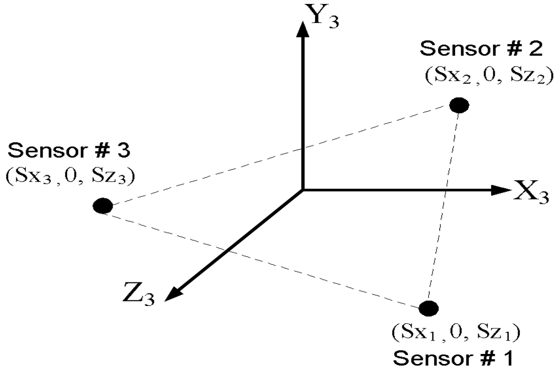

Fig. 4. Sensor locations for the base.

A. Optimal Sensor Location

The frame assignment for the platform is indicated in Figure 3. $Frame\{0\}$ is the stationary frame attached to the middle of the table. Figure 4 shows the schematic diagram of the sensor locations. The platform was instrumented using three resistive type "rope" sensors [16]. Each sensor provides position and velocity information in one direction. The transformation matrices relating the sensor frame $\{s_i\}, (i = 1, 2, 3)$ to $frame\{3\}$ and $frame\{0\}$ are given by

$$ {}^{3}_{s_i}T = \left[\begin{array}{ccc|c} & & & S_{xi} \\ & I_3 & & 0 \\ & & & S_{zi} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{4}$$

and

$$ {}^{0}_{s_i}T = \left[\begin{array}{c|c} {}^{0}_{3}R & {}^{0}_{3}R \begin{bmatrix} S_{xi} \\ 0 \\ S_{zi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \gamma \end{bmatrix} \\ \hline 0 \quad 0 \quad 0 & 1 \end{array}\right] \tag{5}$$

where S_{xi} and $S_{zi}, i = 1, 2, 3$, are the coordinates of the corresponding sensors as shown in Figure 4. The corresponding position vector of each sensor expressed in $frame\{0\}$ is given by

$$
{}^0S_i = \begin{bmatrix} \sin\alpha\sin\beta S_{xi} - \cos\alpha S_{zi} \\ \cos\beta S_{xi} \\ -\cos\alpha\sin\beta S_{xi} - \sin\alpha S_{zi} + \gamma \end{bmatrix} \tag{6}
$$

Assuming small deflections in the platform, we can write (6) as

$$
{}^0S_i = \begin{bmatrix} -S_{zi} \\ S_{xi} \\ -\beta S_{xi} - \alpha S_{zi} + \gamma \end{bmatrix} \tag{7}
$$

As can be seen from Figure 1, three rope sensors were vertically mounted on the fixed table. Each sensor measures the Z component of the above vector, i.e.,

$$
\begin{aligned}
-S_{z1}\alpha - S_{x1}\beta + \gamma &= m_1 \\
-S_{z2}\alpha - S_{x2}\beta + \gamma &= m_2 \\
-S_{z3}\alpha - S_{x3}\beta + \gamma &= m_3
\end{aligned} \tag{8}
$$

where $m_i, i = 1,2,3$, are the sensor readings. The above equations can be written in the form

$$
A \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} \tag{9}
$$

where

$$
A = \begin{bmatrix} -S_{z1} & -S_{x1} & 1 \\ -S_{z2} & -S_{x2} & 1 \\ -S_{z3} & -S_{x3} & 1 \end{bmatrix} \tag{10}
$$

To obtain the optimal sensor locations, we find values of S_{xi} and $S_{zi}, i = 1,2,3$, such that the condition number of A is minimized. This guarantees that α, β and γ obtained by inverting A in (9) are less sensitive to measurement noise. This results in $S_{x1} = 0,\ S_{z2} = 0,\ S_{z3} = 0,\ S_{x3} = 0$. The Z component in equation (6) for the 3 sensors then simplifies to

$$
\begin{aligned}
-S_{z1}\sin\alpha + \gamma &= m_1 \\
-S_{x2}\cos\alpha\sin\beta + \gamma &= m_2 \\
\gamma &= m_3
\end{aligned} \tag{11}
$$

Now equation (11) can be solved to obtain α, β and γ as follows

$$
\begin{aligned}
\gamma &= m_3 \\
\alpha &= \sin^{-1}\left(\frac{m3 - m1}{S_{z1}}\right) \\
\beta &= \sin^{-1}\left(\frac{\gamma - m2}{S_{x2}\cos\alpha}\right)
\end{aligned} \tag{12}
$$

B. Real Time Implementation

The original robot controller was retrofitted to enable access to the low-level parameters of the robot. A Multi-Q multipurpose data acquisition card from Quanser Consulting was chosen to interface the controller with the control computer. A microcontroller card was designed to monitor the emergency switch and control the brakes in case of an emergency.

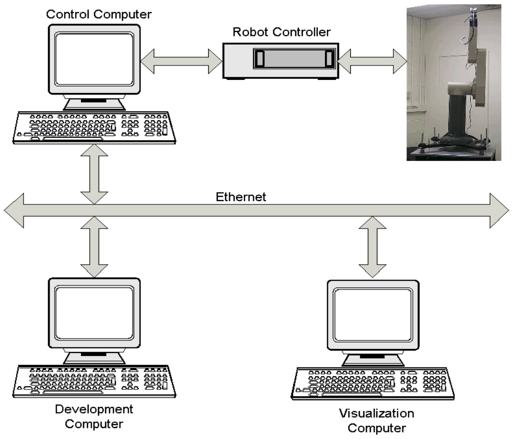

Fig. 5. System Block Diagram.

The Simulink block diagram of the controller was compiled using the Real-Time Workshop [17] to generate C code, which runs in real-time on the control computer. The control computer is an Intel Pentium III 500Mhz PC running the VxWorks RTOS [18]. Figure 5. shows the block diagram for this experimental setup. A sampling rate of 64Hz was chosen for the experimental system.

V. Experimental Results

The control law (3) was modified to take into account only qualitative knowledge about the system dynamics. In this case, only the sign of the gradient of the system was considered. The modified controller is given by

$$
u_d = (-K_{dp}\delta - K_{dv}\dot{\delta})S_{\hat{g}} \tag{13}
$$

where $S_{\hat{g}}$ is a matrix consisting of 0,1,-1 and denotes the signs of the gradients of the system

$$
S_{\hat{g}} = \begin{bmatrix} SGN(\frac{\partial\gamma}{\partial u_1}) & \cdots & SGN(\frac{\partial\dot{\beta}}{\partial u_1}) \\ \vdots & \ddots & \vdots \\ SGN(\frac{\partial\gamma}{\partial u_6}) & \cdots & SGN(\frac{\partial\dot{\beta}}{\partial u_6}) \end{bmatrix} \tag{14}
$$

The matrix $S_{\hat{g}}$ is obtained by simply moving the robot and noticing the direction of the deflection variables with respect to the corresponding joint torques. It was observed that the motions of joints 2 and 3 have the most dominant effect on the platform vibrations. Hence the corresponding elements for the other joints in $S\hat{g}$ were set to zero as shown below.

$$
S_{\hat{g}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{15}
$$

The proposed control algorithm was implemented on the experimental setup mentioned in the previous section.

First, only rigid body control, i.e., $u_d = 0$, was implemented with a fairly slow trajectory. For slower trajectories, the performance of this controller was found to be within reasonable limits. However, as the speed of the desired trajectories is increased, adjustment of these gains becomes more difficult and the system performance degrades significantly. At this speed, for smaller values of K_{dv} and K_{dp}, the vibration suppression is almost negligible. Figure 7 shows the system performance in this case. As the gains are increased, the system tends to become unstable and the joint errors increase significantly.

Next, the proposed neural network based controller shown in Figure 2 was implemented. The faster trajectory used in the previous experiment, was used with this controller. As can be observed, the oscillations at the base are damped out very quickly (see Figure 8). The joint tracking in this case was not affected either.

To study the effects of these oscillations at the end-effector of the manipulator, another trajectory was applied. This trajectory is a closed contour in the X-Z plane. A joint-space trajectory was computed for this contour. End-point calculations were performed using the kinematic model for the PUMA 560 and including the compliant base. Figure 9 shows the oscillations at the base for this trajectory. Figure 10 shows the motion of the end-effector of the manipulator in Cartesian space. As can be seen, a significant improvement in end-point tracking was obtained using the neural-network controller as compared to the conventional inverse dynamics controller.

VI. Conclusions

The problem of controlling a manipulator mounted on a compliant platform was considered in this paper. A control strategy was proposed which consists of a rigid body inverse dynamics controller with a neural network based controller to damp out the oscillations. Control software was developed with Matlab/Simulink based graphical tools in conjunction with the VxWorks RTOS for implementing real-time control. Experimental results obtained by implementing the neural network based controller show the effectiveness of the proposed control scheme.

References

[1] J.Y. Lew and W.J. Book, "Hybrid control of flexible manipulators with multiple contacts," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 242–247, 1994.

[2] D.P. Magee and W.J. Book, "Filtering micro-manipulator wrist commands to prevent flexible base motion," in *Proc. American Control Conference*, pp. 924–928, Seattle, WA, 1995.

[3] T. Yoshikawa, K. Harada and A. Matsumoto, "Hybrid position/force control of flexible-macro/rigid-micro manipulator systems," *IEEE Trans. on Robotics and Automation, Vol. 12, No. 4*, pp. 633–639, 1996.

[4] W-W Chiang, R. Kraft and R.H Cannon, "Design and experimental demonstration of rapid, precise end-point control of a wrist carried by a very flexible manipulator," *International Journal of Robotics Research, Vol. 10, No. 1*, pp. 30–40, 1991.

[5] 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 International Conference on Robotics and Automation, Vol. 12, No. 4*, pp. 2498–2504, Minneapolis, MN, 1996.

[6] 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 Robots and Systems*, pp. 299–304, 1999.

[7] J.Y. Lew and S.M. Moon, "A simple active damping control for compliant base manipulators," *IEEE/ASME Transactions on Mechatronics, Vol. 6*, pp. 305–310, Sep. 2001.

[8] Mavroidis, C., Dubowsky, S. and Thomas, K., "Optimal Sensor Location in Motion Control of Flexibly Supported Long Reach Manipulators," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME,* Vol. 119, No. 4, pp. 718–726, 1997.

[9] H.A. Talebi, R.V. Patel and K. Khorasani, *Control of Flexible Link Manipulators Using Neural Networks.* Springer-Verlag, Heidelberg: Lecture Notes in Control and Information Sciences 261, 2001.

[10] X.P. Cheng and R.V. Patel, "Neural network based tracking control of a flexible macro-micro manipulator system," *Neural Networks, Vol. 16*, pp. 271–286, 2003.

[11] Lewis, F.L., K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks,* Vol. 6, No. 3, pp. 703–715, 1995.

[12] Polycarpou, M.M., "Stable adaptive neural control scheme for nonlinear systems," *IEEE Trans. Automatic Control,* Vol. 41, No. 3, pp. 447–451, March 1996.

[13] Sanner, R.M. and J.-J. Slotine, "Stable adaptive control and recursive identification using radial gaussian networks," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 2116–2123, 1991.

[14] Sadegh, N., "A perceptron network for functional identification and control of nonlinear systems," *IEEE Trans. Neural Networks,* Vol. 4, No. 6, pp. 982–988, Nov. 1993.

[15] M. Saerens and A. Soquet, "A neural controller," in *Proc. of the 1st IEEE Conference on Neural Networks*, pp. 211–215, 1987.

[16] *Displacement/Velocity Transducer, Model No. DVT-5A.* Psi-Tronix, 3950, South "K" Street, Tulare, CA, USA 93274-7155.

[17] *Real-Time Workshop User's Guide.* MathWorks, Natick, MA, Sep. 2000.

[18] *VxWorks Programmer's Guide.* 500 Wind River Way, Alameda, CA, USA 94501.

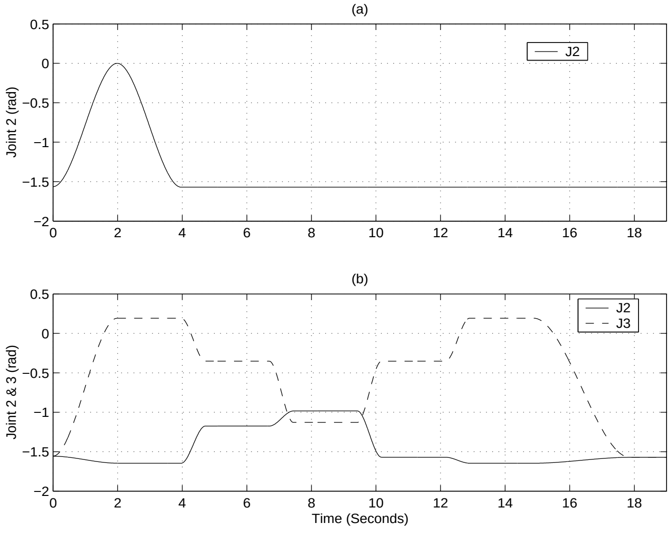

Fig. 6. Trajectories used for the experiment. (a) Trajectory T_1 (b) Trajectory T_2.

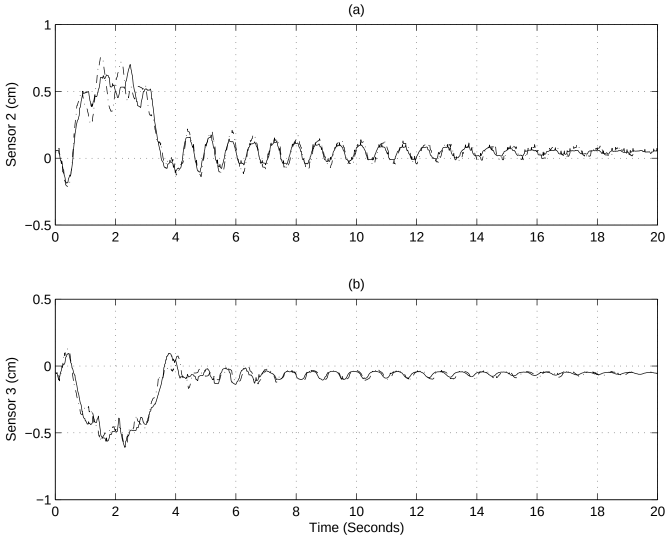

Fig. 7. Experimental system response to trajectory T_1 using u_d Controller. (a) Sensor 2 (b) Sensor 3. The dashed lines correspond to $K_{dp} = K_{dv} = 0$ and the solid lines correspond to $K_{dp} = 8, K_{dv} = 0$.

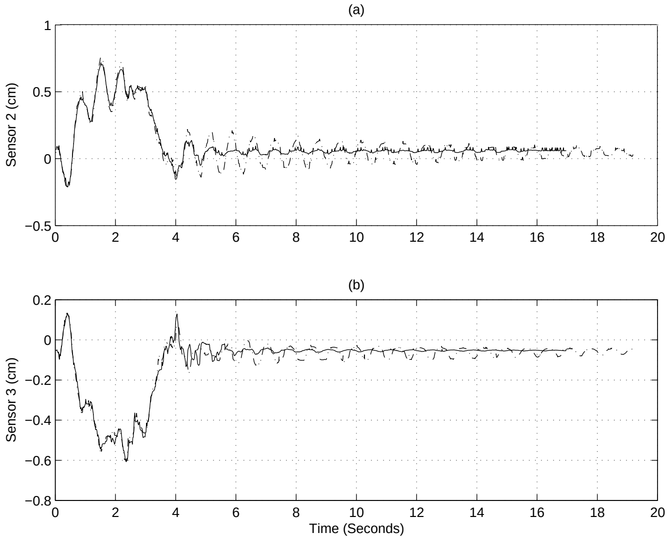

Fig. 8. Experimental system response to trajectory T_1 using the neural network controller. (a) Sensor 2 (b) Sensor 3. The solid lines correspond to the neural network controller and the dashed lines correspond to the inverse dynamics controller ($K_{dp} = K_{dv} = 0$).

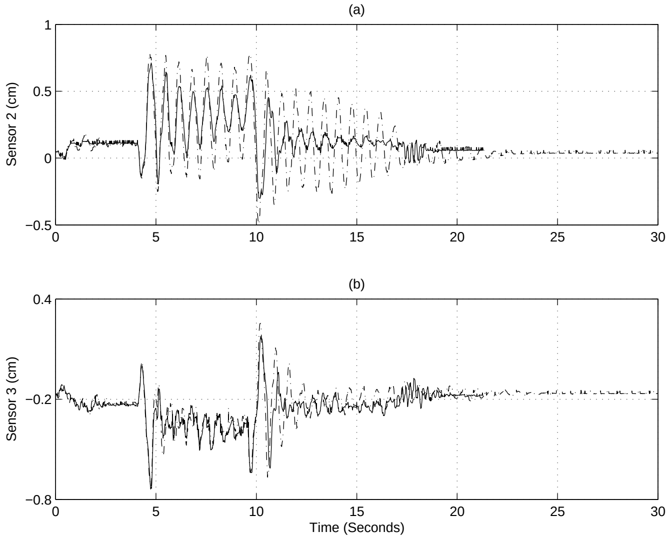

Fig. 9. Experimental system response to trajectory T_2 using the neural network controller. (a) Sensor 2 (b) Sensor 3. The solid lines correspond to the neural network controller and the dashed lines correspond to the inverse dynamics controller ($K_{dp} = 4, K_{dv} = 0.25$).

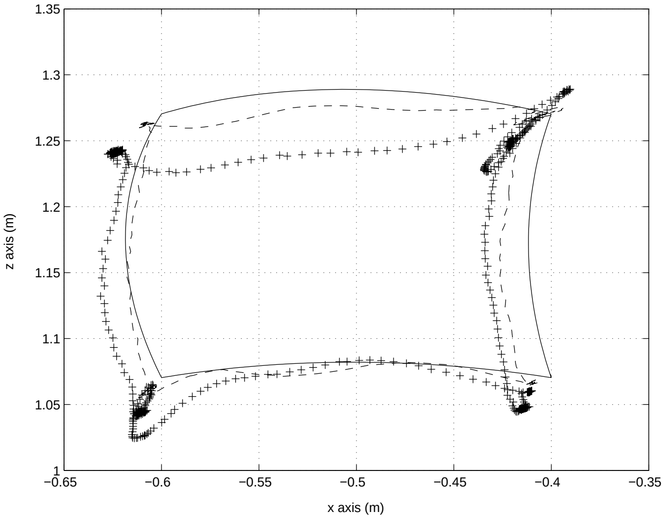

Fig. 10. Experimental system response to trajectory T_2 in Cartesian space (a) With the neural network controller (dash). (b) Without neural network (+) ($K_{dp} = 4, K_{dv} = 0.25$). The solid line corresponds to the desired trajectory.