Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2007, Article ID 65028, 23 pages
doi:10.1155/2007/65028

Research Article

Design of Optimal Hybrid Position/Force Controller for a Robot Manipulator Using Neural Networks

Vikas Panwar and N. Sukavanam

Received 21 November 2004; Revised 25 April 2006; Accepted 21 November 2006

Recommended by Semyon M. Meerkov

The application of quadratic optimization and sliding-mode approach is considered for hybrid position and force control of a robot manipulator. The dynamic model of the manipulator is transformed into a state-space model to contain two sets of state variables, where one describes the constrained motion and the other describes the unconstrained motion. The optimal feedback control law is derived solving matrix differential Riccati equation, which is obtained using Hamilton Jacobi Bellman optimization. The optimal feedback control law is shown to be globally exponentially stable using Lyapunov function approach. The dynamic model uncertainties are compensated with a feedforward neural network. The neural network requires no preliminary offline training and is trained with online weight tuning algorithms that guarantee small errors and bounded control signals. The application of the derived control law is demonstrated through simulation with a 4-DOF robot manipulator to track an elliptical planar constrained surface while applying the desired force on the surface.

1. Introduction

In many robotic applications such as assembly, fine polishing, grasping, grinding, and deburring, the robot comes in extensive contact with its environment. These tasks are better dealt with by directly controlling the forces of interaction between the robot and its environment. The task is to exert a desired profile of force in the constrained degrees of freedom while following the reference trajectory in the unconstrained degrees of freedom. This problem is generally referred to as *compliant motion control* or *impedance control* or *hybrid position/force control* problem.

At present, many control algorithms have been proposed for hybrid position/force control. The basic hybrid position/force control scheme was originally proposed by Craig and Raibert [1]. It neglected the dynamic coupling among each of the robot joints and developed the control scheme within the framework of robot joint control systems. Khatib and Burdick [2] subsequently remedied coupling problem by considering the control problem within the framework of operational space. Exact decoupling of motion and force equations and system linearization is realized by McClamroch [3] and Yoshikawa [4]. Generally, constrained robotic systems are modeled as differential equations subject to the algebraic constraints. McClamroch proposed the dynamic models as singular system of differential equations and developed these models for several robot configurations and presented feedback control schemes for these configurations. Yoshikawa formulated the constraints on the end effector of the manipulator by the hypersurfaces in the effector coordinate frame and computed the joint driving force as the sum of the two forces: the joint driving force for achieving the desired trajectory of the effector position and that for achieving the desired trajectory of the force. Hogan [5–7] considered the design of compliant motion which is important when gripping fragile objects or in interacting between cooperating robots. In [8], Su et al. presented the control algorithm using sliding-mode approach. Roy and Whitcomb proposed an adaptive force control algorithm for velocity/position-controlled robot arms in contact with surfaces of unknown linear compliance with exact dynamic considerations [9]. Su along Yury Stepanenko presented an adaptive variable structure control algorithm for constrained motion of robots. It was shown that the objective could be achieved without exact knowledge of robot dynamics and on-line calculation of nonlinear dynamic functions. The other works relating to force control are also reported in literature [10–13].

The quadratic optimization of robotic motion is well addressed by Johansson [14]. Johansson and Spong [15] considered the case of quadratic optimization for impedance control of robot manipulators. They showed to minimize an appropriate control effort and used Hamilton-Jacobi formulation to find optimal control. Recently, the application of neural networks in closed-loop control has become an intensive area of research. Many works on closed-loop application of neural networks are reported in the literature. Lewis et al. have worked extensively in the closed-loop application of neural networks for robot motion control [16–19]. They derived successfully control algorithms with stable tracking while approximating the unknown dynamics of the manipulators with neural networks that are universally considered to be fine approximators and trained them online, removing any preliminary offline training. The works relating to application of neural networks in case of impedance control are reported in [20–22]. In [20, 22], neural networks are used for adaptive compensation of the structured and unstructured uncertainties while Saadia et al. [21] tried to globally treat the problem of the adaptation of robot behavior to various classes of tasks. Chen and Chang [23] modified the dynamic model to contain two sets of state variables and used sliding-mode approach to present the control scheme. In this paper, we also transform the dynamic model to a model with two sets of variables, one describing the constrained motion and the other describing the unconstrained motion. Using Hamilton-Jacobi formulation, an optimal sliding-mode controller is designed. The uncertainties of the model are learned with feedforward neural networks.

The stability analysis of the resulting system is also carried out. The conventional robust controllers such as model-based adaptive controllers require the knowledge of the system dynamics in detail (e.g., in the form of linear in parameters model). Also the fuzzy-rule learning scheme has stability problems. The use of optimal sliding-mode neural network based scheme motivates to overcome the shortcomings of existing adaptive schemes. The proposed control scheme has salient features: (1) it guarantees the stability of the controlled system, (2) it provides an optimal feedback solution to the problem, and (3) the neural network is able to learn the completely unknown system dynamics.

The paper is organized as follows. In Section 2, through appropriate transformation of coordinates, the dynamic model for analyzing the constrained robot system is presented. Section 3 presents the optimal sliding-mode controller design. A review of feedforward neural networks is given in Section 4. The *NN* controller design is presented in Section 5. Numerical simulation results are included in Section 6. Section 7 gives concluding remarks.

2. Dynamic model of constrained robot

Based on the Euler-Lagrangian formulation, in the absence of friction, the motion equation of an n-link rigid, nonredundant constrained robot can be expressed in joint space as [24]

$$M(q)\ddot{q} + C_m(q,\dot{q})\dot{q} + G(q) = \tau + J^T(q)\lambda, \tag{2.1}$$

where $q \in \mathbb{R}^n$ is the joint displacement vector, $M(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C_m(q,\dot{q}) \in \mathbb{R}^n$ is the vector characterizing Coriolis and centrifugal forces, $G(q) \in \mathbb{R}^n$ is the gravitational force, $\tau \in \mathbb{R}^n$ is the joint space torque, $J^T(q) \in \mathbb{R}^{n\times m}$ is a Jacobian matrix associated with the contact surface geometry, and $\lambda \in \mathbb{R}^m$ (the so-called "Lagrange multiplier") is a vector of contact forces exerted normal to surface, described in coordinates relative to the surface. m is the times of constraints. Due to the stiffness of the environment, it is assumed that

$$\lambda = S\Omega(t), \tag{2.2}$$

where S represents the environment stiffness and is a constant positive definite matrix. $\Omega : \mathbb{R}^1 \to \mathbb{R}^m$ is a function. The following properties should be noted about the dynamic structure in (2.1).

Property 2.1. $M(q)$ is a symmetric positive definite matrix and bounded above and below, that is, there exists positive constants α_M and β_M such that

$$\alpha_M I_n \le M(q) \le \beta_M I_n. \tag{2.3}$$

Property 2.2. The matrix $\dot{M}(q) - 2C_m(q,\dot{q})$ is skew symmetric.

It is assumed that the generalized contact surface with which the manipulator comes into contact can be thought of as an intersection of m mutually independent hypersurfaces. The algebraic equation of the surface can be written as

$$\Phi(q) = 0, \tag{2.4}$$

where the mapping $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ is twice continuously differentiable. Differentiating (2.4) with respect to time leads to

$$\dot{\Phi}(q) = \frac{\partial \Phi(q)}{\partial q}\dot{q} = J(q)\dot{q}, \tag{2.5}$$

where $J(q) = \partial\Phi(q)/\partial q$ is the Jacobian matrix associated with the contact surface geometry.

When the motion of the robot is constrained to be on the surface (2.4), only $(n-m)$ coordinates of the position vector can be specified independently. To analyze the behavior between the robot and contact surface, we define $x_u = \Psi(q) = [\psi_1(q), \psi_2(q), \ldots, \psi_{n-m}(q)]^T$, which depicts the unconstrained motion. Also define $x_c = \Phi(q) = [\phi_1(q), \phi_2(q), \ldots, \phi_m(q)]^T$ as vector of constrained coordinates. Differentiation gives us $\dot{x}_u = (\partial\Psi(q)/\partial q)\dot{q} = D(q)\dot{q}$ and $\dot{x}_c = (\partial\Phi(q)/\partial q)\dot{q} = J(q)\dot{q}$. Let $x \equiv [x_u^T \; x_c^T]^T \in \mathbb{R}^n$ then we have $\dot{q} = T(q)\dot{x}$ where $T(q) = \begin{bmatrix} D(q) \\ J(q) \end{bmatrix}^{-1}$; differentiation gives us $\ddot{q} = T(q)\ddot{x} + \dot{T}(q)\dot{x}$. Substituting for $\dot{q}$ and $\ddot{q}$ in (2.1), we obtain

$$M(q)T(q)\ddot{x} + C_1(q,\dot{q})\dot{x} + G(q) = \tau + J^T(q)\lambda, \tag{2.6}$$

where $C_1(q,\dot{q}) = C_m(q,\dot{q})T(q) + M(q)\dot{T}(q)$. Premultiplying both sides with $T^T(q)$, (2.6) is modified to obtain

$$\overline{M}(q)\ddot{x} + \overline{C}(q,\dot{q})\dot{x} + \overline{G}(q) = \overline{\tau} + \overline{J}(q)\lambda, \tag{2.7}$$

where

$$\begin{aligned} &\overline{M}(q) = T^T(q)M(q)T(q), \qquad \overline{C}(q,\dot{q}) = T^T(q)C_1(q,\dot{q}), \\ &\overline{G}(q) = T^T(q)G(q), \qquad \overline{\tau}(q) = T^T(q)\tau(q), \qquad \overline{J}(q) = T^T(q)J^T(q). \end{aligned} \tag{2.8}$$

By exploiting the structure of (2.7), two properties could be obtained.

Property 2.3. $\overline{M}(q)$ is a symmetric positive definite matrix and bounded above and below, that is, there exists positive constants $\overline{\alpha}_M$ and $\overline{\beta}_M$ such that

$$\overline{\alpha}_M I_n \le \overline{M}(q) \le \overline{\beta}_M I_n. \tag{2.9}$$

The proof is given in Appendix A.

Property 2.4. The matrix $\dot{\overline{M}}(q) - 2\overline{C}(q,\dot{q})$ is skew symmetric.

The proof is given in Appendix B.

3. Optimal sliding-mode controller design

In this section, the optimal control algorithm is designed to obtain the desired contact force λ_d and the position trajectories x_d, where $x_d = [\psi_{1d},\ldots,\psi_{(n-m)d}]^T$. To adopt the sliding-mode technique, two sets of sliding functions, $s_u \in \mathbb{R}^{n-m}$ for the unconstrained motion and $s_c \in \mathbb{R}^m$ for the constrained motion, are defined. For the unconstrained motion, the sliding function is chosen as

$$s_u = \dot{e}_u + \Lambda_u e_u, \tag{3.1}$$

where $e_u = x_d - x_u$ and $\dot{e}_u = \dot{x}_d - \dot{x}_u$. $\Lambda_u \in \mathbb{R}^{(n-m)\times(n-m)}$ is a positive definite matrix. Since Λ_u is a positive definite matrix, $s_u = 0$ implies that $x_u(t) \to x_d(t)$ as $t \to \infty$.

The sliding functions for the constrained motion are chosen to be

$$s_c = \dot{e}_c + \Lambda_c e_c, \tag{3.2}$$

where $e_c = x_c - \Delta x_c$, $\Lambda_c \in \mathbb{R}^{m\times m}$ is a positive definite matrix. The force compensator Δx_c is set to as $\Delta x_c = \int \Gamma(\lambda_d - \lambda)dt$. Due to $\lambda = S\Omega(t)$ and $x_c = \Phi(q)$, if $s_c = 0$, we have

$$\dot{\lambda} + S\Gamma(\lambda - \lambda_d) + \Lambda_c\left(\lambda + \int S\Gamma(\lambda_d - \lambda)dt\right) = 0. \tag{3.3}$$

Taking the first derivatives of (3.3) yields

$$\ddot{\lambda} + (S\Gamma + \Lambda_c)\dot{\lambda} + \Lambda_c S\Gamma(\lambda_d - \lambda) = 0. \tag{3.4}$$

To analyze the stability of (3.4), the following Lyapunov function is chosen:

$$V_f = \frac{1}{2}\dot{\lambda}^T\dot{\lambda} + \frac{1}{2}(\lambda - \lambda_d)^T\Lambda_c S\Gamma(\lambda - \lambda_d). \tag{3.5}$$

From (3.4) and (3.5), we have

$$\dot{V}_f = -\dot{\lambda}^T(S\Gamma + \Lambda_c)\dot{\lambda}. \tag{3.6}$$

Therefore, from Lyapunov stability and LaSalle's theorem [24], we conclude that $\dot{\lambda}(t) \to 0$ as $t \to \infty$. This implies that the force λ will equal the desired value λ_d no matter the stiffness K is.

Define $e = [\,e_u^T\ e_c^T\,]^T$ and $s = [\,s_u^T\ s_c^T\,]^T$. Then error dynamics and robot dynamics can be written, respectively, as follows:

$$\dot{e}(t) = -\Lambda e(t) + s(t), \tag{3.7}$$

$$\overline{M}(q)\dot{s}(t) = -\overline{C}(q,\dot{q})s(t) - \overline{\tau}(t) + h(y), \tag{3.8}$$

where $\Lambda = \left[\begin{smallmatrix}\Lambda_u & 0\\ 0 & \Lambda_c\end{smallmatrix}\right]$ and for instance $y(t) = [\,e_u^T\ \dot{e}_u^T\ e_c^T\ \dot{e}_c^T\ e_f^T\ \dot{e}_f^T\ x_d^T\ \dot{x}_d^T\ \ddot{x}_d^T\,]^T$, where $e_f = \lambda_d - \lambda$. The nonlinear function $h(y)$, capturing the unknown dynamics the robot manipulator is

$$h(y) = \overline{M}(q)\begin{bmatrix}\ddot{x}_d + \Lambda_u\dot{e}_u\\ -\Gamma\dot{e}_f + \Lambda_c\dot{e}_c\end{bmatrix} + \overline{C}(q,\dot{q})\begin{bmatrix}\dot{x}_d + \Lambda_u e_u\\ -\Gamma e_f + \Lambda_c e_c\end{bmatrix} + \overline{G}(q) - \overline{J}(q)\lambda. \tag{3.9}$$

Now, define a new control-input vector as

$$u(t) = h(y) - \overline{\tau}(t). \tag{3.10}$$

The new control effort $u(t) \in \mathbb{R}^n$ is to be minimized later. The closed-loop robot dynamics becomes

$$\overline{M}(q)\dot{s}(t) = -\overline{C}(q,\dot{q})s(t) + u(t). \tag{3.11}$$

With (3.7) and (3.11), the following augmented system can be obtained:

$$\dot{\widetilde{x}}(t) = \begin{bmatrix} \dot{e} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} -\Lambda & I \\ 0 & -\overline{M}^{-1}\overline{C} \end{bmatrix} \begin{bmatrix} e \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ \overline{M}^{-1} \end{bmatrix} u(t) \tag{3.12}$$

or

$$\dot{\widetilde{x}}(t) = A(q,\dot{q})\widetilde{x}(t) + B(q)u(t) \tag{3.13}$$

with $A(q,\dot{q}) \in \mathbb{R}^{2n\times 2n}$, $B(q) \in \mathbb{R}^{2n\times 2n}$, and $\widetilde{x}(t) = \left[\, e(t)^T \;\; s(t)^T \,\right] \in \mathbb{R}^{2n\times 1}$. Our control objective is to find the control input $u(t)$ so that the following quadratic performance index $\zeta(u)$ is minimized:

$$\zeta(u) = \int_{t_0}^{\infty} L(\widetilde{x},u)dt \tag{3.14}$$

with the Lagrangian

$$L(\widetilde{x},u) = \frac{1}{2}\widetilde{x}^T(t)Q\widetilde{x}(t) + \frac{1}{2}u^T(t)Ru(t) = \frac{1}{2}\begin{bmatrix} e^T & s^T \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} e \\ s \end{bmatrix} + \frac{1}{2}u^T Ru. \tag{3.15}$$

Hamilton-Jacobi-Bellman optimization. Let $u^o(t)$ denote the required optimal control. A necessary and sufficient condition for $u^o(t)$ to minimize (3.15) subject to (3.12) is that $\exists$ a function $V = V(\widetilde{x},t)$ satisfying Hamilton-Jacobi-Bellman (HJB) equation [25]

$$\frac{\partial V(\widetilde{x},t)}{\partial t} + \min_u \left[\mathrm{H}\left(\widetilde{x}, u, \frac{\partial V(\widetilde{x},t)}{\partial \widetilde{x}}, t \right) \right] = 0, \tag{3.16}$$

where the Hamiltonian of optimization is defined as

$$\mathrm{H}\left(\widetilde{x}, u, \frac{\partial V(\widetilde{x},t)}{\partial \widetilde{x}}, t \right) = L(\widetilde{x},u) + \frac{\partial V(\widetilde{x},t)}{\partial \widetilde{x}}\dot{\widetilde{x}}. \tag{3.17}$$

The function $V(\widetilde{x},t)$ satisfies the following partial differential equation:

$$-\frac{\partial V(\widetilde{x},t)}{\partial t} = L(\widetilde{x},u^o) + \frac{\partial V(\widetilde{x},t)}{\partial \widetilde{x}}\dot{\widetilde{x}}. \tag{3.18}$$

The minimum is attained for the optimal control $u(t) = u^o(t)$ and the Hamiltonian is then given by

$$\mathrm{H}^o = \min_u \left[L(\widetilde{x}, u) + \frac{\partial V(\widetilde{x}, t)}{\partial \widetilde{x}} \dot{\widetilde{x}} \right] = \mathrm{H}\left(\widetilde{x}, u^o, \frac{\partial V(\widetilde{x}, t)}{\partial \widetilde{x}}, t \right) = -\frac{\partial V(\widetilde{x}, t)}{\partial t}. \tag{3.19}$$

Lemma 3.1. *The following function V satisfies the Hamilton-Jacobi-Bellman equation*

$$V = \frac{1}{2}\widetilde{x}^T P(q)\widetilde{x} = \frac{1}{2}\widetilde{x}^T \begin{bmatrix} K & 0 \\ 0 & \overline{M}(q) \end{bmatrix} \widetilde{x}, \tag{3.20}$$

where $K = K^T \in \mathbb{R}^{n\times n}$ is a positive symmetric matrix. The matrices K and Λ in (3.12) can be found by solving the matrix differential Riccati equation

$$\dot{P} + PA + A^T P + Q - PBR^{-1}B^T P^T = 0. \tag{3.21}$$

Now, consider the following relations:

$$\begin{aligned} K = K^T &= -\frac{1}{2}(Q_{12} + Q_{21}), \\ K\Lambda + \Lambda^T K = Q_{11}, &\qquad R^{-1} = Q_{22}. \end{aligned} \tag{3.22}$$

With the choice of these relations, the matrix $P(q)$ given by (3.20) satisfies the Riccati equation (3.21). Then, the optimal control $u^o(t)$ that minimizes (3.14) subject to (3.13) is given by

$$u^o(t) = -R^{-1}B^T P(q)\widetilde{x}. \tag{3.23}$$

See Appendix C for proof.

With the optimal-feedback control law $u^o(t)$ calculated using (3.23), the torques $\overline{\tau}(t)$ to apply to the robotic system are computed according to the following control input:

$$\overline{\tau}^o(t) = h(y) - u^o(t). \tag{3.24}$$

Stability analysis. Suppose that the matrices K and Λ exist that satisfy the hypotheses of Lemma 3.1 and the matrix $P(q)$ is bounded over (t_0, ∞). Then using the optimal-feedback control law in (3.13) and (3.23), the controlled nonlinear system becomes

$$\dot{\widetilde{x}}(t) = \{A(q, \dot{q}) - B(q)R^{-1}B^T(q)P\}\widetilde{x}(t). \tag{3.25}$$

This is globally exponentially stable (GES) with respect to the origin in $\mathbb{R}^{2n}$.

Proof. The quadratic function $V(\widetilde{x}, t)$ is chosen to be the candidate Lyapunov function as it is positive radially, increasing with $\|\widetilde{x}\|$. It is continuous and has a unique minimum at the origin. Now, it remains to show that $dV/dt < 0$ for all $\|\widetilde{x}\| \neq 0$. From the solution of the HJB equation (3.16), we have

$$\frac{dV(\widetilde{x}, t)}{dt} = -L(\widetilde{x}, u^o). \tag{3.26}$$

Substituting (3.23) in (3.15), it follows that

$$\begin{aligned}\frac{dV(\tilde{x},t)}{dt} &= -\frac{1}{2}\left\{\tilde{x}^T Q\tilde{x} + \left(R^{-1}B^T P\tilde{x}\right)^T R\left(R^{-1}B^T P\tilde{x}\right)\right\} \\ &= -\frac{1}{2}\left\{\tilde{x}^T Q\tilde{x} + \left(B^T P\tilde{x}\right)^T \left(R^{-1}\right)^T R R^{-1}\left(B^T P\tilde{x}\right)\right\} \\ &= -\frac{1}{2}\left\{\tilde{x}^T Q\tilde{x} + \left(B^T P\tilde{x}\right)^T R^{-1}\left(B^T P\tilde{x}\right)\right\} < 0 \quad \forall t > 0 \|\tilde{x}\| \neq 0.\end{aligned} \tag{3.27}$$

The time derivative of the Lyapunov function is negative definite and from Lyapunov second theorem it follows that the nonlinear control system (3.25) is globally exponentially stable with respect to the origin in $\mathbb{R}^{2n}$. □

4. Feedforward neural networks

A two-layer feedforward neural network (FFNN) with n input units, m output units, and N units in the hidden layer, is shown in Figure 4.1. The output vector y is determined in terms of the input vector x by the formula

$$y_i = \sum_{j=1}^{N}\left[w_{ij}\sigma\left(\sum_{k=1}^{n} v_{jk}x_k + \theta_{vj}\right) + \theta_{wi}\right]; \quad i = 1,\ldots,m, \tag{4.1}$$

where $\sigma(\cdot)$ are the activation functions of the neurons of the hidden layer. The inputs-to-hidden-layer interconnection weights are denoted by v_{jk} and the hidden-layer-to-outputs interconnection weights by w_{ij}. The bias weights are denoted by θ_{vj}, θ_{wi}. There are many classes of activation functions, for example, sigmoid, hyperbolic tangent, and Gaussian. The sigmoid activation function used in our work is given by

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{4.2}$$

By collecting all the NN weights v_{jk}, w_{ij} into matrices of weights V^T, W^T, we can write the NN equation in terms of vectors as

$$y = W^T\sigma(V^T x) \tag{4.3}$$

with the vector of activation functions defined by $\sigma(z) = [\,\sigma(z_1)\ \cdots\ \sigma(z_n)\,]^T$ for a vector $z \in \mathbb{R}^n$. The bias weights are included as the first column of the weight matrices. To accommodate bias weights, the vectors x and $\sigma(\cdot)$ need to be augmented by replacing 1 as their first element, for example, $x \equiv [\,1\ x_1\ x_2\ \cdots\ x_n\,]^T$.

Function approximation property. Let $f(x)$ be a smooth function from $\mathbb{R}^n$ to $\mathbb{R}^m$. Let $U_x \subseteq \mathbb{R}^n$ then for $x \in U_x$ there exists some number of hidden layer neurons N and weights W and V such that [26]

$$f(x) = W^T\sigma(V^T x) + \varepsilon. \tag{4.4}$$

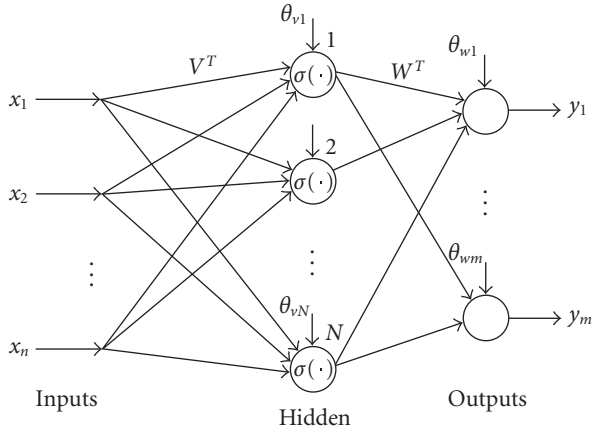

Figure 4.1

The value of ε is called the *NN* functional approximation error. In fact, for any choice of a positive number ε_N, one can find an *NN* such that $\varepsilon < \varepsilon_N$ in U_x. For a specified value of ε_N, the ideal approximating *NN* weights exist. The, an estimate of $f(x)$ can be given by

$$\hat{f}(x) = \widehat{W}^T \sigma(\hat{V}^T x), \tag{4.5}$$

where $\widehat{W}$ and $\hat{V}$ are estimates of the ideal *NN* weights that are provided by some on-line weight-tuning algorithms.

Error backpropagation algorithm. This is a common weight tuning algorithm that is based on gradient descent algorithm. If the *NN* is training offline to match specified exemplar pairs (x_d, y_d), with x_d the ideal *NN* input that yields the desired *NN* output y_d, then the continuous-time version of the backpropagation algorithm for the two-layer *NN* is given by

$$\dot{\widehat{W}} = F\sigma(\hat{V}^T x_d)E^T, \qquad \dot{\hat{V}} = Gx_d(\hat{\sigma}'^T \widehat{W}E)^T, \tag{4.6}$$

where F, G are positive definite design learning parameter matrices. The backpropagated error E is selected as the desired *NN* output minus the actual *NN* output $E = y_d - y$. For the scalar sigmoid activation function (4.2), the hidden-layer output gradient $\hat{\sigma}'$ is

$$\hat{\sigma}' \equiv \operatorname{diag}\{\sigma(\hat{V}^T x_d)\}[I - \operatorname{diag}\{\sigma(\hat{V}^T x_d)\}], \tag{4.7}$$

where I denotes the identity matrix, and $\operatorname{diag}\{z\}$ means a diagonal matrix whose diagonal elements are the components of the vector z. In the next section, design of *NN* controller is presented.

5. *NN* controller design

We will use a feedforward neural network to approximate the nonlinear function given in (3.9)

$$h(y) = W^T\sigma(V^T y) + \varepsilon. \tag{5.1}$$

The functional approximation error can be made arbitrarily small by selecting appropriate weights of the network. Then a functional estimate $\hat{h}(y)$ of $h(y)$ can be written as

$$\hat{h}(y) = \widehat{W}^T\sigma(\hat{V}^T y), \tag{5.2}$$

where $\widehat{W}$ and $\hat{V}$ are estimated *NN* weights. Then the external torque is given by

$$\overline{\tau}^o(t) = \widehat{W}^T\sigma(\hat{V}^T y) - u^o(t) - v(t), \tag{5.3}$$

where $v(t)$ is the robustifying term that is given by

$$v(t) = k_x \frac{s(t)}{\Delta(||s(t)||,\rho)}, \tag{5.4}$$

where

$$\Delta(||s(t)||,\rho) = \begin{cases} ||s(t)||, & ||s(t)|| > \rho, \\ \rho, & ||s(t)|| \le \rho. \end{cases} \tag{5.5}$$

Then (3.11) becomes

$$\overline{M}(q)\dot{s}(t) = -\overline{C}(q,\dot{q})s(t) + W^T\sigma(V^T y) + \varepsilon - \widehat{W}^T\sigma(\hat{V}^T y) + u^o(t) + v(t). \tag{5.6}$$

In order to proceed further, we need the following definitions [17].

Definition 5.1. The solution of a nonlinear system with state $y(t) \in \mathbb{R}^n$ is uniformly ultimately bounded (UUB) if there exists a compact set $U_y \subset \mathbb{R}^n$ such that for all $y(t_0) = y_0 \in U_y$, there exists a $\delta > 0$ and a number $T(\delta, y_0)$ such that $\|y(t)\| < \delta$ for all $t \ge t_0 + T$.

Definition 5.2. Define the norm of a vector $y \in \mathbb{R}^n$ as $\|y\| = \sqrt{y_1^2 + y_2^2 + \cdots + y_n^2}$ and the norm of a matrix $A \in \mathbb{R}^{m\times n}$ as $\|A\| = \sqrt{\lambda_{\max}[A^T A]}$, where $\lambda_{\max}[\cdot]$ and $\lambda_{\min}[\cdot]$ are the largest and smallest eigenvalues of a matrix. To be specific, denote the p-norm by $\|\cdot\|_p$ and the absolute value as $|\cdot|$.

Definition 5.3. Given $A = [a_{ij}]$, $B \in \mathbb{R}^{m\times n}$, the Frobenius norm is defined by $\|A\|_F^2 = \mathrm{tr}\{A^T A\} = \sum_{i,j} a_{ij}^2$ with $\mathrm{tr}\{\cdot\}$ as the trace operator. The associated inner product is $\langle A, B\rangle_F = \mathrm{tr}\{A^T B\}$. The Frobenius norm is compatible with 2-norm so that $\|Ay\|_2 \le \|A\|_F\|y\|_2$.

Definition 5.4. For notational convenience, define the matrix of all *NN* weights as $Z \equiv \mathrm{diag}\{W, V\}$ and the weight estimation errors as $\widetilde{W} = W - \widehat{W}$, $\tilde{V} = V - \hat{V}$, and $\widetilde{Z} = Z - \hat{Z}$. The ideal *NN* weights bounded so that $\|Z\|_F \le Z_M$ with known Z_M. Also define the hidden-layer output error for a given y as $\tilde{\sigma} = \sigma - \hat{\sigma} = \sigma(V^T y) - \sigma(\hat{V}^T y)$.

Adding and subtracting $W^T\sigma(\hat{V}^T y)$ in (5.6), we get

$$\overline{M}(q)\dot{s}(t) = -\overline{C}(q,\dot{q})s(t) + W^T\left(\sigma(V^T y) - \sigma(\hat{V}^T y)\right) + \widetilde{W}^T\sigma(\hat{V}^T y) + u^o(t) + v(t) + \varepsilon. \tag{5.7}$$

Again adding and subtracting $\widehat{W}^T\tilde{\sigma}$ in (5.7), we get

$$\overline{M}(q)\dot{s}(t) = -\overline{C}(q,\dot{q})s(t) + \widetilde{W}^T\tilde{\sigma} + \widehat{W}^T\tilde{\sigma} + \widetilde{W}^T\hat{\sigma} + u^o(t) + v(t) + \varepsilon. \tag{5.8}$$

The Taylor series expansion of $\sigma(V^T y)$ about given $V^T y$ gives us

$$\sigma(V^T y) = \sigma(\hat{V}^T y) + \sigma'(\hat{V}^T y)\tilde{V}^T y + O(\tilde{V}^T y)^2 \tag{5.9}$$

with $\sigma'(\hat{z}) = (d\sigma(z)/dz)|_{z=\hat{z}}$ the Jacobian matrix and $O(z)^2$ denoting terms of second order. Denoting $\hat{\sigma}' = \sigma'(\hat{V}^T y)$, we have $\tilde{\sigma} = \hat{\sigma}'\tilde{V}^T y + O(\tilde{V}^T y)^2$. Replacing for $\tilde{\sigma}$ in (5.8), we get

$$\overline{M}(q)\dot{s}(t) = -\overline{C}(q,\dot{q})s(t) + \widehat{W}^T\hat{\sigma}'\tilde{V}^T y + \widetilde{W}^T\hat{\sigma} + u^o(t) + v(t) + w_1, \tag{5.10}$$

where the disturbance terms are

$$w_1(t) = \widetilde{W}^T\hat{\sigma}'\tilde{V}^T y + W^T O(\tilde{V}^T y)^2 + \varepsilon. \tag{5.11}$$

Finally, adding and subtracting $\widetilde{W}^T\hat{\sigma}'\hat{V}^T y$ (5.10) becomes

$$\overline{M}(q)\dot{s}(t) = -\overline{C}(q,\dot{q})s(t) + \widehat{W}^T\hat{\sigma}'\tilde{V}^T y + \widetilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T y) + u^o(t) + v(t) + w, \tag{5.12}$$

where the disturbance terms are

$$w(t) = \widetilde{W}^T\hat{\sigma}'V^T y + W^T O(\tilde{V}^T y)^2 + \varepsilon. \tag{5.13}$$

The state-space description of (5.13) can be given by

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + B\left[u^0(t) + \widetilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T y) + \widehat{W}^T\hat{\sigma}'\tilde{V}^T y + v + w\right]. \tag{5.14}$$

Inserting the optimal feedback control law (3.23) into (5.14), we obtain

$$\dot{\tilde{x}}(t) = (A - BR^{-1}B^T P)\tilde{x}(t) + B\left[\widetilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T y) + \widehat{W}^T\hat{\sigma}'\tilde{V}^T y + v + w\right]. \tag{5.15}$$

NN weights update law. With positive definite design parameters F, G, and $\kappa > 0$, the adaptive NN weight update law is given by

$$\begin{aligned}\dot{\widehat{W}} &= F\hat{\sigma}(B^T P\tilde{x})^T - F\hat{\sigma}'\hat{V}^T y(B^T P\tilde{x})^T - \kappa F||B^T P\tilde{x}||\widehat{W},\\ \dot{\hat{V}} &= Gy(\hat{\sigma}'\widehat{W}B^T P\tilde{x})^T - \kappa G||B^T P\tilde{x}||\hat{V}.\end{aligned} \tag{5.16}$$

Then the errors state vector $\tilde{x}$ and $\widetilde{W}$, $\tilde{V}$ are uniformly ultimately bounded and the errors $\tilde{x}$ can be made arbitrary small by adjusting the weights.

Proof. Consider the following Lyapunov function candidate:

$$L = \frac{1}{2}\tilde{x}^T P(q)\tilde{x} + \frac{1}{2}\operatorname{tr}\left(\widetilde{W}^T F^{-1}\widetilde{W}\right) + \frac{1}{2}\operatorname{tr}\left(\tilde{V}^T G^{-1}\tilde{V}\right). \tag{5.17}$$

The time derivative $\dot{L}$ of the Lyapunov function becomes

$$\dot{L} = \dot{\tilde{x}}^T P(q)\tilde{x} + \frac{1}{2}\tilde{x}^T \dot{P}(q)\tilde{x} + \operatorname{tr}\left(\widetilde{W}^T F^{-1}\dot{\widetilde{W}}\right) + \operatorname{tr}\left(\tilde{V}^T G^{-1}\dot{\tilde{V}}\right). \tag{5.18}$$

Evaluating (5.18) along (5.15), we get

$$\begin{aligned}\dot{L} = {} & \tilde{x}^T PA\tilde{x} - \tilde{x}^T PBR^{-1}B^T P\tilde{x} + \tilde{x}^T PB\left[\widetilde{W}^T\left(\hat{\sigma} - \hat{\sigma}'\hat{V}^T y\right) + \widehat{W}^T\hat{\sigma}'\tilde{V}^T y + v + w\right] \\ & + \frac{1}{2}\tilde{x}^T\dot{P}(q)\tilde{x} + \operatorname{tr}\left(\widetilde{W}^T F^{-1}\dot{\widetilde{W}}\right) + \operatorname{tr}\left(\tilde{V}^T G^{-1}\dot{\tilde{V}}\right).\end{aligned} \tag{5.19}$$

From Riccati equation, we have

$$\dot{P} + PA + A^T P + Q - PBR^{-1}B^T P = 0 \tag{5.20}$$

or we can transform it into the following form:

$$\frac{1}{2}PA + \frac{1}{2}A^T P + \frac{1}{2}\dot{P} = -\frac{1}{2}Q + \frac{1}{2}PBR^{-1}B^T P. \tag{5.21}$$

Using $\tilde{x}^T PA\tilde{x} = (1/2)\tilde{x}^T\{PA + A^T P\}\tilde{x}$ and using (5.21), we obtain

$$\begin{aligned}\dot{L} = {} & -\frac{1}{2}\tilde{x}^T Q\tilde{x} - \frac{1}{2}\tilde{x}^T PBR^{-1}B^T P\tilde{x} + \tilde{x}^T PB\left[\widetilde{W}^T\left(\hat{\sigma} - \hat{\sigma}'\hat{V}^T y\right) + \widehat{W}^T\hat{\sigma}'\tilde{V}^T y + v + w\right] \\ & + \operatorname{tr}\left(\widetilde{W}^T F^{-1}\dot{\widetilde{W}}\right) + \operatorname{tr}\left(\tilde{V}^T G^{-1}\dot{\tilde{V}}\right).\end{aligned} \tag{5.22}$$

Since $\dot{\widetilde{W}} = -\dot{\widehat{W}}$, $\dot{\tilde{V}} = -\dot{\hat{V}}$ and applying adaptive learning rule (5.16), we have

$$\begin{aligned}\dot{L} = {} & -\frac{1}{2}\tilde{x}^T Q\tilde{x} - \frac{1}{2}\tilde{x}^T PBR^{-1}B^T P\tilde{x} + \tilde{x}^T PB(v + w) + \tilde{x}^T PB\left[\widetilde{W}^T\left(\hat{\sigma} - \hat{\sigma}'\hat{V}^T y\right) + \widehat{W}^T\hat{\sigma}'\tilde{V}^T y\right] \\ & + \operatorname{tr}\left(-\widetilde{W}^T\hat{\sigma}\tilde{x}^T PB + \widetilde{W}^T\hat{\sigma}'\tilde{V}^T y\tilde{x}^T PB + \kappa\widetilde{W}^T\left\|B^T P\tilde{x}\right\|\widehat{W}\right) \\ & + \operatorname{tr}\left(-\tilde{V}^T y\tilde{x}^T PB\widehat{W}^T\hat{\sigma}' + \kappa\tilde{V}^T\left\|B^T P\tilde{x}\right\|\hat{V}\right) \\ = {} & -\frac{1}{2}\tilde{x}^T Q\tilde{x} - \frac{1}{2}\tilde{x}^T PBR^{-1}B^T P\tilde{x} + \tilde{x}^T PB(v + w) \\ & + \operatorname{tr}\left(-\widetilde{W}^T\hat{\sigma}\tilde{x}^T PB + \widetilde{W}^T\hat{\sigma}'\tilde{V}^T y\tilde{x}^T PB + \widetilde{W}^T\left(\hat{\sigma} - \hat{\sigma}'\hat{V}^T y\right)\tilde{x}^T PB + \kappa\widetilde{W}^T\left\|B^T P\tilde{x}\right\|\widehat{W}\right) \\ & + \operatorname{tr}\left(-\tilde{V}^T y\tilde{x}^T PB\widehat{W}^T\hat{\sigma}' + \tilde{V}^T y\tilde{x}^T PB\widehat{W}^T\hat{\sigma}' + \kappa\tilde{V}^T\left\|B^T P\tilde{x}\right\|\hat{V}\right)\end{aligned} \tag{5.23}$$

or

$$\begin{aligned}\dot{L} = &-\frac{1}{2}\widetilde{x}^T Q\widetilde{x} - \frac{1}{2}\widetilde{x}^T PBR^{-1}B^T P\widetilde{x} + \widetilde{x}^T PB(v+w)\\ &+ \operatorname{tr}\left(\kappa\widetilde{W}^T||B^T P\widetilde{x}||\widehat{W}\right) + \operatorname{tr}\left(\kappa\widetilde{V}^T||B^T P\widetilde{x}||\widehat{V}\right).\end{aligned} \tag{5.24}$$

Since $B^T P = [0\ I]$, $PB = [0\ I]^T$, and assuming $v = 0$, $w = 0$ (though v and w can be shown to be bounded), we get

$$\dot{L} \le -\frac{1}{2}\|\widetilde{x}\|^2\{\lambda_{\min}(Q) + \lambda_{\min}(R^{-1})\} + \kappa\|\widetilde{x}\|\left(\|\widetilde{Z}\|_F Z_M - \|\widetilde{Z}\|_M^2\right). \tag{5.25}$$

The following inequality is used in the derivation of (5.25),

$$\operatorname{tr}\left(\widetilde{Z}^T\widehat{Z}\right) = \operatorname{tr}\left(\widetilde{Z}^T(Z-\widetilde{Z})\right) = \langle\widetilde{Z},Z\rangle_F - \|\widetilde{Z}\|_F^2. \tag{5.26}$$

Using Cauchy-Schwartz inequality, we have $\langle\widetilde{Z},Z\rangle_F \le \|\widetilde{Z}\|_F Z_M$, and then (5.25) is derived. Completing the square term, we get

$$\dot{L} \le -\frac{1}{2}\|\widetilde{x}\|\left[\|\widetilde{x}\|\{\lambda_{\min}(Q) + \lambda_{\min}(R^{-1})\} + \kappa\left(\|\widetilde{Z}\|_F - \frac{1}{2}Z_M\right)^2 - \frac{1}{4}\kappa Z_M^2\right]. \tag{5.27}$$

The expression for $\dot{L}$ given by (5.27) remains negative as long as the quantity in the bracket is positive, that is, either (5.28) or (5.29) hold

$$\|\widetilde{x}\| \ge \frac{(1/4)\kappa Z_M^2}{\{\lambda_{\min}(Q) + \lambda_{\min}(R^{-1})\}} \equiv C_{\widetilde{x}}, \tag{5.28}$$

$$\|\widetilde{Z}\|_F \ge \sqrt{\frac{1}{4}\kappa Z_M^2} + \frac{1}{2}Z_M \equiv C_{\widetilde{Z}}, \tag{5.29}$$

where $C_{\widetilde{x}}$ *and* $C_{\widetilde{Z}}$ are the convergence regions. According to Lyapunov theory and LaSalle extension, the UUB of $\widetilde{x}$ and $\widetilde{W}$, $\widetilde{V}$ is proved. □

6. Simulation results

The simulation has been performed for a 4-DOF robotic manipulator (see Figure 6.1) in contact with a planar environment moving along an ellipse. The mathematical model of the manipulator is expressed as

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14}\\ m_{21} & m_{22} & m_{23} & m_{24}\\ m_{31} & m_{32} & m_{33} & m_{34}\\ m_{41} & m_{42} & m_{43} & m_{44}\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\\ \ddot{q}_3\\ \ddot{q}_4\end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14}\\ c_{21} & c_{22} & c_{23} & c_{24}\\ c_{31} & c_{32} & c_{33} & c_{34}\\ c_{41} & c_{42} & c_{43} & c_{44}\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\\ \dot{q}_4\end{bmatrix} + \begin{bmatrix} g_1\\ g_2\\ g_3\\ g_4\end{bmatrix} = \begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\\ \tau_4\end{bmatrix} + J^T(q)\lambda, \tag{6.1}$$

where the mass matrix and gravity terms are given as follows:

$$
\begin{aligned}
m_{11} &= \left(\frac{m_2}{3} + m_3 + m_4\right)a_2^2c_2^2 + \left(\frac{m_3}{3} + m_4\right)a_3^2c_{23}^2 + \frac{m_4a_4^2c_{234}^2}{3} \\
&\quad + (m_3 + 2m_4)a_2a_3c_2c_{23} + m_4a_2a_4c_2c_{234} + m_4a_3a_4c_{23}c_{234}, \\
m_{12} &= m_{21} = m_{13} = m_{31} = m_{14} = m_{41} = 0, \\
m_{22} &= \left(\frac{m_2}{3} + m_3 + m_4\right)a_2^2 + \left(\frac{m_3}{3} + m_4\right)a_3^2 + \frac{m_4a_4^2}{3} \\
&\quad + (m_3 + 2m_4)a_2a_3c_3 + m_4a_2a_4c_{34} + m_4a_3a_4c_4, \\
m_{23} &= m_{32} = \left(\frac{m_3}{3} + m_4\right)a_3^2 + \frac{m_4a_4^2}{3} \\
&\quad + \left(\frac{m_3}{2} + m_4\right)a_2a_3c_3 + \frac{m_4a_2a_4c_{34}}{2} + m_4a_3a_4c_4, \\
m_{24} &= m_{42} = \frac{m_4a_4^2}{3} + \frac{m_4a_2a_4c_{34}}{2} + \frac{m_4a_3a_4c_4}{2}, \\
m_{33} &= \left(\frac{m_3}{3} + m_4\right)a_3^2 + \frac{m_4a_4^2}{3} + m_4a_3a_4c_4, \\
m_{34} &= m_{43} = \frac{m_4a_4^2}{3} + \frac{m_4a_3a_4c_4}{2}, \qquad m_{44} = \frac{m_4a_4^2}{3}, \\
g_1 &= 0, \qquad g_2 = \left(\frac{m_2a_2c_2}{2} + m_3\left(a_2c_2 + \frac{a_3c_{23}}{2}\right) + m_4\left(a_2c_2 + a_3c_{23} + \frac{a_4c_{234}}{2}\right)\right)g, \\
g_3 &= \left(\frac{m_3a_3c_{23}}{2} + m_4\left(a_3c_{23} + \frac{a_4c_{234}}{2}\right)\right)g, \qquad g_4 = \frac{m_4a_4c_{234}}{2}g,
\end{aligned}
\tag{6.2}
$$

where

$$
\begin{aligned}
&c_i = \cos(q_i), \qquad c_{23} = \cos(q_2 + q_3), \qquad c_{234} = \cos(q_2 + q_3 + q_4), \\
&c_{34} = \cos(q_3 + q_4), \qquad s_i = \sin(q_i), \qquad s_{23} = \sin(q_2 + q_3), \\
&s_{234} = \sin(q_2 + q_3 + q_4), \qquad s_{34} = \sin(q_3 + q_4), \quad i = 1,2,3,4.
\end{aligned}
\tag{6.3}
$$

The parameter values for the manipulator model are set to be

$$
\begin{aligned}
&m_1 = 12.0, \qquad m_2 = 9.0, \qquad m_3 = 7.0, \qquad m_4 = 6.0, \\
&a_1 = 5.0, \qquad a_2 = 4.0, \qquad a_3 = 3.0, \qquad a_4 = 2.0, \qquad g = 9.8.
\end{aligned}
\tag{6.4}
$$

Let (x, y, z, θ) denote the end-effector configuration of the robot manipulator then interacting environment under consideration is described as the intersection of the following surfaces: $b^2x^2 + a^2y^2 = a^2b^2$ and $z = c$, where a, b, and c are constants. The constant values are taken to be $a = 3$, $b = 4$, $c = 9$. The end-effector configuration and constraint

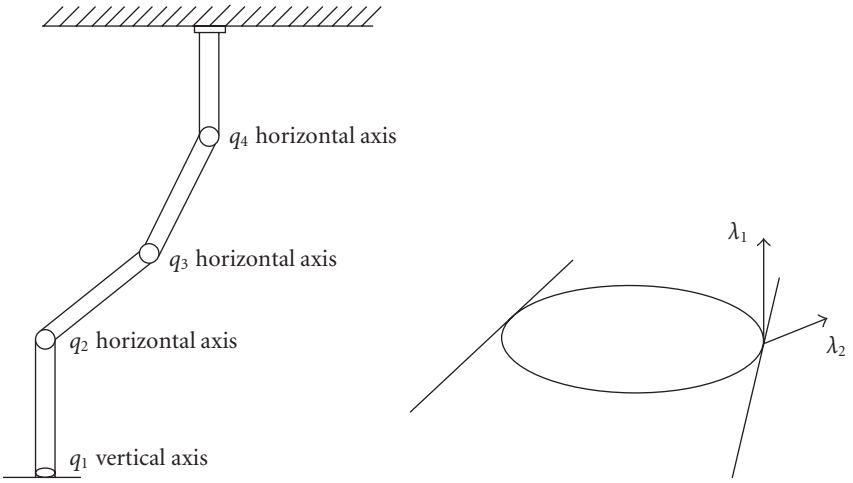

Figure 6.1

equations can be described in the joint space as

$$\begin{aligned} x &= (a_2c_2 + a_3c_{23} + a_4c_{234})c_1, \\ y &= (a_2c_2 + a_3c_{23} + a_4c_{234})s_1, \\ z &= a_1 + a_2s_2 + a_3s_{23} + a_4s_{234}, \\ \theta &= q_2 + q_3 + q_4, \\ \phi_1(q_1,q_2,q_3,q_4) &\equiv (b^2c_1^2 + a^2s_1^2)(a_2c_2 + a_3c_{23} + a_4c_{234})^2 - 1 = 0, \\ \phi_2(q_1,q_2,q_3,q_4) &\equiv a_1 + a_2s_2 + a_3s_{23} + a_4s_{234} - c = 0. \end{aligned} \tag{6.5}$$

Define $x_u = [\psi_1, \psi_2]^T$ and $x_c = [\phi_1, \phi_2]^T$, where $\psi_1 = q_1$, $\psi_2 = q_2 + q_3 + q_4$. The weighting matrices used in defining the performance index are as follows:

$$Q_{11} = 10I_4, \qquad Q_{12} = -\begin{bmatrix} 0.5I_2 & 0 \\ 0 & 5I_2 \end{bmatrix}, \qquad Q_{21} = Q_{12}^T, \qquad Q_{22} = R^{-1} = 30I_4. \tag{6.6}$$

Solving for the matrices K and Λ, we get

$$K = \begin{bmatrix} 0.5I_2 & 0 \\ 0 & 5I_2 \end{bmatrix}, \qquad \Lambda = \begin{bmatrix} 10I_2 & 0 \\ 0 & I_2 \end{bmatrix}. \tag{6.7}$$

Also we have set $\Gamma = 1$. The control objective is to let the end-effector track the elliptical trajectory on the planar surface while maintaining a desired orientation along the surface. The joint q_1 is desired to track the desired trajectory $q_{1d}(t)$ and desired orientation is θ_d.

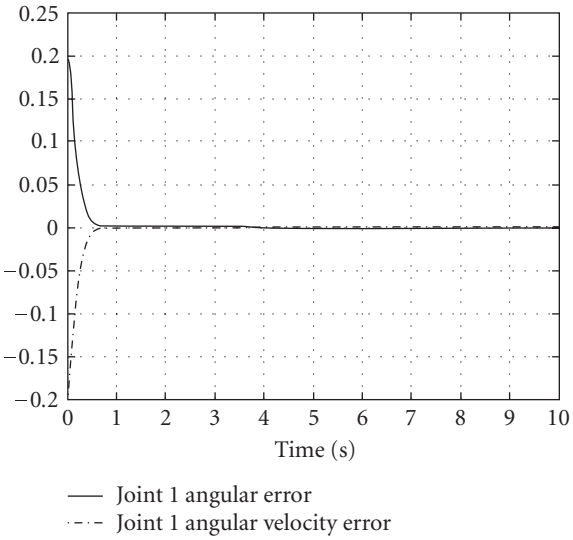

Figure 6.2. Difference between desired and actual joint 1 motion with exactly known parameters.

The Lagrangian multiplier λ is set to the desired constant vector $[\lambda_{d1}\ \lambda_{d2}]^T$. In numerical simulation, we have chosen

$$\begin{aligned} \psi_{1d}(t) = q_{1d}(t) &= \tan^{-1}\left(\frac{b}{a}\tan(\omega t)\right), \\ \psi_{2d}(t) = \theta_d(t) &= (q_2 + q_3 + q_4)_d = 90^\circ, \\ \lambda_{1d} &= 20, \qquad \lambda_{2d} = 20. \end{aligned} \tag{6.8}$$

With $\omega = 0.1$ and $\rho = .001$ for the robustifying term, Figures 6.2–6.4 depict the performance of the designed optimal sliding-mode controller with perfectly known model parameter values. In case of unknown parameter values, the FFNN-based controller is used to learn the unknown robot dynamics. The architecture of the FFNN is composed of 18 input units and 1 bias unit, 24 hidden sigmoidal units and 1 bias unit and 4 output units. The learning rate in the weight-tuning algorithm is $F = 200I_{25}$, $G = 200I_{19}$, and $\kappa = .0005$. The whole system is simulated for 10 seconds. Figures 6.5–6.8 show the ability of the FFNN controller to learn the unknown dynamical behavior of the system and produce desired response.

7. Conclusion

In this paper, the design of an optimal hybrid motion and force control scheme is presented for a constrained robotic manipulator with unknown dynamics. The optimal control law is derived using HJB optimization. An online adaptive neural network controller, using a two-layer feedforward neural network, is proposed to compensate for the dynamic model of the constrained robot. It has been shown that the entire system depends on the user-specified performance index matrices Q and R. The stability of the overall system

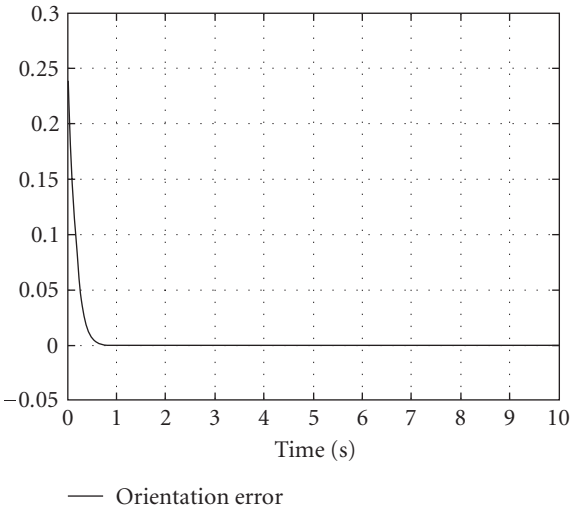

Figure 6.3. Difference between desired and actual end-effector orientation with exactly known parameters.

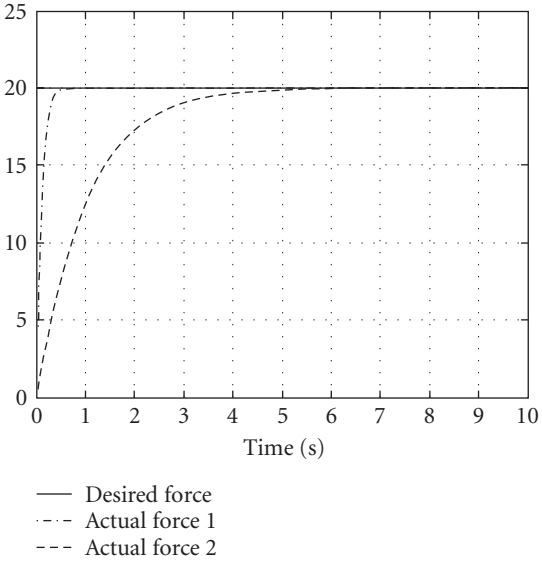

Figure 6.4. Difference between desired and actual contact forces with exactly known parameters.

is proved using Lyapunov function that is generated by weighting matrices. Simulation of a 4-DOF robot manipulator has been used to illustrate the control methodology. The simulation results show that the feedforward neural network with the on-line updating law can compensate the robot dynamics efficiently.

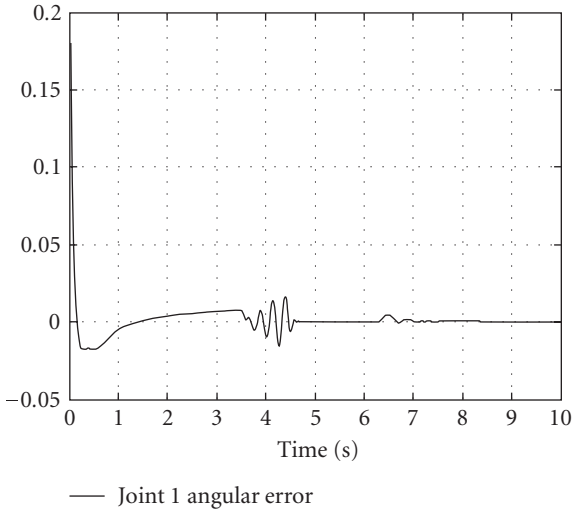

Figure 6.5. Difference between desired and actual joint 1 trajectory with *NN* controller.

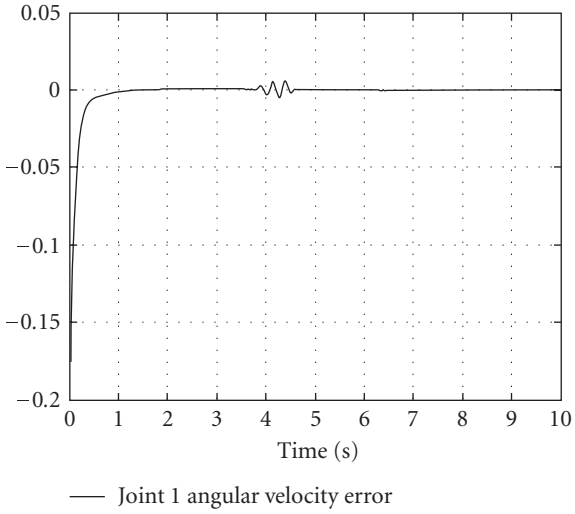

Figure 6.6. Difference between desired and actual joint 1 angular velocity with *NN* controller.

Appendices

A.

Proof of Property 2.3. $M(q)$ is symmetric and positive definite. Thus

$$\overline{M}^T(q) = \left(T^T(q)M(q)T(q)\right)^T = T^T(q)M^T(q)\left(T^T(q)\right)^T = T^T(q)M(q)T(q). \tag{A.1}$$

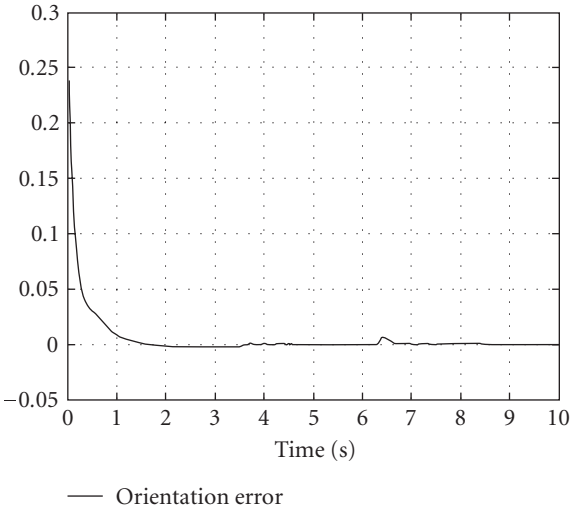

Figure 6.7. Difference between desired and actual end-effector orientation with *NN* controller.

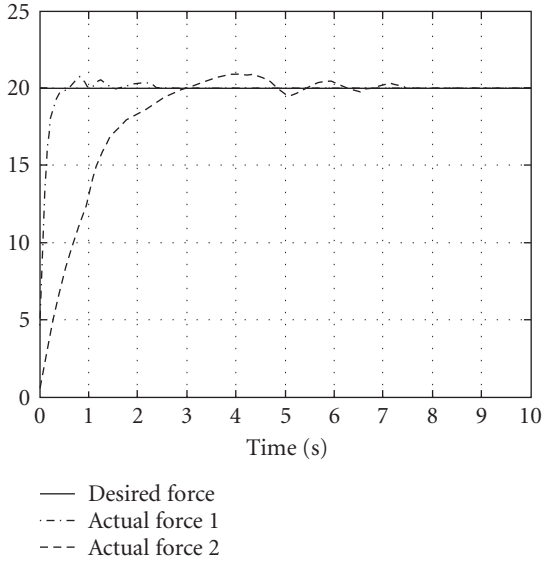

Figure 6.8. Difference between desired and actual contact forces with *NN* controller.

Therefore $\overline{M}(q)$ is symmetric and positive definite. Also

$$\overline{M}(q) = T^T(q)M(q)T(q) \geq \alpha_M \lambda_{\min}\left(T^T(q)T(q)\right)I_n = \overline{\alpha}_M I_n, \tag{A.2}$$

where $\lambda_{\min}(T^T(q)T(q))$ is the minimum eigenvalue of the matrix $T^T(q)T(q)$ and

$$\begin{aligned} \overline{\alpha}_M &= \alpha_M \lambda_{\min}(T^T(q)T(q)), \\ \overline{M}(q) = T^T(q)M(q)T(q) &\leq \beta_M \lambda_{\max}(T^T(q)T(q))I_n = \overline{\beta}_M I_n, \end{aligned} \tag{A.3}$$

where $\lambda_{\max}(T^T(q)T(q))$ is the maximum eigenvalue of the matrix $T^T(q)T(q)$ and

$$\overline{\beta}_M = \beta_M \lambda_{\max}(T^T(q)T(q)). \tag{A.4}$$

Thus the matrix $\overline{M}(q)$ is bounded above and below. □

B.

Proof of Property 2.4.

$$\begin{aligned} \dot{\overline{M}}(q) - 2\overline{C}(q,\dot{q}) &= \dot{T}^T(q)M(q)T(q) + T^T(q)\dot{M}(q)T(q) + T^T(q)M(q)\dot{T}(q) \\ &\quad - 2T^T(q)M\dot{T}(q) - 2T^T(q)C_m(q,\dot{q})T(q) \\ &= T^T(q)(\dot{M}(q) - 2C_m(q,\dot{q}))T(q) \\ &\quad + (\dot{T}^T(q)M(q)T(q) - T^T(q)M(q)\dot{T}(q)). \end{aligned} \tag{B.1}$$

Since

$$(\dot{T}^T(q)M(q)T(q) - T^T(q)M(q)\dot{T}(q))^T = -\dot{T}^T(q)M(q)T(q) + T^T(q)M(q)\dot{T}(q) \tag{B.2}$$

which is a skew-symmetric matrix. From Property 2.1, $\dot{M}(q) - 2C_m(q,\dot{q})$ is skew-symmetric matrix, henceforth $\dot{\overline{M}}(q) - 2\overline{C}(q,\dot{q})$ is also skew-symmetric matrix. □

C.

Proof of Lemma 3.1. For the function $V = (1/2)\tilde{x}^T P(q)\tilde{x} = (1/2)\tilde{x}^T \left[\begin{smallmatrix} K & 0 \\ 0 & \overline{M}(q) \end{smallmatrix}\right]\tilde{x}$, the partial derivative of V with respect to the vector $\tilde{x}$ is given by

$$\frac{\partial V(\tilde{x},t)}{\partial \tilde{x}} = \frac{1}{2}\tilde{x}^T[P(q) + P^T(q)] + \frac{1}{2}\tilde{x}^T \frac{\partial P(q)}{\partial \tilde{x}}\tilde{x} = \tilde{x}^T P(q) + \frac{1}{2}\tilde{x}^T \Delta. \tag{C.1}$$

Since $P(q)$ is symmetric, that is, $P(q) = P^T(q)$, we also have

$$\Delta = \left[\frac{\partial P(q)}{\partial e_1}\tilde{x}, \frac{\partial P(q)}{\partial e_2}\tilde{x}, \ldots, \frac{\partial P(q)}{\partial e_n}\tilde{x}, 0_{2n\times 1}, \ldots, 0_{2n\times 1}\right]. \tag{C.2}$$

The last entries in the matrix Δ are zero vectors $0_{2n\times 1}$ as $\partial P(q)/\partial s_1 = \partial P(q)/\partial s_2 = \cdots = \partial P(q)/\partial s_n = 0_{2n\times 1}$.

The Hamiltonian of the optimization for the system is given by

$$\mathrm{H}\left(\tilde{x}, u, \frac{\partial V(\tilde{x},t)}{\partial \tilde{x}}, t\right) = L(\tilde{x},u) + \frac{\partial V(\tilde{x},t)}{\partial \tilde{x}}\dot{\tilde{x}}. \tag{C.3}$$

The optimal control $u^o(t)$ is followed by the partial derivative of the Hamiltonian with respect to $u(t)$ by setting equal to zero, that is,

$$\frac{\partial}{\partial u}\left[\frac{1}{2}\tilde{x}^T Q\tilde{x}+\frac{1}{2}u^T Ru+\frac{\partial V(\tilde{x},t)}{\partial \tilde{x}}\dot{\tilde{x}}\right]_{u=u^o}=0. \tag{C.4}$$

Using $\dot{\tilde{x}}=A\tilde{x}+Bu$, we get

$$\frac{1}{2}u^{oT}(R+R^T)+\frac{\partial V}{\partial \tilde{x}}B=0. \tag{C.5}$$

Since $R=R^T$, we have $u^{0^T}=-(\partial V/\partial\tilde{x})BR^{-1}$ or $u^o=-R^{-1}B^T(\partial V/\partial\tilde{x})^T$. Using (C.1), we have

$$u^o=-R^{-1}B^TP(q)\tilde{x}. \tag{C.6}$$

Now

$$\begin{aligned}\Delta\dot{\tilde{x}}&=\left[\frac{\partial P(q)}{\partial e_1}\tilde{x},\frac{\partial P(q)}{\partial e_2}\tilde{x},\ldots,\frac{\partial P(q)}{\partial e_n}\tilde{x},0_{2n\times1},\ldots,0_{2n\times1}\right]\\&\quad\times\begin{bmatrix}\dot{e}_1 & \dot{e}_2 & \cdots & \dot{e}_n & \dot{s}_1 & \dot{s}_2 & \cdots & \dot{s}_n\end{bmatrix}^T\\&=\left[\frac{\partial P(q)}{\partial e_1}\dot{e}_1\tilde{x}+\frac{\partial P(q)}{\partial e_2}\dot{e}_2\tilde{x}+\cdots+\frac{\partial P(q)}{\partial e_n}\dot{e}_n\tilde{x}\right]=\dot{P}(q)\tilde{x}\left(\operatorname{sinc}e\frac{\partial P(q)}{\partial t}=0\right).\end{aligned} \tag{C.7}$$

Using $\dot{\tilde{x}}=A\tilde{x}+Bu$ and inserting (C.1), (C.6), and (C.7) in HJB equation, we have

$$\tilde{x}^TPA\tilde{x}-\tilde{x}^TPBR^{-1}B^TP\tilde{x}+\frac{1}{2}\tilde{x}^T\dot{P}(q)\tilde{x}+\frac{1}{2}\tilde{x}^TQ\tilde{x}+\frac{1}{2}(-R^{-1}B^TP\tilde{x})^TR(-R^{-1}B^TP\tilde{x})=0. \tag{C.8}$$

Since $\tilde{x}^TPA\tilde{x}=(1/2)\tilde{x}^T\{PA+A^TP\}\tilde{x}$ and $(1/2)(-\mathbb{R}^{-1}B^TP\tilde{x})^TR(-\mathbb{R}^{-1}B^TP\tilde{x})=(1/2)\tilde{x}^T PBR^{-1}B^TP\tilde{x}$, the equation can be written as

$$\frac{1}{2}\tilde{x}^T(\dot{P}+A^TP+PA+Q-PBR^{-1}B^TP)\tilde{x}=0. \tag{C.9}$$

This leads to the following Riccati equation:

$$\dot{P}+A^TP+PA+Q-PBR^{-1}B^TP=0. \tag{C.10}$$

Also using

$$\dot{P} = \begin{bmatrix} 0 & 0 \\ 0 & \dot{\overline{M}} \end{bmatrix}, \qquad A^T P = \begin{bmatrix} -\Lambda^T & 0 \\ I & -\overline{C}^T \overline{M}^{-1} \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & \overline{M} \end{bmatrix} = \begin{bmatrix} -\Lambda^T K & 0 \\ K & -\overline{C}^T \end{bmatrix},$$
$$PA = \begin{bmatrix} K & 0 \\ 0 & \overline{M} \end{bmatrix} \begin{bmatrix} -\Lambda & I \\ 0 & -\overline{M}^{-1}\overline{C} \end{bmatrix} = \begin{bmatrix} -K\Lambda & K \\ 0 & -\overline{C} \end{bmatrix},$$
$$PBR^{-1}B^T P = \begin{bmatrix} 0 \\ I \end{bmatrix} R^{-1} \begin{bmatrix} 0 & I \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & R^{-1} \end{bmatrix} \tag{C.11}$$

we get the following relations:

$$-\Lambda^T K - K\Lambda + Q_{11} = 0 \quad \text{or} \quad \Lambda^T K + K\Lambda = Q_{11},$$
$$K + Q_{12} = 0, \qquad K + Q_{21} = 0 \quad \text{or} \quad K = -\frac{1}{2}(Q_{12} + Q_{21}),$$
$$\dot{\overline{M}} - \overline{C}^T - \overline{C} + Q_{22} - R^{-1} = 0 \quad \text{or} \quad \left(\left(\dot{\overline{M}} - 2\overline{C}^T\right) - \left(\overline{C} - \overline{C}^T\right)\right) + \left(Q_{22} - R^{-1}\right) = 0. \tag{C.12}$$

Using the robot Property 2.4, we can set $R^{-1} = Q_{22}$. Thus the proof is completed. □

Acknowledgment

This work was financially supported by the Council of Scientific and Industrial Research (CSIR), New Delhi.

References

[1] J. J. Craig and M. H. Raibert, "A systematic method of hybrid position/force control of a manipulator," in *Proceedings of the 3rd IEEE International Computer Software Applications Conference (COMPSAC '79)*, pp. 446–451, Chicago, Ill, USA, November 1979.

[2] O. Khatib and J. Burdick, "Motion and force control of robot manipulators," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1381–1386, San Francisco, Calif, USA, April 1986.

[3] N. H. McClamroch, "Singular systems of differential equations as dynamic models for constrained robot systems," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 21–28, San Francisco, Calif, USA, April 1986.

[4] T. Yoshikawa, "Dynamic hybrid position/force control of robot manipulators description of hand constraints and calculation of joint driving force," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1393–1398, San Francisco, Calif, USA, April 1986.

[5] N. Hogan, "Impedance control: an approach to manipulation—part I: theory," *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, vol. 107, no. 1, pp. 1–7, 1985.

[6] N. Hogan, "Impedance control: an approach to manipulation—part II: implementation," *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, vol. 107, no. 1, pp. 8–16, 1985.

[7] N. Hogan, "Impedance control: an approach to manipulation—part III: applications," *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, vol. 107, no. 1, pp. 17–24, 1985.

[8] C.-Y. Su, T.-P. Leung, and Q. J. Zhou, "Force/motion control of constrained robots using sliding mode," *IEEE Transactions on Automatic Control*, vol. 37, no. 5, pp. 668–672, 1992.
[9] J. Roy and L. L. Whitcomb, "Adaptive force control of position/velocity controlled robots: theory and experiment," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 121–137, 2002.
[10] A. Lanzon and R. J. Richards, "Compliant motion control for non-redundant rigid robotic manipulators," *International Journal of Control*, vol. 73, no. 3, pp. 225–241, 2000.
[11] N. H. McClamroch and D. Wang, "Feedback stabilization and tracking of constrained robots," *IEEE Transactions on Automatic Control*, vol. 33, no. 5, pp. 419–426, 1988.
[12] S.-R. Wang, Z.-Z. Qiao, and P.-C. Tung, "Application of the force control on the working path tracking," *Journal of Marine Science and Technology*, vol. 10, no. 2, pp. 98–103, 2002.
[13] D. Xiao, B. K. Ghosh, N. Xi, and T. J. Tarn, "Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 635–645, 2000.
[14] R. Johansson, "Quadratic optimization of motion coordination and control," *IEEE Transactions on Automatic Control*, vol. 35, no. 11, pp. 1197–1208, 1990.
[15] R. Johansson and M. W. Spong, "Quadratic optimization of impedance control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 616–621, San Diego, Calif, USA, May 1994.
[16] Y. H. Kim, F. L. Lewis, and D. M. Dawson, "Intelligent optimal control of robotic manipulators using neural networks," *Automatica*, vol. 36, no. 9, pp. 1355–1364, 2000.
[17] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis, London, UK, 1999.
[18] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 703–715, 1995.
[19] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 388–399, 1996.
[20] S. Hu, M. H. Ang Jr., and H. Krishnan, "On-line neural network compensator for constrained robot manipulators," in *Proceedings of the 3rd Asian Control Conference*, pp. 1621–1627, Shanghai, China, July 2000.
[21] N. Saadia, Y. Amirat, J. Pontnaut, and A. Ramdane-Cherif, "Neural adaptive force control for compliant robots," in *Proceedings of International Conference on Artificial Neural Networks (ICANN '01)*, pp. 906–913, Vienna, Austria, August 2001.
[22] S. Uran, A. Hace, and K. Jezernik, "Neural network based impedance controller," in *Proceedings of the 5th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD '96)*, pp. 547–552, Budapest, Hungary, June 1996.
[23] Y.-P. Chen and J.-L. Chang, "Sliding-mode force control of manipulators," *Proceedings of the National Science Council, Republic of China, Part A: Physical Science and Engineering*, vol. 23, no. 2, pp. 281–288, 1999.
[24] R. N. Murray, Z. X. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Boca Raton, Fla, USA, 1994.
[25] D. S. Naidu, *Optimal Control Systems*, CRC Press, Boca Raton, Fla, USA, 2003.
[26] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

Vikas Panwar: Department of Mathematics, Chaudhary Devi Lal University, Sirsa, Haryana, India
Email address: vikasdma@yahoo.co.in

N. Sukavanam: Department of Mathematics, Indian Institute of Technology, Roorkee 247667, Uttaranchal, India
Email address: nsukvfma@iitr.ernet.in