Intelligent Automation and Soft Computing, Vol. 5, No. 3, pp. 171-190, 1999

NEURAL NETWORK LEARNING CONTROL FOR POSITION AND FORCE TRACKING OF MULTI-MANIPULATOR SYSTEMS

Peter C.Y. Chen and James K. Mills
*Department of Mechanical
and Industrial Engineering
University of Toronto
5 King's College Road, Toronto
Ontario, Canada M5S 3G8*

Kenneth C. Smith
*Department of Electrical
and Computer Engineering
University of Toronto
10 King's College Road, Toronto
Ontario, Canada M5S 1A4*

ABSTRACT—In this article, an approach to improving the performance of multi-manipulator systems using neural networks is presented. This approach is formulated in the constrained motion framework, within which a nominal feedback control augmented by a neural network is derived. It is shown that the closed-loop system with the neural network learning on-line is stable in the sense that all signals in the systems are bounded. It is further proved that the performance of the multi-manipulator system is improved in the sense that the "size" of a certain error measure decreases as the learning process of the neural network is iterated. Results of computer simulations conducted to verify the analytical conclusions are presented. The results of this work suggest that neural networks could be used as "add-on" control modules to improve the performance of industrial robots in execution of tasks involving two or more cooperative manipulators.

Key Words: multi-manipulator control, neural networks, simultaneous position and force tracking, performance improvement

1. INTRODUCTION

In automated manufacturing, robots are most often employed for material transport such as pick-and-place operations. The capability of robots, however, far exceeds that required to execute these mundane tasks. By fully utilizing the capability of robots, greater economic benefits can be realized. An immediate example is the so-called fixtureless assembly [15], where, for instance, two robots hold two parts together while the parts are being bonded (possibly by using a third robot). Once it becomes a proven technology, fixtureless assembly could bring the automotive industry drastic reductions in fixture-replacement (i.e., retooling) costs necessitated by the annual upgrade of existing models or introduction of new product lines. The key element of a fixtureless assembly system is the coordination of two or more robots. Such coordination not only involves the positioning of the individual robots, but, more importantly, the dynamic interaction among the robots through the transmission of force.

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., [1, 12] and the hybrid force/position control method (e.g., [8,11]). Often implicit in these approaches is the assumption that the dynamic parameters of the manipulators involved are known precisely. Such an assumption, however, can not be satisfied in practice. To circumvent this difficulty, other control strategies, such as adaptive control [10] and robust control [6], have been proposed to deal with parameter uncertainty associated with the manipulators.

A class of computational models known as neural networks has been applied to system control in general and to robot control in particular. Justification for using neural networks for robot control is based on the following properties of neural networks [9]: (i) the ability of neural networks to "learn" (through a repetitive training process) enables a controller incorporated with a neural network to improve its performance; (Learning in general refers to the process where a neural network, having been stimulated by an input signal, adjusts its internal parameters so as to produce a desired output signal.) (ii) the ability of a neural network to "generalize" what it has learned enables the controller also to respond to unexpected situations; and (Generalization in refers the ability of a neural network to produce a "meaningful" output when stimulated by an input signal that has not been used during learning.) (iii) the structure of neural networks allows massive parallel processing, especially when the neural networks are implemented in hardware using VLSI technology (e.g., [14]); such inherent collective processing capability enables a neural network to respond quickly in generating timely control actions.

It is due to these properties that a neural-network-based approach to uncertainty compensation could be considered potentially advantageous over other approaches, such as robust control and adaptive control. The learning ability of neural networks is especially desirable in the control of robots that perform repetitive manufacturing tasks. One reason for using robots instead of human workers in manufacturing is that robots can perform repetitive tasks with better quality and consistency. Unavoidable in repetitive robotic operations in an industrial setting, however, is the sustained "wear-and-tear" (e.g., joint friction, wear of gears, etc., [2]) of the robot. Such wear-and-tear inevitably affects the dynamic characteristic of the robot. In other words, the wear-and-tear introduces uncertainty into the robotic system, and consequently degrades its performance. A neural network that learns (iteratively) to compensate the effect of such wear-and-tear would enable the manipulator to maintain satisfactory performance consistently throughout its expected lifetime.

Recently neural networks have been employed in the area of robotic control. Application of neural networks to free motion control and contact task control have been reported. It has been demonstrated that such properties of neural networks can be utilized in developing performance-enhancing controller modules for various robotic applications, such as robot trajectory tracking (e.g., [17] and contact task control (e.g., [17]). In the area of multi-manipulator control, an approach using a neural network has been proposed in [20], where the neural network is used in conjunction with a hybrid force/position control scheme. In that approach, both force and position (plus its higher order time derivatives) signals are used as input to the neural network to generate a compensating torque signal. The issues pertaining to closed-loop stability and performance improvement with respect to the neural network learning process, however, are not addressed rigorously.

In this article, a different approach for uncertainty compensation using a neural network in multi-manipulator system control is proposed. 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. This formulation also leads to the conclusion that it is not necessary to use force measurement as part of the input to the neural network for uncertainty compensation, and as a consequence, the dimension of the input layer of the neural network is reduced.

In this proposed approach, uncertainty in the multi-manipulator system is quantified, and a neural network is used to reduce the effect of the uncertainty on the robotic system so that performance improvement can be achieved. Closed-loop stability of the multi-manipulator system (incorporated with a neural network) is analyzed using techniques from nonlinear system theory [21]. Results of the analysis show that the closed-loop system is stable in the sense that all signals in the system are bounded. Stability of closed-loop systems embedded with neural networks has been a key issue that has not been adequately addressed in the literature. The analyses presented in this paper offer a possible method in dealing with this issue. The performance of the multi-manipulator system with the neural network learning on-line is subsequently analyzed. Through this analysis, the effect of the dynamics of the neural network on the performance of the robotic system is revealed. It is shown that the performance of the multi-manipulator system is improved as the learning process of the neural network is iterated. Numerical simulations are conducted. The results of the simulations confirm the conclusions of the theoretical analysis.

It is emphasized that the main objective of this work has been to investigate the effectiveness of the learning ability of neural networks for improving the performance of multi-manipulator systems. It is based on the premise is that, by using a neural network as an "add-on" module, the performance of the robotic system under some "nominal" control (e.g., hybrid position/force control) can be enhanced. The focus of this work is on the analytical and empirical verification (through computer simulation) of this premise.

A logical extension of this investigation is to compare the effectiveness of this proposed approach with that of other approaches, such as adaptive control and robust control. To make such comparisons quantitatively, however, requires (as a prerequisite) an understanding of the dynamics of the neural network "add-on" module in conjunction with the "nominal" control. The work reported in this article represents an attempt in developing such an understanding.

The remainder of this article is organized as follows. 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 stability and performance analyses. Section 6 describes the computer simulations. Section 7 summarizes the results and discusses their implications.

2. DYNAMICS

The class of robotic systems under consideration consists of N cooperative manipulators handling a common object. The manipulators are identified as manipulator. The interaction among these manipulators is represented by the generalized forces exerted on the payload by the manipulators. (In this article, the term "generalized force" refers to the force and torque generated by the manipulators. Sometimes the word "generalized" is omitted when the nature of the force is clear from the context.) The equations of motion of a manipulator i (i=1, ..., N) with n_i joints can be expressed as

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

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

The following assumptions are made regarding the class of systems under consideration: (i) the object is jointless and is rigidly held by the manipulators, (ii) the mass and inertia of the object can be lumped into the last links of the manipulators, and (iii) each manipulator is completely rigid. Let $n=\sum_{i=1}^{N} n_i$, q = $(q_1, q_2, \ldots, q_i, \ldots, q_N)^T \in R^{n\times 1}$, $M(q) = diag(M_1, M_2, \ldots, M_i, \ldots, M_N) \in R^{n\times n}$, $h(q,\dot{q}) = (h_1, h_2, \ldots, h_i, \ldots, h_N)^T \in R^{n\times 1}$, $\tau = (\tau_1, \tau_2, \ldots, \tau_i, \ldots, \tau_N) \in R^{n\times 1}$, and $f = (f_1, f_2, \ldots, f_i, \ldots, f_N)^T \in R^{n\times 1}$, then (1) can be written in a compact form as

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

Remark: The dynamics model (2) is quite general in the sense that it does not specify how the assumption (ii) stated above (i.e., that the mass and inertia of the object can be lumped into the last links of the manipulators) is to be met. This concerns the issue of load distribution in multi-manipulator systems, an important problem which has been investigated extensively in the robotics literature (e.g., [18, 22]). For the purpose of studying the utility of neural networks in multi-manipulator system control, it suffices to recognize that a dynamics model of the system (with a given load distribution) is available for developing control laws augmented by the neural network to improve the performance of the overall system.

Since the manipulators are in rigid contact with the object, the motion of the manipulators is constrained. The constraint may be expressed as: $\varphi(q) = 0$, where $\varphi(\cdot) \in R^m$. It is assumed that $\varphi(\cdot)$ is continuous and twice differentiable. (Note that, through the constraint φ, the 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 $\lambda \in R^m$ is a vector of Lagrange multipliers associated with the constraint $\varphi = 0$.) To reflect the fact the n joint variables of the robots are related through the m constraints, the vector q is partitioned into two subvectors $\bar{q}_1$ and $\bar{q}_2$ i.e., $q = \left(\bar{q}_1^T, \bar{q}_2^T\right)$, where $\bar{q}_1 \in R^m$, and $\bar{q}_2 \in R^{n-m}$. With the assumption that a function Ω exists such that the constraint can be expressed as: $\varphi(\Omega(\bar{q}_2), \bar{q}_2) = 0$, the following nonlinear coordinate transformation as described in [13] is introduced. Let $x = [\bar{q}_1 - \Omega(\bar{q}_2), \bar{q}_2]^T$. Then $q = [x_1 + \Omega(x_2), x_2]^T \overset{\Delta}{=} Q(x)$. Since $\dot{q} = T\dot{x}$ and $\ddot{q} = T\ddot{x} + \dot{T}\dot{x}$, where $T \overset{\Delta}{=} \frac{\partial Q(x)}{\partial x}$. Equation (2) becomes, with arguments suppressed,

$$\bar{M}\ddot{x} + \bar{h} = T^T\tau - T^T f, \tag{3}$$

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 $x_1 = 0$. Thus (3) can be expressed in "reduced" form [13] as

$$E_1\overline{M}\,E_2^T\ddot{x}_2 + E_1\overline{h} = E_1T^T\tau - E_1T^T f, \tag{4}$$

$$E_2\overline{M}\,E_2^T\ddot{x}_2 + E_2\overline{h} = E_2T^T\tau. \tag{5}$$

where $E_1 \in \mathcal{R}^{m\times n}$ and $E_2 \in \mathcal{R}^{(n\text{-}m)\times n}$ are obtained by partitioning the identity matrix $I_n \in R^{n\times n}$ as $I_n = [\,E_1^T, E_2^T\,]$, which also results in $E_2T^Tf = 0$.

Remark: As stated, the derivation of the system dynamics in reduced form (4)-(5) follows the methodology presented in [13]. Other similar method of obtaining reduced order dynamics models of multi-manipulator systems have also been reported (e.g., [11, 23]. The utilization of the particular formulation in this work is mainly due to the observation that, 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, thus leading to a simplified framework to facilitate the development of the uncertainty compensation scheme using a neural network. This formulation also leads to the conclusion that it is not necessary to use force measurement as part of the input to the neural network for uncertainty compensation, and as a consequence, the dimension of the input layer of the neural network is reduced.

3. CONTROL

In practical robotic applications, the parameters $\overline{M}$ and $\overline{h}$ are not known exactly. In the proposed control law presented below, "nominal" (or estimated) values of these parameters, denoted by $\widetilde{M}$ and $\widetilde{h}$ respectively, are used to construct a control law based on the dynamics model. The effect of the discrepancy between the nominal and actual parameter values is to be compensated by some additional signal. The proposed control law for the system (3) is a modified computed-torque control as in [13], plus an additional compensating signal v, and is specified as

$$T^T\tau = \widetilde{M}\left(\ddot{x}^d + E_2^TK_vE_2\left(\dot{x}^d - \dot{x}\right) + E_2^TK_pE_2\left(x^d - x\right)\right)\widetilde{M}v + \widetilde{h} + E_1^TK_fE_1T^T\left(f^d - f\right) + T^Tf^d, \tag{6}$$

where $K_v \in \mathcal{R}^{(n\text{-}m)\times(n\text{-}m)}$, $K_p \in \mathcal{R}^{(n\text{-}m)\times(n\text{-}m)}$, 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 trajectories, f^d and f are respectively the desired force and the actual force.

With the control law (6), the closed-loop system dynamics can be expressed as

$$\dot{e} = Ae + B(\Delta V - \Delta f). \tag{7}$$

where $e = \begin{bmatrix} e_x \\ \dot{e}_x \end{bmatrix}$, $e_x = x^d - x$, $A = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ I \end{bmatrix}$, $\Delta v = \eta - v$, $\overline{K}_v = E_2^TK_vE_2$, $\overline{K}_p = E_2^TK_pE_2$, $\eta = \left(\widetilde{M}^{-1}\overline{M} - I\right)\ddot{x} + \widetilde{M}^{-1}\Delta\overline{h}$, $\Delta\overline{h} = \overline{h} - \widetilde{h}$, $\Delta_f = \widetilde{M}^{-1}\ K\Delta_f\,(f^d - f)$, $K\Delta_f = \overline{K}_f + T^T$, and $\overline{K}_f = E_1^TK_fE_1T^T$. Note that $K\Delta_f$ is nonsingular due to the definition of T and Q. Figure 1 schematically depicts the closed-loop system.

It remains to show that if the compensator is designed such that $\Delta_v \rightarrow 0$, then the asymptotic trajectory tracking of the manipulators can be achieved while the actual force tracks the desired force. Recall that $\eta = \left(\widetilde{M}\,\overline{M} - I\right)\ddot{x} + \widetilde{M}^{-1}\Delta\overline{h}$. Expressing $\Delta_v \equiv \eta - v = 0$ in reduced form yields

$$E_1\left(\overline{M} - \widetilde{M}\right)E_2^T\ddot{x}_2 + E_1\Delta\overline{h} - E_1\widetilde{M}v = 0, \tag{8}$$

$$E_2\left(\overline{M} - \widetilde{M}\right)E_2^T\ddot{x}_2 + E_2\Delta\overline{h} - E_2\widetilde{M}v = 0, \tag{9}$$

Since $E_2T^Tf = 0$, $E_1E_1^T = I_m$, and $E_2E_1^T = 0$, the proposed control law can also be expressed in reduced form as

$$E_1T^T\tau = E_1\widetilde{M}E_2^T\left(\ddot{x}_2^d + K_v\dot{e}_2 + K_pe_2\right) + E_1\widetilde{M}v + E_1\widetilde{h} + K_fE_1T^T\left(f^d - f\right) + E_1T^Tf^d, \tag{10}$$

$$E_2T^T\tau = E_2\widetilde{M}E_2^T\left(\ddot{x}_2^d + K_v\dot{e}_2 + K_pe_2\right) + E_2\widetilde{M}v + E_2\widetilde{h}, \tag{11}$$

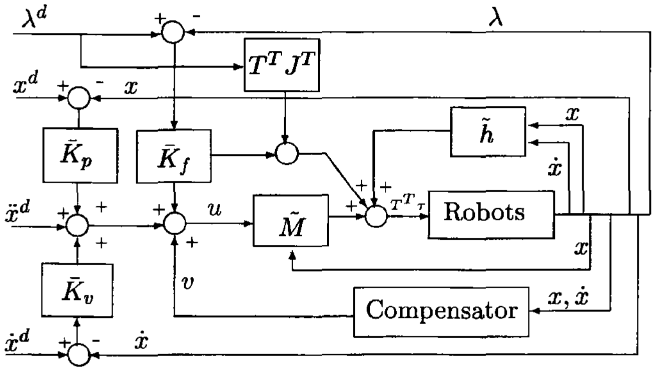

Figure 1. Multi-Manipulator System Control.

where $e_2 = x_2^d - x_2$. Substituting (10-11) into (4-5) yields the closed-loop system equations in reduced form

$$E_1\tilde{M}E_2^T\left(\ddot{e}_2 + K_v\dot{e}_2 + K_pe_2\right) = E_1\left(\overline{M} - \tilde{M}\right)E_2^T\ddot{x}_2 + E_1\Delta\bar{h} - E_1\tilde{M}v + K_{\Delta f}\left(f^d - f\right), \tag{12}$$

$$E_2\tilde{M}E_2^T\left(\ddot{e}_2 + K_v\dot{e}_2 + K_pe_2\right) = E_2\left(\overline{M} - \tilde{M}\right)E_2^T\ddot{x}_2 + E_2\Delta\bar{h} - E_2\tilde{M}v. \tag{13}$$

Equation (13) characterizes the motion of the manipulators under the constraint $\varphi = 0$, while equation (12) expresses the forces in terms of the dynamics of the constrained motion. The overall dynamic equation is decomposed to arrive at this reduced form because, put in such a form, the "motion aspect" and the "force aspect" of the manipulators can be readily examined. Now substituting (8-9) into (12-13), yields

$$E_1\tilde{M}E_2^T\left(\ddot{e}_2 + K_v\dot{e}_2 + K_pe_2\right) = K_{\Delta f}\left(f^d - f\right) \tag{14}$$

$$E_2\tilde{M}E_2^T\left(\ddot{e}_2 + K_v\dot{e}_2 + K_pe_2\right) = 0. \tag{15}$$

It can be seen that equation (15) characterizes the desired asymptotic motion of the manipulators under the constraint φ=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 (14), it follows that $f \to f^d$.

It has now been shown that if the compensator is designed such that $\Delta_v \to 0$, then asymptotic tracking of the desired motion of the manipulators, and consequently the desired forces can be achieved. Hence Δ_v is referred to as the *control error*. The objective of uncertainty compensation is to reduce this control error.

4. UNCERTAINTY COMPENSATION

Note that $\Delta_v = 0$ implies that $v = \eta(x, \dot{x}, \ddot{x})$. Although the structure of the function $\eta(\cdot)$ is known, the exact values of the parameters of this function are not known explicitly. Consequently, it is not possible to predict the exact output of this function. (otherwise the problem of designing N is trivial). 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, uncertainty compensation can 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 (as illustrated in Figure 2), the network output is generated [9] according to vi = g $\left(\sum_{j=1}^{J_n} W_{ijg}\left(\sum_{k=1}^{K_n} R_{jkg}\left(\sum_{l=1}^{L_n} S_{klzl}\right)\right)\right)$, where g(·) = ctanh(·), and c is a scaling factor. For convenience a generalized weight vector Θ is defined as: Θ = [W1, ..., WIn, R1, ..., RJn, S1, ..., SKn] ∈ Rcθ, where (·)i represents the ith row of the matrix (·), and cθ = In × Jn + Jn × Kn + Kn × Ln. Thus the mapping realized by the network can be compactly expressed as: v = g(Z,Θ), where Z is the input vector, i.e., Z = [z1, z2, ..., zl, ..., zLn]T.

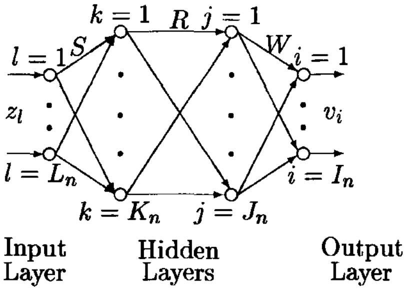

Figure 2. Neural Network Structure.

To approximate the function η(x, $\dot{x}$, $\ddot{x}$), the neural network takes x, $\dot{x}$, and $\ddot{x}$ as its input, and produces an output v. The objective of neural network learning is to adjust the weights of the neural network to minimize the control error Δ_v. The error-backpropagation algorithm [9] is an effective algorithm for neural network learning.

Let the cost function to be minimized be: $J_{\Delta v} = \frac{1}{2}\Delta v^T \Delta v$. Proper application of the error-backpropagation algorithm yields the weight update rule: $\dot{\Theta} = -\lambda_n \Delta v^T \frac{\partial \Delta v}{\partial \Theta}$, where λ_n is the learning rate. Since the objective of neural network learning is to produce a network output v such that v approaches η, at a given instant η can be considered as an "desired" output of the network; consequently, $\frac{\partial \Delta v}{\partial \Theta} = \frac{\partial(n-v)}{\partial \Delta} = -\frac{\partial v}{\partial \Theta}$. Thus the update rule can be implemented as: $\dot{\Theta} = -\lambda_n \Delta v^T \frac{\partial v}{\partial \Theta}$.

This learning rule requires the control error Δv, which can be computed from (7) as: $\Delta v = \ddot{e}_x + \overline{K}_v \dot{e}_x + \overline{K}_p e_x + \Delta f$. This expression for Δ$v$ implicitly contains the joint acceleration $\ddot{q}$, which can be estimated based on $\dot{q}$ using appropriate filtering techniques a Kalman filter.

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

$$\begin{cases} \dot{e} = Ae + B(\Delta v(x,\dot{x},\ddot{x},\Theta) - \Delta f) \\ \dot{\Theta} = -\lambda_n \Delta v^T(x,\dot{x},\ddot{x},\Theta)\frac{\partial \Delta v(x,\dot{x},\ddot{x},\Theta)}{\partial \Theta} \end{cases} \tag{16}$$

5. ANALYSIS

In this section, the dynamical behavior of the closed-loop system (16) is analyzed. First, the closed-loop system with the neural network learning on-line is proved to be stable in the sense that all the signals in the system are bounded. It is then proved that the performance of the closed-loop system is improved in the sense that the $\mathcal{L}_2$-norm of the control error Δ_v decreases as the learning process is iterated (i.e., as the number of trials increases). The subsequent conjecture is that reduction in the control error eventually leads to reduction in the force and position errors. These conclusions are verified by the results of the numerical simulation presented in Section 6.

5.1 Stability

Since the learning process is concurrent with the execution of the robotic task, it is important to ensure that the robotic system be stable so that neural network learning can be conducted effectively. (The dynamical effect of neural network learning on the stability of the robot is discussed later in Proposition 1.) For this purpose, it suffices to require that all signals in the closed-loop system are bounded, that is, the system is bounded-input bounded-output stable. (A system is said to be bounded-input bounded-output (BIBO) stable if for a bounded input, the output of the system is also bounded. This term is rigorously defined in Appendix A.) The following theorem (Theorem 1) and corollaries (Corollaries 1-3) assert that in (16) such a requirement is met.

Theorem 1 *Given a continuous and twice-differentiable reference position trajectory $x^d(t)$ and a continuous reference force trajectory $f^d(t)$, the system (16) is BIBO stable for sufficiently large gains K_v, K_p, and K_f.*

Corollary 1 *The acceleration signal $\ddot{x}(t)$ is bounded.*

Corollary 2 *The control error Δ_v is bounded.*

Corollary 3 *The weights of the neural network remain bounded during a given trial.*

Proofs of Theorem 1 and Corollaries 1-3 are presented in Appendix A through Appendix D. With Theorem 1 and Corollaries 1-3, it is rigorously established that the closed-loop system with the neural network learning on-line is stable in the sense that all signals are bounded. However, these results do not specify the evolution of the behavior of the system during the neural network learning process (e.g., the difference in the system behavior between two trials). The effect of the network learning process on the performance of the multi-manipulator system is examined next.

5.2 Performance Improvement

In this section, it is proved that the performance of the closed-loop system is improved in the sense that the $\mathcal{L}_2$-norm of the control error Δv decreases as the learning process is iterated (i.e., as the number of trials increases).

Preliminaries

Let t represent the continuous time variable, i.e., $0 \le t < \infty$. Let learning start at time $t = 0$, and let each trial last T seconds. Then the p^{th} trial spans the time period from $t = (p-1)T$ to $t = pT$; the second trial spans the time period from $t = T$ to $t = 2T$; and so the p^{th} trial spans the time period from $t=(p-1)T$ to $t=pT$. Note that p is thus implicitly defined as a positive integer. Let ξ be the time variable associated with one trial, i.e., $0 \le \xi \le T$. (The notation $\dot{x}$ from here on means either $\frac{dx}{dt}$ or $\frac{dx}{d\xi}$ as it should be clear from the context.) Let $x(p, \xi)$ denote the value of the variable x at the ξ^{th} second of the p^{th} trial. Then $x(p,0)$ represents the value of the variable x at the beginning of the p^{th} trial, and $x(p,T)$ represents the value of the variable x at the end of the p^{th} trial. In other words, $x(p,0)$ and $x(p,T)$ represent respectively the initial value and the final value of the variable x for the trial p. Note that $x(p,0) = x(p-1,T)$. Let Θ^* denote a set of optimal weights, i.e., $\left.\frac{\partial v}{\partial \Theta}\right|_{\Theta^*} = 0$. Let Θ_m be an element of the weight vector Θ, and let $\Delta\theta_m(p,\xi)$ denote the change of Θ_m during the first ξ seconds of the p^{th} trial, i.e., $\Delta\theta_m(p,\xi) = \int_0^\xi \dot{\Theta}_m(p,\sigma)d\sigma$.

The $\mathcal{L}_\infty^n$ – norm of a Lebesque integrable function $f(t) : R_+ \to R_n$, denoted by $\|f\|_\infty$, is defined as: $\|f\|_\infty =$ $ess\ sup_{t\in[0,\infty)} \|f(t)\| < \infty$. The extended $\mathcal{L}_\infty^n$ – space, (for the truncated $\mathcal{L}_\infty^n$ – norm), denoted by $\mathcal{L}_{\infty e}^n$, is defined as: $\mathcal{L}_{\infty e}^n = \{f : R_+ \to R^n \mid f_T \in \mathcal{L}_\infty^n, \forall T < \infty\}$, where $f_T(t) = f(t)$ for $t \in [0,T]$, and $f_T(t) = 0$ for $t \notin [0,T]$. For convenience, the notation $\|f\|_{T\infty}$ is used to denote $\|f_T\|_\infty$.

Let $C[0,T]$ denote the family of Lebesgue integrable function $f_i(\xi)$ for all $\xi \in [0,T]$. The $\mathcal{L}_2$ – norm of a vector function $f(\xi) = (f_1, f_2, \ldots, f_i, \ldots, f_n)$, $f_i \in C[0,T]$, over the time interval $[0,T]$ is defined [21] as $\|f\|_{2T} = \left(\int_0^T f^T(\xi)f(\xi)d\xi\right)^{\frac{1}{2}}$. The subscript T in the notation $\|\cdot\|_{2T}$ indicates that the norm is defined in the extended L_p space, since we are only interested in the size of the various signals of the robotic system over the closed time interval $[0,T]$. For convenience, the notation $\|\cdot\|$ instead of $\|\cdot\|_{2T}$ is used to denote this norm in the sequel.

Proposition 1 *For the closed-loop system (16), there exists some small $\lambda_n > 0$ such that $\| \Delta\theta_m(p+1, \xi) - \Delta\theta_m(p,\xi) \|T\infty << 1$, for all $\xi \in [0,T]$.*

A proof of this proposition is presented in Appendix E.

Remark: Since $\Delta\theta_m(p+1,\xi)$ represents the amount of weight change during the first ξ seconds of the $(p+1)^{th}$ trial, and $\Delta\theta_m(p,\xi)$ represents the amount of weight change during the first ξ seconds of the p^{th} trial, this proposition (illustrated in Figure 3) means that, for $\lambda_n <<1$, although the difference between the initial and final values of Θ_m, i.e.

$\Theta_m(p+1,0)$ - $\Theta_m(p,0)$, of any trial could be significant, the difference between the change in Θ_m of any two successive trials can be considered to be negligible, i.e., $\Delta\Theta_m(p+1,\xi) \approx \Delta\Theta_m(p,\xi)$, $\forall\ \xi \in [0,T]$.

A qualitative interpretation for this proposition can be constructed based on the observation on the time-scale difference between the dynamics of the manipulator and the dynamics of the neural network with the neural network learning on-line. With a small learning rate λ_n, the overall system can be considered as a two time-scale system with the manipulators exhibiting a "fast" dynamics while the network exhibiting a "slow" dynamics. As the learning rate λ_n approaches zero, the change of the state of the network (i.e., the change in the values of the connection weights) becomes infinitesimally small. One consequence is that, with $\lambda_n << 1$, the change in the weights *per trial* can be expected to be small. Such small change in the weights will not have significant effect on the state of the robots. Therefore, between any two *successive* trials, the change in the state of the robots, and consequently can be considered negligible. Since the rate of weight change depends on the state of the robot, it is reasonable to consider that the difference between the two amounts of weight change, can be considered negligible. This proposition is further verified by computer simulation presented in Section 6.

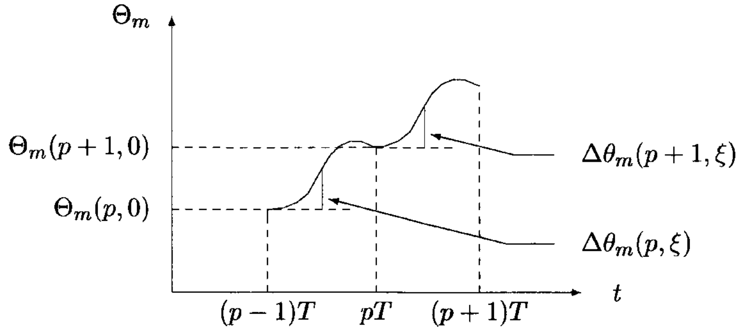

Figure 3. Weight and Weight Change.

Main Results

Let μ_p denote the $\mathcal{L}_2$-norm of the control error Δ_v of the p^{th} trial, i.e., $\mu_p = \| \Delta v(p)\|$. For a series of trials, a sequence $\langle\mu_p\rangle = \{\mu_1, \mu_2, \ldots\}$ is formed. Suppose that the learning process converges after k trials, then a finite sequence $\langle\mu_p\rangle = \{\mu_1, \mu_2, \ldots, \mu_k\}$ is obtained, with $\mu_k = \min_{p\in[1,k]} \langle\mu_p\rangle$. If the learning algorithm converges globally, then μ_k can be made arbitrarily small by using a sufficient number of hidden units [5]. In the limiting case, $\mu_k \to 0$ as $K+J \to \infty$, where K and J are the number of units in the first and second hidden layers respectively. (Note that in practice K and J are finite. Hence, however small μ_k may be, it will not reach [3].) If convergence is achieved only locally, then $\mu_k > 0$. In any case, it is established that convergence of the learning process guarantees the existence of a minimum value of the $\mathcal{L}_2$-norm of the control error.

Theorem 2 *For the closed-loop system (16) with a learning rate satisfying Proposition 1 and a set of initial weights $\Theta \neq \Theta^*$, the sequence $\langle\mu_p\rangle$ is finite, strictly monotonically decreasing, and converges to some $\mu_k \geq 0$.*

The proof of this theorem is presented in Appendix F. This theorem asserts that, given the system dynamics as specified in (16), the $\mathcal{L}_2$-norm of the control error Δ_v converges monotonically to some minimum in finite number of trials. The subsequent conjecture is that reduction in the control error Δ_v eventually leads to reduction in the force and position errors. This is verified by the results of the simulation described in Section 6.

Remark: An important issue concerning neural network learning is whether the weights will remain bounded. This issue can be resolved (in the context of the above analysis) by observing the facts that (i) the weights are bounded during a given trial (i.e., Corollary 3), and (ii) from an implementation standpoint, the learning process can be terminated once the $\mathcal{L}_2$-norm of the control error no longer decreases from trial to trial. Thus, if the condition specified by Proposition 1 is satisfied, then it is assured that the weight values are finite at the point when the learning process is terminated.

6. SIMULATION

One of the purposes of conducting computer simulations was to confirm, through a numerical example of multi-manipulator coordination, that (i) for a sufficiently small learning rate λ_n, the weight change between two successive trials can be considered negligible (Proposition 1), (ii) upon confirmation of (i), the $\mathcal{L}_2$-norm of the control error Δ_v decreases as the number of learning trial p increases (Theorem 2), and (iii) reduction in the control error Δ_v results in reduction in both the force tracking error and the position tracking error.

The multi-manipulator system used in the simulations consists of two planar robots as depicted schematically in Figure 4. One of the manipulators (on the left) has three joints while the other has two. All links (except the last links) of the manipulators are of the length of 1 *m*; the last link of each robot has the length of 0.5 *m*. The distance between the robot bases is 1 *m*. The tip of the robots are assumed to be in rigid contact. Thus the last links of the robots form one "common" rigid link. The entire system can be alternatively considered as consisting of two robots handling a common object represented by the "common link".

The constraints on the manipulators are: $\varphi_1 = \cos q_{11} + \cos(q_{11} + q_{12}) + \frac{1}{2}\cos(q_{11} + q_{12} + q_{13})$ -1-cos q_{21} - $\frac{1}{2}\cos(q_{21} + q_{22})$, $\varphi_2 = \sin q_{11} + \sin(q_{11} + q_{12}) + \frac{1}{2}\sin(q_{11} + q_{12} + q_{13}) - \sin q_{21} - \frac{1}{2}\sin(q_{21} + q_{22})$, and $\varphi_3 = q_{11} + q_{12} + q_{13} - q_{21} - q_{22} + \pi$. The dynamics of this two-manipulator system can be expressed as

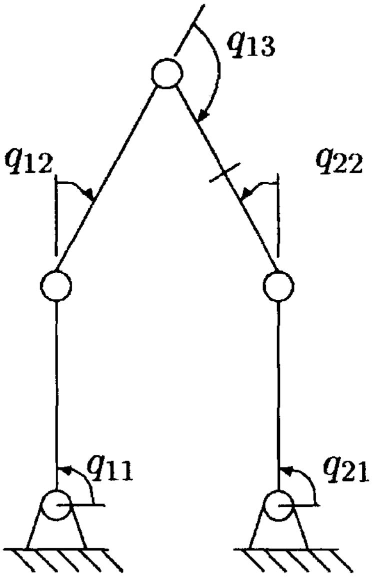

Figure 4. Two Cooperative Planar Manipulators.

$$\mathrm{M}\ddot{q} + \mathrm{h} = \tau - \mathrm{JT}\,\lambda, \tag{17}$$

where $q = (q_1, q_2)^T$, $q_1 = (q_{11}, q_{12}, q_{13})^T$, $q_2 = (q_{21}, q_{22})^T$, $M = diag(M_1, M_2)$, $h = (C_1 \dot{q}_1, C_2\ \dot{q}_2)^T$, $\tau = (\tau_1, \tau_2)^T$, $J = \frac{\partial\varphi}{\partial q}$, and $\varphi = (\varphi_1, \varphi_2, \varphi_3)^T$. The elements of the matrices M_1, M_2, C_1, and C_2 are given in Appendix G. The partition of q is as follows: $\bar{q}_1 = (q_{11}, q_{12}, q_{13})^T$, and $\bar{q}_2 = (q_{21}, q_{22})^T$. The nonlinear transformation **(described in Section 2.4)** needed for generating the control signals were determined using the *Maple* software. The source code is included in the Appendix.

In simulating the dynamics of this two-manipulator system, the parameter values as listed in Table I under the heading "True" were used. To introduce uncertainty into the system, the estimated values of the parameters, also listed in Table I, were used in the control law.

A neural network with six input units, twenty units in each of its two hidden layers, and five output units was used in the simulations. It is noted that currently there exists no analytical methodology for determining the "optimal" size of a network for a given problem. Empirical evidence has suggested that the "optimality" of network architecture and size depends on various factors (such as the dynamics and the system complexity) associated with the specific application. In the absence of analytical tools, the "optimality" of a neural network can be estimated via empirical means. However, since the purpose of this simulation is to verify the analyses presented in Section 5 (as opposed to the actual development of the neural network as a prototype device), it suffices to use a suitable network size, with consideration of available computational resources and complexity of implemetation.

Table I. True and Estimated Parameter Values.

Parameter	Three-Link Robot		Two-Link Robot	
	True	*Estimated*	*True*	*Estimated*
c_1	0.313	0.25	--	--
c_2	0.625	0.40	--	--
c_3	0.625	.05	--	--
c_4	4.168	1.95	0.313	0.10
c_5	5.00	8.30	0.625	0.40
c_6	9.168	11.00	4.168	5.50

The input to the neural network were q_{21}, q_{22}, $\dot{q}_{21}$, $\dot{q}_{22}$, $\ddot{q}_{21}$, and $\ddot{q}_{22}$. The acceleration signals $\ddot{q}_{21}$ and $\ddot{q}_{22}$ were obtained by filtering the velocity signals $\dot{q}_{21}$ and $\dot{q}_{22}$ using a Kalman filter [24]. The learning rate of the neural network was set at a small value of 5×10^{-5}, so as to be consistent with the argument for Proposition 1 discussed in Section 5. The initial weights of the network were set randomly in the order of 10^{-5}. The scaling factor c of the activation function was set at unity for the hidden units and at 30, 25, 5, 30 and 25 for the five output units. (The values of the scaling factors for the output units were selected to be of the same order of magnitude as the uncertainty η when no compensator was used.) The control gains were as follows: K_p = *diag*(50,50), K_v = *diag*(10,20), and K_f= *diag*(10,10,10). 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 was set to be 5.0, and $\delta = y_f - y_0$, with y_0 and y_f being respectively the initial and final value of y. The other two forces, λ_2 and λ_3, were to be regulated to zero. The initial configuration of the multi-manipulator system was set as: q_{11} = 90°, q_{12} = -30°, q_{13} = -120°, q_{21} = 90°, q_{22} = 30°, λ_1 = 0 *N*, λ_2 = 0 *N*, and λ_3 = 0 *Nm*. The final state of the system was specified to be: q_{21} = 70°, q_{22} = 60°, λ_1 = 5 *N*, λ_2 = 0 *N*, and λ_3 = 0 *Nm*.

The desired task trajectory of the tips of the manipulators is as shown in Figure 5. The desired trajectory for the force λ_1 is as shown in Figures 6.

To solve the set of differential-algebraic equations in (17), the second-order back-difference formula [4] was used in the numerical integration algorithm, with the step size fixed at 0.005*s*. A series of trials was 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.

To examine the dynamical behavior of the weights between two successive trials, Figure 7 shows the dynamics of a connection weight $R_{(5,5)}$ (the weight between the 5th unit of the first hidden layer and the 5th unit of the second hidden layer) during the 5th trial and the 6th trial, Figure 7 shows and the difference between the change of $R_{(5,5)}$ in these two trials.[1]

Figure 8 shows the dynamics and the difference between the change of the same weight during the 10th trial and the 11th trial.

It can be seen that the difference between the weight change of two successive trials (i.e., 5th and 6th, 10th and 11th) is indeed negligible.

Figures 9 and 10 show the dynamics and the difference between the change of another weight $W_{(2,5)}$ (the weight between the 5th unit in the second hidden layer and the 2nd unit of the output layer) during the same pair of trials (i.e., 5th and 6th, 10th and 11th).

Comparing Figures 7 – 10 with Figure 3 (which illustrates the proposition that the difference between the weight change of any two successive trials is negligible), it is evident that, when the learning rate is small, the neural network does indeed possess the dynamical behavior as predicted.

[1] It is shown here, as an example, how the difference between change in $R_{(5,5)}$ of the two trials, 5th and 6th, is calculated. Using the notations defined in Section 5, the difference between the weight change of trial 5 and trial 6 can be written as: $\Delta r_{(5,5)}(\xi) = \Delta R_{(5,5)}(6, \xi) - \Delta R_{(5,5)}(5, \xi)$, where $\Delta R_{(5,5)}(6, \xi) = R_{(5,5)}(6, \xi) - R_{(5,5)}(6,0)$, $\Delta R_{(5,5)}(5, \xi) = R_{(5,5)}(5, \xi) - R_{(5,5)}(5,0)$, and ξ is the time variable, i.e., ξ [0, 2].

Recall that the analytical conclusion (following the proposition about the weight dynamics between successive trials) is that the $\mathcal{L}_2$-norm of the control error Δ_v decreases as the number of trials p increases. Figure 11 shows the $\mathcal{L}_2$-norm of the control error Δ_v versus the trial number p for the simulations.

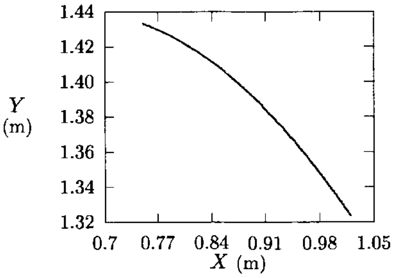

Figure 5. Desired Task Trajectory.

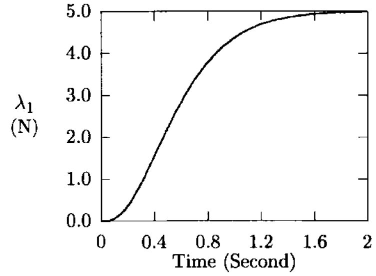

Figure 6. Desired Trajectory of λ_1.

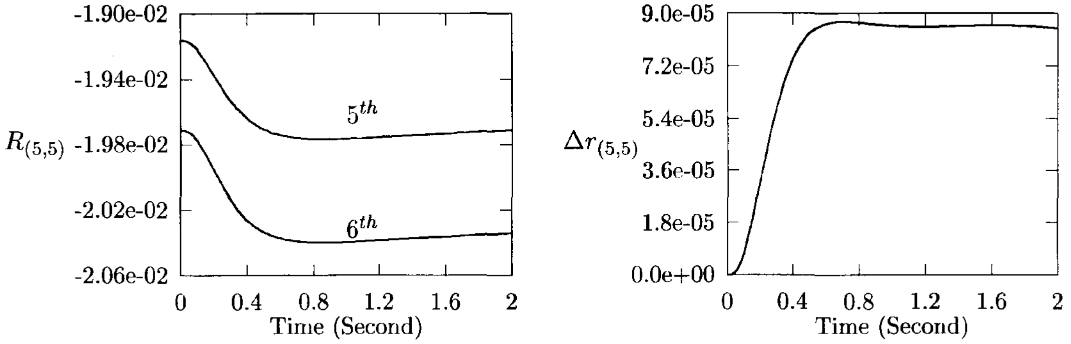

Figure 7. Dynamics and Difference between Change of Weight $R_{(5,5)}$: 5th and 6th Trial.

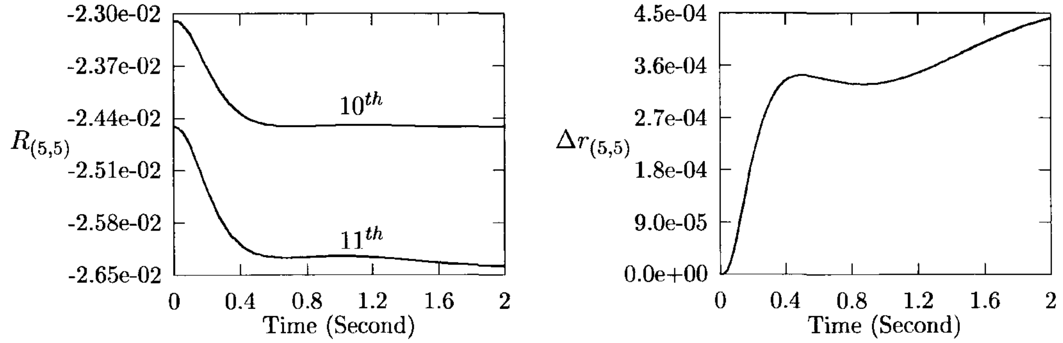

Figure 8. Dynamics and Difference between Change of Weight $R_{(5,5)}$: 10th and 11th Trial.

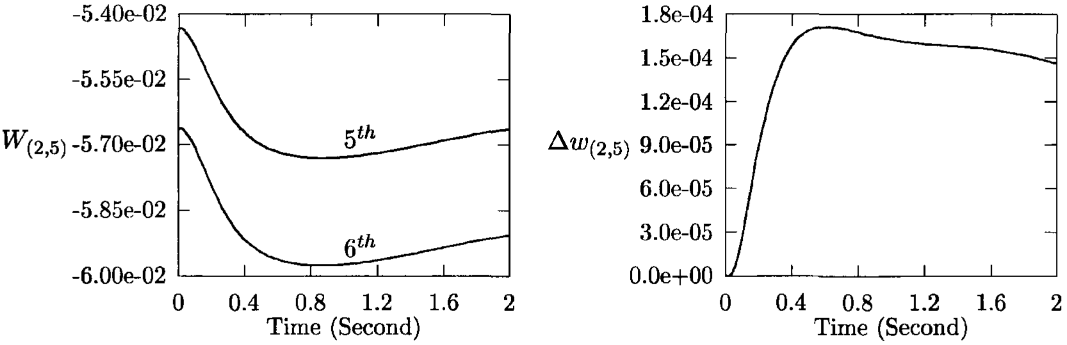

Figure 9. Dynamics and Difference between Change of Weight $W_{(2,5)}$: 5th and 6th Trial.

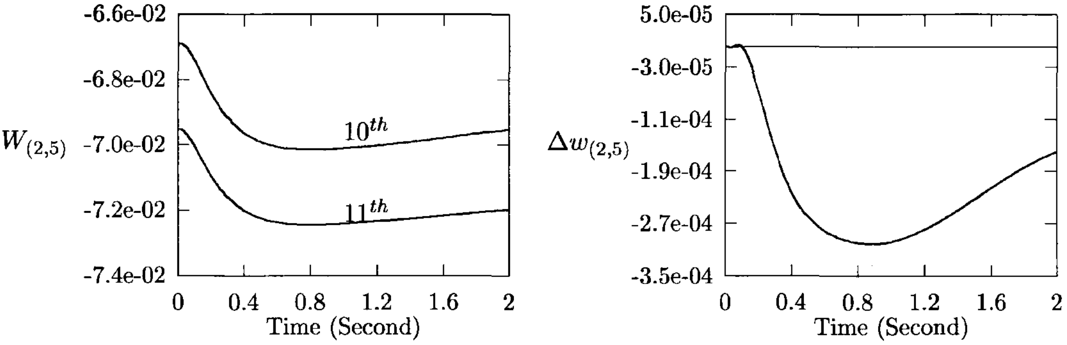

Figure 10. Dynamics and Difference between Change of Weight $W_{(2,5)}$: 10th and 11th Trial.

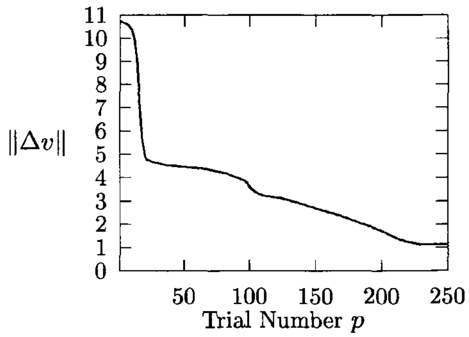

Figure 11. Control Error.

It can be seen that the control error Δ_v indeed decreases as the number of trials p increases. This confirms the theoretical conclusion presented in Section 5.

The subsequent conjecture is that reduction in the control error Δ_v eventually results in reduction in the position error and the force error in the multi-manipulator system. Figure 12 shows the task trajectory of the tips of the manipulators during the initial trial (i.e., without compensation) and during the 250th trial. Figures 13, 14 and 15 show, respectively, the trajectories λ_1, λ_2, and λ_3 during the initial trial and the 250th trial. Note that λ_2 and λ_3 were to be regulated to zero.

It can be seen that both the force error and the position error are reduced as the learning of the network progresses.

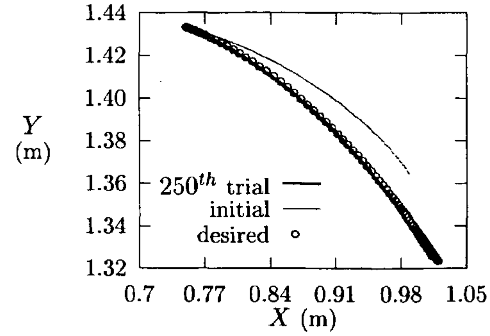

Figure 12. Position Trajectory in Task Coordinates.

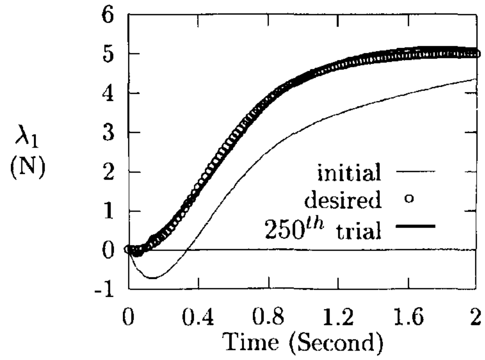

Figure 13. Trajectory of λ_1.

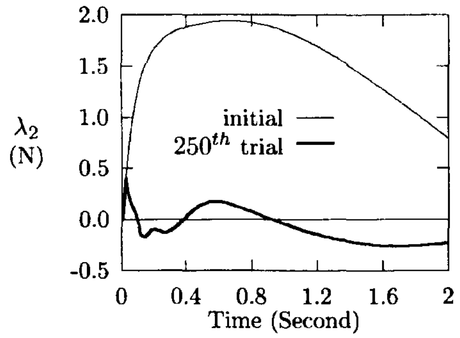

Figure 14. Trajectory of λ_2.

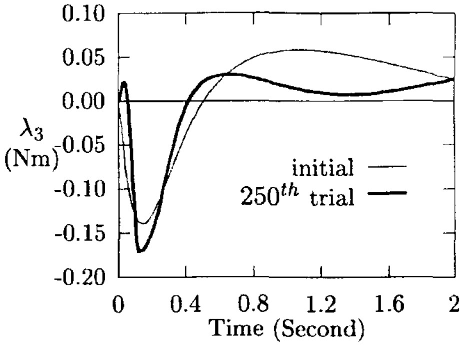

Figure 15. Trajectory of λ_3.

From the results of the simulations, it can be seen that both the proposition (that with a small learning rate, the weight change between two successive trials can be considered negligible) and the conclusion (that the $\mathcal{L}_2$-norm of the control error decreases as the number of trials p increases) are valid. It can also be seen that reduction in the control error Δ_v indeed results in reduction in both the position error and the force error.

7. SUMMARY AND IMPLICATIONS OF RESULTS

An approach to uncertainty compensation using a neural network in the control of multi-manipulator systems has been presented. This proposed approach is formulated in the framework of constrained motion. It has been shown that, by such a formulation, the resulting dynamic equations of motion can be 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. It also leads to the conclusion that it is not necessary to use the force information as part of the input to the neural network for uncertainty compensation. It has also been shown that, in this proposed approach, uncertainty in the multi-manipulator system can be quantified and a neural network can be used to compensate for the effect of the uncertainty so that performance improvement can be achieved.

Analysis on the closed-loop stability of the multi-manipulator system (incorporated with a neural network) using techniques from nonlinear system theory [21] has been presented. Results of the analysis assert that the closed-loop system is stable in the sense that all signals in the system are bounded. Stability of closed-loop systems embedded with neural networks has been a key issue that has not been adequately addressed in the literature. The analyses presented in this paper offer a possible method in dealing with this issue. In addition, analytical results concerning the performance of the multi-manipulator system with the neural network learning on-line have been obtained. This analysis reveals the effect of the dynamics of the neural network on the performance of the robotic system. It has further been shown that the performance of the multi-manipulator system is improved as the learning process of the neural network is iterated. Numerical simulations have been described and results have been presented. The results of the simulations have confirmed the conclusions of the analyses.

The analytical and empirical results reported in this article suggest that neural networks could be used as "add-on" control modules to improve the performance of industrial robots in execution of tasks involving two or more cooperative manipulators.

ACKNOWLEDGEMENT

This work is financially supported by the Institute for Robotics and Intelligent Systems, the Ontario Graduate Scholarship, and the Natural Sciences and Engineering Council of Canada. The authors 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.

REFERENCES

[1] S. Arimoto, 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*, 1987, pp 1407-1412.

[2] B. Armstrong-H'elouvry. *Control of Machines with Friction.* Boston, Massachusetts: Kluwer Academic Publishers, 1991.

[3] A. R. Barron. Universal Approximation Bounds for Superposition of a Sigmoidal Function. *IEEE Transactions on Information Theory.* Vol. 39, No. 3, 1993, pp 930-945.

[4] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* Elsevier Science, 1989.

[5] G. Cybenko. Approximation by Superposition of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, Vol. 2, 1989, pp 303-314.

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

[7] T. Fukuda 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, 1990, pp 310-315.

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

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

[10] Y. R. Hu 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*, Scottsdale, Arizona, 1989, pp 1091-1096.

[11] A. J. Kiovo and M. A. Unseren. Reduced Order Model ad Decoupled Control Architecture for Two Manipulators Holding a Rigid Object. *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 113, December 1991, pp 646-654.

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

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

[14] C. Mead. *VLSI and Neural Systems*. Reading, Massachusetts: Addison-Wesley, 1989.

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

[16] M. Nahon and J. Angeles. Real-time Force Optimization in Parallel Kinematic Chains under Inequality Constraints. *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, 1991, pp 2198-2203.

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

[18] D. E. Orin and S. Y. Oh. Control of Force Distribution in Robotic Mechanisms Containing Closed Kinematic Chains. *ASME Journal of Dynamic Systems, Measurement, and Control* Vol. 102, June 1981, pp 134-141.

[19] M. W. Spong and M. Vidyasagar. Robust Linear Compensator Design for Nonlinear Robotic Control. *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, August 1987, pp 345-351.

[20] Jian M. Tao 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*, 1993, pp 144-148.

[21] M. Vidyasagar. *Nonlinear Systems Analysis*. 2nd ed., New Jersey: Prentice Hall, 1992.

[22] I. D. Walker, R. A. Freeman, and S. I. Marcus. Analysis of Motion and Internal Loading of Objects Grasped by Multiple Cooperating Manipulators. *International Journal of Robotics Research*, Vol. 10, No. 4, August 1991, pp 396-409.

[23] L. T. Wang and M. J. Kuo. Dynamic Load-Carrying Capacity and Inverse Dynamics of Multiple Cooperating Robotic Manipulators. *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 1, February 1994, pp 71-77.

[24] G. Yanovski. *Experimental Analysis of a Robust Centralized/Decentralized Controller for Robotic Manipulators*. Master of Engineering Thesis, Department of Electrical Engineering, University of Toronto, 1991.

[25] Y. F. Zheng and J. Y. Luh. Optimal Load Distribution for Two Industrial Robots Handling a Single Object. *Proceedings of the 1988 IEEE Robotics and Automation Conference*, Philidalphia, PA, April 1988.

APPENDIX

A. Proof of Theorem 1

The following definitions are based on material presented in [21]. The convolution of a Laplace transformable signal $f(t)$ and a transfer function matrix $M(s)$ is denoted by Mf, i.e., $Mf = (m * f)(t)$, where * denotes the convolution operator. The l_2-norm of a vector $x \in R^n$, denoted by $\|x\|$, is defined as: $\|x\| = \left(\sum_{j=1}^{n} |x_j|^2 \right)^{1/2}$. The l_2 norm of a matrix $A \in R^{nxn}$, denoted by $\|A\|$, is defined as: $\|A\| = [\max_i \lambda_i (A^T A)]^{½}$, where $\lambda_i(\cdot)$ denotes the eigenvalue of the matrix $(\cdot)$. The $\mathcal{L}^n_\infty$-norm of a transfer matrix P is defined as: $\|P\|_\infty = \sup_{x \in \mathcal{L}^n_\infty - 0} \frac{\|Px\|\infty}{\|x\|\infty}$. Thus $\|P\|\infty$ is the $\mathcal{L}_\infty$-gain of P. Let β denote the norm $\|P\|\infty$, then $\|Px\|_{T\infty} \leq \beta\|x\|_{T\infty}$. A system with input x and output y is said to be bounded-input bounded-output stable (or BIBO stable) if for every $x \in \mathcal{L}_\infty$, $y \in \mathcal{L}_\infty$.

The method presented in [19] is utilized in constructing this proof. Through algebraic operations, the closed-loop dynamics of the multi-manipulator system (7) can alternatively be expressed as

$$\dot{\bar{e}} = \bar{A}\bar{e} + B(\bar{\eta} + u), \tag{18}$$

where $\bar{e} = \begin{bmatrix} x - x^d \\ \dot{x} - \dot{x}^d \end{bmatrix}, \bar{A} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ I \end{bmatrix}$, $\bar{\eta} = \eta_0 + Eu$, $\eta_0 = E\ddot{x}^d + \bar{M}^{-1}\Delta h + \bar{M}^{-1}K_\lambda\Delta\bar{\lambda}$, $u = K\bar{e} + \upsilon$, $E = \bar{M}^{-1}\hat{M} - I$, $\Delta h = \hat{h} - \bar{h}$, $K = \lfloor -K_p - K_\upsilon \rfloor$, $\Delta\bar{\lambda} = \lambda^d - \lambda$, and v is the neural network output. The following assumptions [19] are made regarding the nominal model of the multi-manipulator system: *A1*: For the inertia matrix $\bar{M}$ of the multi-manipulator system, there exist constants M_1 and M_2 such that $M_1 \leq \|\bar{M}^{-1}\| \leq M_2 < \infty$. *A2*: There exists a nonnegative constant $\alpha < 1$ such that $\|\bar{M}^{-1}\hat{M} - I\| \leq \alpha$. *A3*: There exist nonnegative constants δ and ρ such that $\|\hat{h} - \bar{h}\| \leq \delta\|y\| + \rho$, where $y = \lfloor x^T, \dot{x}^T \rfloor^T$. *A4*: There exists nonnegative constant β_λ such that $\|\Delta\bar{\lambda}\|_{T\infty} \leq \beta_\lambda \|\bar{e}\|_{T\infty}$.

Let $G(s) = (sI - \bar{A})^{-1}B$. Then (18) can be expressed as: $\bar{e} = G\in$, $\in = \bar{\eta} + u$, $u = K\bar{e} + \upsilon$, and $\bar{\eta} = \eta_0 + Eu$. Through algebraic operations, $\bar{e}$ and u can be expressed in terms of $\bar{\eta}$ and v as $\bar{e} = (I - GK)^{-1}G\bar{\eta} + (I - GK)^{-1}G\upsilon$ and $u = K(I - GK)^{-1}G\bar{\eta} + (K(I - GK)^{-1}G + I)\upsilon$. Let $P_1 = (I - GK)^{-1}$ G, $P_2 = K(I - GK)^{-1}$ G, and $P^3 = K(I - GK)^{-1}$ $+ I$, then $\bar{e} = P_1\bar{\eta} + P_1\upsilon$, and $u = P_2\bar{\eta} + P_3\upsilon$. Now taking the truncated $\mathcal{L}_\infty$-norm yields

$$\|\bar{e}\|_{T\infty} \leq \beta_1\|\bar{\eta}\|_{T\infty} + \beta_1\|\upsilon\|_{T\infty}, \tag{19}$$

$$\|u\|_{T\infty} \leq \beta_2\|\bar{\eta}\|_{T\infty} + \beta_3\|\upsilon\|_{T\infty}, \tag{20}$$

where $\beta_1 = \|(I - GK)^{-1}G\|_{T\infty}$, $\beta_2 = \|(K - GK)^{-1}G\|_{T\infty}$, and ss. Since $\eta_0 = E\ddot{x}^d + \bar{M}^{-1}\Delta h + M^{-1}K_\lambda\Delta\bar{\lambda}$ so $\|\eta_0\|_{T\infty} \leq M_2\delta^*\|\bar{e}\|_{T\infty} + b$, where $b = \alpha\|\ddot{x}^d\|_{T\infty} + M_2\delta\|y^d\|_{T\infty} + M_2\rho$, $\delta^* = \delta + \beta_\lambda\|K_\lambda\|_{T\infty}$, and $y^2 = \{(x^d)^T, (\dot{x}^d)^T]^T$. It follows that $\|\bar{\eta}\|_{T\infty} \leq M_2\delta^*\|\bar{e}\|_{T\infty} + \alpha\|u\|_{T\infty} + b$. Since the output of the neural network is bounded by construction (due to the *tanh* activation function of the output units), let $\phi = \|\upsilon\|_{T\infty}$. Hence $\begin{bmatrix} \|\bar{\eta}\|_{T\infty} \\ \|u\|_{T\infty} \end{bmatrix} \leq \begin{bmatrix} M_2\delta^*\beta_1 & \alpha \\ \beta_2 & 0 \end{bmatrix} \begin{bmatrix} \|\bar{\eta}\|_{T\infty} \\ \|u\|_{T\infty} \end{bmatrix} + \begin{bmatrix} M_2\delta^*\beta_1\phi + b \\ \beta_3\phi \end{bmatrix}$. Let $Q = \begin{bmatrix} M_2\delta^*\beta_1 & \alpha \\ \beta_2 & 0 \end{bmatrix}$. Then $det(I - Q) = 1 - M_2\delta^* \beta_1 - \alpha\beta_2 \overset{\Delta}{=} \Delta_m$. If $\Delta_m > 0$, then

$$\begin{bmatrix} \|\bar{\eta}\|_{T\infty} \\ \|u\|_{T\infty} \end{bmatrix} \leq \frac{1}{\Delta_m} \begin{bmatrix} 1 & \alpha \\ \beta_2 & 1 - M_2\delta^*\beta_1 \end{bmatrix} \begin{bmatrix} M_2\delta^*\beta_1\phi + b \\ \beta_3\phi \end{bmatrix}. \tag{21}$$

Consequently, from (19) and (21), $\|\bar{e}\|_{T\infty} \leq \frac{\beta_1 b}{\Delta_m} + \frac{\beta_1\phi}{\Delta_m}(\beta_1 M_2\delta^* + \alpha\beta_3) + \beta_1\phi$. Therefore, if the condition $\Delta_m > 0$ is satisfied, then $\bar{\eta}$, u, and $\bar{e}$ are bounded during each trial. It is thus clear that a sufficient condition for BIBO stability of the closed-loop system is $\Delta_m > 0$.

Recall that $\beta_1 = \left\|(I - GK)^{-1}G\right\|_{T\infty}$, and $\beta_2 = \left\|K(I - GK)^{-1}G\right\|_{T\infty}$. It can be seen that the condition $\Delta_m > 0$ can be satisfied by selecting sufficiently large values for K such that β_1 approaches zero and β_2 approaches unity simultaneously.

Here we see that the selection of K has nothing to do with the network output v. Thus the closed-loop system can first be made BIBO stable by selecting appropriate values for K, and then its performance can be improved by properly training the neural network to generate the correct compensating signal.

B. Proof of Corollary 1

From Section 3, the closed-loop dynamics of a multi-manipulator system can be written as: $\overline{M}\ddot{x} + \overline{h} = \hat{M}(\ddot{x}^d + u) + \hat{h} + K_\lambda \Delta\overline{\lambda}$, where $u = K_\upsilon(\dot{x}^d - \dot{x}) + K_p(x^d - x) + \upsilon$. So , where E, Δh, $K_{\lambda,}$ and $\Delta\overline{\lambda}$ are as defined in (18). Therefore, from the modeling assumptions specified in Appendix A, $\|\ddot{x}\|_{T\infty} \le \left(1 + \|E\|_{T\infty}\right)$ $\left\|\ddot{x}^d + u\right\|_{T\infty} + \left\|\overline{M}^{-1}\right\|_{T\infty}$ $\left(\|\Delta h\|_{T\infty} + \beta_\lambda \|K_\lambda\|_{T\infty}\|\overline{e}\|_{T\infty}\right)$ $\le (1 + \alpha)$ $\left(\left\|\ddot{x}^d\right\|_{T\infty} + \|u\|_{T\infty}\right)$ $+ M_2\left(\left(\delta^* + \beta_\lambda \|K_\lambda\|_{T\infty}\right)\|\overline{e}\|_{T\infty} + \delta^* \left\|y^d\right\|_{T\infty} + \rho\right)$, where $y^d = \left[\left(x^d\right)^T, \left(\dot{x}^d\right)^T\right]^T$. Since u and $\overline{e}$ have been proved to be bounded (Theorem 1), it can be concluded that $\ddot{x}$ is bounded.

C. Proof of Corollary 2

From Section 3, the control error Δv in the multi-manipulator system can be written as: $\Delta v = \ddot{x}^d - \ddot{x} + K\overline{e} + \Delta f$, where K is as specified in the definition of (18). Therefore, $\|\Delta v\|_{T\infty} \le \left\|\ddot{x}^d\right\|_{T\infty}$ + $\|\ddot{x}\|_{T\infty} + \|K\overline{e}\|_{T\infty} + M_2\beta_\lambda \|K\lambda\|_{T\infty}\|\overline{e}\|_{T\infty}$. Since $\ddot{x}$ and $\overline{e}$ have been proved to be bounded, it can be concluded that v Δv is also bounded.

D. Proof of Corollary 3

With reference to Figure 2, the dynamics of the network weights during a trial p can be expressed, using the notations defined in Section 5, as [9]

$$\dot{W}_{ij}(p,\xi) = \lambda_n \Delta v_i(p,\xi) g'(v_i(p,\xi))\, \overline{v}_j(p,\xi)$$

$$\dot{R}_{jk}(p,\xi) = \lambda_n g'(\overline{v}_j(p,\xi)) \sum_{i=1}^{I_n} \left(\Delta v_i(p,\xi) g'(v_i(p,\xi)) W_{ij}(p,\xi)\right),$$

$$\dot{S}_{kl}(p,\xi) = \lambda_n g'\left(\overline{\overline{v}}_k(p,\xi) \sum_{j=1}^{J_n} \left(g'(\overline{v}_j(p,\xi)) \sum_{i=1}^{I_n} \left(\Delta v_i(p,\xi)\, g'(v_i(p,\xi))\, W_{ij}(p,\xi) R_{jk}(p,\xi)\right)\right)\right),$$

where $g'(\cdot) = \dfrac{\partial(c \tanh(\cdot))}{\partial(\cdot)}$, and c is a scaling factor.

During the subsequent development, the symbols λ_w, λ_r, and λ_s are used (instead of the general notation λ_n) to indicate that these learning rates are specifically associated with W_{ij}, R_{jk}, and S_{kl} respectively. Since during a given trial p, Δv_i, v_i, $\overline{v}_j, \overline{\overline{v}}_k$, and $g'(\cdot)$ are bounded, i.e., $\|\Delta v_i(p,\xi)\|_{T\infty} \le \gamma_p < \infty$, $\|v_i(p,\xi)\|_{T\infty} \le \tilde{c}$, $\|\overline{v}_j(p,\xi)\|_{T\infty} \le \tilde{c}$, $\|\ \overline{\overline{v}}_k(p,\xi)\|_{T\infty} \le \tilde{c}$, and $\|g'(\cdot)\|_{T\infty} \le \tilde{c}$, where $\tilde{c}$ is the largest scaling factor, it can be concluded that, $\dot{W}_{ij}$, and consequently W_{ij}, are bounded, i.e., $\|\dot{W}_{ij}(p,\xi)\|_{T\infty} = \|\lambda_w \Delta v_i(p,\xi) g'(v_i(p,\xi)) \overline{v}_j(p,\xi)\|_{T\infty} \le \lambda_w \tilde{c}^2 \gamma_p$, $\|W_{ij}(p,\xi)\|_{T\infty} = \left\|\int_0^\xi \dot{W}_{ij}(p,\xi) d\sigma\right\|_{T\infty} \le \lambda_w \tilde{c}^2 \gamma_p \xi$. Similarly, R_{jk} and S_{kl} can be shown to be bounded.

E. Proof of Proposition 1

As defined in Section 4, the generalized weight vector Θ consists of elements of the weight matrices *W*, *R*, and *S*. Hence, $\Delta w_{ij}(p,\xi) = \int_0^\xi \dot{W}_{ij}(p,\sigma)d\sigma, \Delta r_{jk}(p,\xi) = \int_0^\xi \dot{R}_{jk}(p,\sigma)d\sigma$, and $\Delta s_{kl}(p,\xi) = \int_0^\xi \dot{S}_{kl}(p,\sigma)\, d\sigma$.

Now for two successive trials p and $(p+1)$, $\|\Delta w_{ij}(p+1,\xi) - \Delta w_{ij}(p,\xi)\|_{T\infty} \leq \|\Delta w_{ij}(p+1,\xi)\|_{T\infty} + \|\Delta w_{ij}(p,\xi)\|_{T\infty} = \left\|\int_0^\xi \dot{W}_{ij}(p+1,\sigma)d\sigma\right\|_{T\infty} + \left\|\int_0^\xi \dot{W}_{ij}(p,\sigma)d\sigma\right\|_{T\infty} \leq \lambda_w \tilde{c}^2(\gamma_{p+1} + \gamma_p)$. Therefore, to satisfy $\|\Delta w_{ij}(p+1,\xi) - \Delta w_{ij}(p,\xi)\|_{T\infty} << 1$, λ_w can be chosen such that $\lambda_w \tilde{c}^2(\gamma_{p+1} + \gamma_p) << 1$, that is, $\lambda_w << \frac{1}{\tilde{c}^2(\gamma_{p+1} + \gamma_p)}$. Similarly, $\|\Delta r_{jk}(p+1,\xi) - \Delta r_{jk}(p,\xi)\|_{T\infty} \leq \lambda_r^2 \tilde{c}^4 I_n \xi(\gamma_{p+1}^2 + \gamma_p^2)$. To satisfy $\|\Delta r_{jk}(p+1,\xi) - \Delta r_{jk}(p,\xi)\|_{T\infty} << 1$, λ_r can be chosen as: $\lambda_r << \left(\frac{1}{\tilde{c}4 I_n \xi(\gamma_{p+1}^2 + \gamma_p^2)}\right)^{\frac{1}{2}}$. And finally, $\|\Delta s_{kl}(p+1,\xi) - \Delta s_{kl}(p,\xi)\|_{T\infty} \leq$ s. To satisfy $\|\Delta s_{kl}(p+1,\xi) - \Delta s_{kl}(p,\xi)\|_{T\infty} << 1$, λ_s can be chosen as: $\lambda_s << \left(\frac{1}{\tilde{c}^9 I_n^2 j_n \xi^3(\gamma_{p+1}^4 + \gamma_p^4)}\right)^{\frac{1}{4}}$.

It can be seen that by choosing λ_n to be the smallest of the three learning rates λ_w, λ_r, and λ_s, i.e., $\lambda_n = \min[\lambda_w, \lambda_r, \lambda_s]$ the general condition $\|\Delta\theta_m(p+1,\xi) - \Delta\theta_m(p,\xi)\|_{T\infty} << 1$ can be met.

F. Proof of Theorem 2

Since (from Section 4) $J_{\Delta v}(p,\xi) = \frac{1}{2}\Delta v^T(\text{p},\xi)\Delta v(\text{p},\xi)$, from the definition of the $\mathcal{L}_2$-norm, $\mu_p^2 = \|\Delta v(\text{p})\|^2 = 2\int_0^T J_{\Delta v}(\text{p},\xi)d\xi$. The sequence $\langle\mu_p\rangle$ is convergent if there exists some k such that $\mu_k = \mu_{k-1}$. The sequence $\langle\mu_p\rangle$ is strictly monotonically decreasing leading to convergence if for any two successive trials p and $p+1$ (where $p<k-1$), $\mu_{p+1} < \mu_p$.

Now $\mu_{p+1}^2 - \mu_p^2 = 2\int_0^T (J_{\Delta v}(p+1,\xi) - J_{\Delta v}(p,\xi))\, d\xi$. Based on the fact that the change in the control error Δv between any two successive trials p and $(p+1)$ is a direct consequence of the change in the network weights, expanding $J_{\Delta v}(p+1,\xi)$ about $J_{\Delta v}(p,\xi)$, while ignoring the higher order terms (because the learning rate is small), yields $J_{\Delta v}(p+1,\xi) - J_{\Delta v}(p,\xi) \approx \sum_{m=1}^{c\theta}\left[\frac{\partial J_{\Delta v}}{\partial\Theta_m}(\Theta_m(p+1,\xi) - \Theta_m(p,\xi))\right]$, where c_Θ is the total number of weights, i.e., $c_\Theta = I_n \times J_n + J_n \times K_n + K_n \times L_n$. Note that $\frac{\partial J_{\Delta v}(p,\xi)}{\partial\Theta_m(p,\xi)} = \frac{\partial J_{\Delta v}(p,\xi)}{\partial\Delta v(p,\xi)}\frac{\partial\Delta v(p,\xi)}{\partial\Theta_m(p,\xi)} = \Delta v^T(p,\xi)\frac{\partial\Delta v(p,\xi)}{\partial\Theta_m(p,\xi)}$. Since $\Theta_m(p+1,\xi) = \Theta_m(p+1,0) + \Delta\Theta_m(p+1,\xi)$, and $\Theta_m(p,\xi) = \Theta_m(p,0) + \Delta\Theta_m(p,\xi)$, and from Proposition 1, $\|\Delta\Theta_m(p+1,\xi) - \Delta\Theta_m(p,\xi)_{T\infty} << 1$, hence $\Theta_m(p+1,\xi) - \Theta_m(p,\xi) \approx \Theta_m(p+1,0) - \Theta(p,0)$, and consequently

$$\mu_{p+1}^2 - \mu_p^2$$

$$= 2\int_0^T \sum_{m=1}^{c\theta}\left[\Delta v^T(p,\xi)\frac{\partial\Delta v(p,\xi)}{\partial\Theta_m(p,\xi)}(\Theta_m(P=1,0) - \Theta_m(p,0))\right]d\xi$$

$$= 2\sum_{m=1}^{c\theta}\left[\left(\int_0^T \Delta v^T(p,\xi)\frac{\partial\Delta v(p,\xi)}{\partial\Theta_m(p,\xi)}d\xi\right)(\Theta_m(p=1,0) - \Theta_m(p,0))\right].$$

Note that $\Theta_m(p+1,0) - \Theta_m(p,0) = \int_0^T \dot{\Theta}_m(p,\xi)d\xi = -\lambda_n\ \Delta v^T(p,\xi)\ \dfrac{\partial \Delta v(p,\xi)}{\partial \Theta_m(p,\xi)}$. Thus

$$\mu_{p+1}^2 - \mu_p^2$$

$$= 2\sum_{m=1}^{c\theta}\left[\left(\int_0^T \Delta v^T(p,\xi)\frac{\partial \Delta v(p,\xi)}{\partial \Theta_m(p,\xi)}d\xi\right)\left(-\lambda_n\int_0^T \Delta v^T(p,\xi)\frac{\partial \Delta v(p,\xi)}{\partial \Theta_m(p,\xi)}d\xi\right)\right].$$

$$= -2\lambda_n\sum_{m=1}^{c\theta}\left[\left(\int_0^T \Delta v^T(p,\xi)\frac{\partial \Delta v(p,\xi)}{\partial \Theta_m(p,\xi)}d\xi\right)^2\right]$$

$$\leq 0.$$

Now μ_{p+1}^2 - μ_p^2 = 0 if $\sum_{m=1}^{c\theta}\int_0^T \Delta v^T(p,\xi)\left(\dfrac{\partial \Delta v(p,\xi)}{\partial \Theta_m(p,\xi)}\right)d\xi = 0$. But this implies that $\sum_{m=1}^{c\theta}\int_0^T \dot{\Theta}m\, d\xi = 0$, which states that the total change in the weight Θ_m for the trial p is zero. This means that the gradient search conducted by the error-backpropagation algorithm has reached either a global minimum or a local minimum, and so $\mu_{p+1} = \mu_p$. If this is not the case, then μ_{p+1}^2 - $\mu_p^2 < 0$, that is, $\mu_{p+1} < \mu_p$, which implies that the sequence $\langle\mu_p\rangle$ is strictly monotonically decreasing. Since by definition $\mu_p \geq 0$, the fact that $\langle\mu_p\rangle$ is strictly decreasing and convergent implies that $\langle\mu_p\rangle$ is finite. This completes the proof for Theorem 2.

G. Elements of Matrices of the Two-Manipulator System

The elements of the matrices M_1, M_2, C_1, and C_2 are as follows: $M_l\,[1,1] = c_1 + c_4 + c_6 + 2\,(c_2 + c_5)\cos q_{12} + 2c_3\cos(q_{12} + q_{13})$, $M_1\,[1,2] = M_1\,[2,1] = c_1 + c_4 + c_5 \cos q_{12} + 2\,c_2 \cos q_{13} + c_3 \cos(q_{12} + q_{13})$, $M_1\,[1,3] = M_1\,[3,1] = c_1 + c_2 \cos q_{13} + c_3 \cos(q_{12} + q_{13})$, $M_1\,[2,2] = c_1 + c_4 + 2c_2 \cos q_{13}$, $M_1\,[2,3] = c_1 + c_2 \cos q_{13}$, $M_1\,[3,2] = c_1 + c_2 \cos q_{13}$, $M_1\,[3,3] = c_1$, $M_2\,[1,1] = c_4 + c_6 + 2\,c_5 \cos q_{22}$, $M_2\,[1,2] = c_4 + c_5 \cos q_{22}$, $M_2\,[2,1] = c_4 + c_5 \cos q_{22}$, $M_2\,[2,2] = c_4$, $C_1\,[1,2] = -c_3\,(\dot{q}_{11} + \dot{q}_{12} + \dot{q}_{13})\sin(q_{12} + q_{13}) - (c_2 + c_5)\,\dot{q}_{11} \sin q_{12} - c_5\ \dot{q}_{12} \sin q_{12} - c_2\ \dot{q}_{13} \sin q_{13}$, $C_1\,[1,3] = -c_3\,(\dot{q}_{11} + \dot{q}_{12} + \dot{q}_{13})\sin(q_{12} + q_{13}) - c_2\,(\dot{q}_{12} + \dot{q}_{13})\sin q_{13}$, $C_1\,[2,1] = ((c_2 + c5)\sin q_{12} + c_3 \sin(q_{12} + q_{13}))\ \dot{q}_{11} - c_2\ \dot{q}_{13} \sin q_{13}$, $C_1\,[1,1] = -c_3\ (\dot{q}_{12} + \dot{q}_{13})$, $C_1\,[2,2] = -c_2\ \dot{q}_{13} \sin q_{13}$, $C_1\,[2,3] = -c_2\ (\dot{q}_{11} + \dot{q}_{12} + \dot{q}_{13})\sin q_{13}$, $C_1\,[3,1] = c_3\ \dot{q}_{11} \sin(q_{12} + q_{13})$, $C_1\,[3,2] = c_2\ \dot{q}_{12} \sin q_{13}$, $C_1\,[3,3] = C_2\,[2,2] = 0$, $C_2\,[1,1] = -c_5\ \dot{q}_{22} \sin q_{22}$, $C_2\,[1,2] = -c_5\ (\dot{q}_{21} + \dot{q}_{22}) \sin q_{22}$, $C_2\,[2,1] = c_5\ \dot{q}_{21} \sin q_{22}$, where $c_1 = m_3 l_{c3}^2 + I_3$, $c_2 = m_3 l_2 lc_3$, $c_3 = m_3 l_1 l_{c3}$, $c_4 = m_2 l_{c2}^2 + m_3 l_2^2 + I_2$, $c_5 = m_2 l_1 l_{c2} + m_3 l_1 l_2$, $c_6 = m_1\ l_{c1}^2 + m_2 l_1^2 + m_3 l_1^2 + I_1$.