

# Stability analysis of robust adaptive hybrid position/force controller for robot manipulators using neural network with uncertainties

H. P. Singh · N. Sukavanam

Received: 12 May 2011 / Accepted: 21 May 2012 / Published online: 15 June 2012  
© Springer-Verlag London Limited 2012

**Abstract** The aim of this paper is to design a robust adaptive neural network-based hybrid position/force control scheme for robot manipulators in the presence of model uncertainties and external disturbance. The feedforward neural network employed to learn a highly nonlinear function requires no preliminary learning. The control purposes are to achieve the stability in the sense of Lyapunov for desired interaction force between the end-effector and the environment and to regulate robot tip position in cartesian space. An adaptive compensator is also developed to eliminate the effect of disturbance term of neural network approximation error and external disturbance or unmodeled dynamics etc. A key feature of this compensator is that the prior information of the disturbance bound is not required. Finally, a comparative simulation study with a model-based robust control scheme for a two-link robot manipulator is presented.

**Keywords** Robust hybrid position/force control · Lyapunov stability · Interaction force · Feedforward neural network · Adaptive compensator · Uncertainty

## 1 Introduction

The dynamics of robot manipulators is governed by highly nonlinear coupled, time-varying system with many uncertainties such as load variation, friction, and external disturbances etc. Due to these uncertainties system error becomes large, when a robot manipulator operates at high

speed. It is a challenging problem in control field to find an effective control scheme to achieve accurate tracking of the desired motion. The control design problem becomes more sophisticated when the desired motion of a robot manipulator is defined at the end-effector frame level, referred as operational space.

In various industrial application of robot manipulators such as contour following and assembly related tasks, the manipulator end-effector is required to make contact with the environment. During the execution of such tasks, the motion of the end-effector is constrained by the environment. For the successful execution of such type of tasks robot positions as well as contact forces should be controlled accurately. Many researchers have investigated this problem in recent years. As a result, two approaches to achieving compliant motion have emerged. The first approach, usually called hybrid position/force control, is based on the observation that when the robot end-effector is in contact with the environment, the Cartesian space of the end-effector coordinates may be naturally decomposed into a position subspace and a force subspace. The objective of hybrid position/force controller is to track a position (and orientation) trajectory in the position subspace and a force (and moment) trajectory in the force subspace. The second approach to compliant motion for robots is called impedance control and proposes that the control objective should not be the tracking of position/force trajectories, but rather should involve the regulation of the mechanical impedance of the robot end-effector, which relates position and force. Raibert and Craig [1] developed a hybrid scheme that decomposes a task space into two orthogonal subspaces: position and force. Lozano and Brogliato [2] proposed an adaptive force/position control scheme, based on a particular decomposition of the robot jacobian and environment stiffness matrices. By extending the method of Raibert and Craig [1], Yoshikawa and Sudou [3]

---

H. P. Singh (✉) · N. Sukavanam  
Department of Mathematics, Indian Institute of Technology  
Roorkee, Roorkee 247667, Uttarakhand, India  
e-mail: harendramaths@gmail.com

proposed dynamic hybrid position/force control scheme, which takes into consideration the manipulator dynamics and the constraints on the end-effector specified by the given task. Kwan [4] proposed a new robust adaptive control scheme for simultaneous force/motion control of constrained rigid robots including motor dynamics. Xiao et al. [5] utilized the force/torque and vision sensors simultaneously to propose a hybrid position/force controller such that the end-effector moved along a path on an unknown surface with the aid of a single camera assumed to be uncelebrated with respect to the robot coordinates. Kouya et al. [6] presented a general study of an adaptive force/position control using the (SFB) strict-feedback backstepping technique, based on passivity and applied to a robotic system. Roy and Whitcomb [7] presented an adaptive force control algorithm with low-level position/velocity controllers for robot arms in contact with surfaces of unknown linear compliance. Cheah et al. [8] proposed a motion and force tracking controller for robots with uncertain kinematics and dynamics. Adaptive Jacobian motion/force controller and Lyapunov functions were also presented for the stability analysis of the robot control systems. Kouya [9] presented a hybrid force-position control scheme for uncertain robot manipulator interacting with its environment. Filaretov and Zuev [10] proposed a new synthesis method of force/position control systems of robot manipulators. The control systems synthesized on the basis of this method without using force/moment sensors and other additional devices provide simultaneously accurate control of both the position of robot's end-effector and the force exerted by end-effector on some object (or environment). Wang and Xie [11] studied the force/position tracking control problem of a free-flying manipulator interacting with an uncertain compliant surface. An adaptive Jacobian controller is designed to deal with uncertainties arising from the free-flyer's kinematics, dynamics and surface stiffness and position. Huang and Enomoto [12] proposed a hybrid position, posture, force, and moment control for a robot manipulator for the surface contact work by expansion of the conventional hybrid position and force control by separating the posture physical vector and the moment physical vector from the position vector and the force vector.

During the last decade, neural network technology has gained popularity among control community due to their versatile features such as parallel distributed structure and learning capabilities. The basic idea behind neural network-based control is to learn unknown nonlinear dynamics and compensate for uncertainties existing in the dynamic model. Many NN-based control schemes are available for control of robot manipulators [13–15], in which stable tracking control algorithms have been derived while approximating the unknown dynamics of the manipulators with neural networks trained online without requiring any preliminary

offline training. Panwar and Sukavanam [16] designed an optimal hybrid motion and force control scheme for a constrained robotic manipulator with unknown dynamics. The optimal control law is derived using Hamilton Jacobi Bellman (HJB) optimization. An on-line adaptive neural network controller, using a two-layer feedforward neural network, was proposed to compensate for the dynamic model of the constrained robot. Karayiannidis et al. [17] proposed a novel neuro-adaptive controller for the force and position trajectory tracking of a robot in compliant contact with a flat surface under non-parametric uncertainties existing in the dynamic and contact model. Ferguene and Toumi [18] designed a dynamic external force feedback loop control scheme using a neural network compensator for uncertainties. Bechlioulis et al. [19] developed a neuro-adaptive controller that achieves force/position tracking with prescribed performance as well as contact maintenance in the presence of uncertainties in the force-deformation model and bounded piecewise continuous disturbances acting at the joints.

In this paper, our aim is to design a robust adaptive neural network (NN)-based hybrid position/force control scheme for robot manipulators in the presence of model uncertainties and external disturbance. The proposed control scheme has the following salient features: (1) the exact knowledge of the robot dynamics, linear-parametrization assumption, or lengthy and tedious preliminary analysis to determine a regression matrix is not required (2) the feedforward neural network is employed to learn a highly nonlinear function, which requires no preliminary learning (3) an adaptive compensator is used to eliminate the effect of disturbance term of neural network approximation error and external disturbance or unmodeled dynamics etc. (4) the NN controller achieves the stability in the sense of Lyapunov for desired interaction force between the end-effector and the environment as well as regulate robot tip position in cartesian space.

The remainder of this paper is organized as follows. In Sect. 2, robot dynamical model and its decomposition are presented. A review of feedforward neural network is presented in Sect. 3. In Sect. 4, model-based and feedforward neural network-based hybrid position/force controllers are designed. A comparative simulation study with model-based robust control scheme for a two-link robot manipulator is presented in Sect. 5. Final conclusion is given in Sect. 6.

## 2 Robot manipulator model

### 2.1 Robot kinematic and dynamic model

Euler–Lagrange formulation [13] of the revolute rigid robot dynamics with friction term, external disturbance, and environment contact is given by

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F_e(\dot{q}) + T_d = \tau - \tau_e \quad (1)$$

where  $M(q) \in R^{n \times n}$  represents the inertia matrix,  $V_m(q, \dot{q}) \in R^{n \times n}$  represents the centripetal-coriolis matrix,  $G(q) \in R^n$  represents the gravity effects,  $F(\dot{q}) \in R^n$  represents the friction effects,  $T_d \in R^n$  is a vector of unknown but bounded disturbance,  $\tau \in R^n$  represents the torque input vector, and  $\tau_e \in R^n$  the interaction torque due to contact with the environment. The robot dynamics given in Eq. (1) has the following useful properties.

**Property 1** The inertia matrix  $M(q)$  is symmetric, positive definite, and satisfies the following inequalities

$$m_1 \|\xi\|^2 \leq \xi^T M(q) \xi \leq m_2 \|\xi\|^2 \quad \forall \xi \in R^n \quad (2)$$

where  $m_1, m_2$  are positive constants, and  $\|\cdot\|$  denotes the standard Euclidean norm.

**Property 2** The inertia and centripetal-coriolis matrices satisfy the following skew-symmetric relationship

$$\xi^T \left( \frac{1}{2} \dot{M}(q) - V_m(q, \dot{q}) \right) \xi = 0 \quad \forall \xi \in R^n \quad (3)$$

where  $\dot{M}(q)$  denotes the time derivatives of the inertia matrix.

**Property 3** The left-hand side of Eq. (1) can be linearly parameterized as shown below

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F_e(\dot{q}) = W(q, \dot{q}, \ddot{q})\theta \quad (4)$$

where  $\theta \in R^r$  contains the constant system parameters and the regression matrix  $W(\cdot) \in R^{n \times r}$  contains known functions dependent on the signals  $q(t), \dot{q}(t)$  and  $\ddot{q}(t)$ .

The interaction force  $\bar{F} \in R^m$  at the end-effector is related to the interaction torque  $\tau_e \in R^n$  in the joint space through

$$\tau_e = J^T(q) \bar{F} \quad (5)$$

where  $J(q)$  is the  $m \times n$  ( $m \leq n$ ) Jacobian matrix. Its elements are bounded for a revolute joint robot. The end-effector position and orientation in cartesian space is related to the joint space by the robot kinematics equation

$$x = f(q) \quad (6)$$

where  $x \in R^m$  is the position and orientation vector, and  $f(q)$  is a kinematic transformation of the robot manipulator. Then

$$\dot{x} = J(q)\dot{q} \quad (7)$$

The interaction force  $\bar{F} \in R^m$  in task space is proportional to environmental deformation  $(x - x_e)$  through

$$\bar{F} = \bar{K}(x - x_e) \quad (8)$$

where the stiffness matrix  $\bar{K} \in R^{m \times m}$  is assumed to be constant and  $x_e \in R^m$  is the coordinate of the contact. It is

also assumed that, by rearranging the terms in  $\bar{F}$ , Eq. (8) can be written as

$$\bar{F} = \begin{bmatrix} F \\ F' \end{bmatrix} = \begin{bmatrix} K \\ K' \end{bmatrix} (x - x_e) \quad (9)$$

where  $F \in R^p$ ,  $F' \in R^{m-p}$ ,  $K \in R^{p \times m}$  ( $p \leq m$ ) and is of fullrank and  $K'$  depends linearly on the rows of  $K$ .  $p$  is the dimension of the subspace in the task space ( $R^m$ ) where the force are to be controlled. Since there are only  $n$  ( $n \leq 6$ ) control inputs in the system, it is not possible to control force (six dimensions) and position (six dimensions) of the robot in all dimensions. The best we can do is to control only a small subspace of the whole space. To be more specific, we can control  $p$  forces,  $(m - p)$  cartesian positions and  $(n - m)$  redundant joint velocities. The total dimension of the subspace is  $p + (m - p) + (n - m) = n$ .

## 2.2 Decomposition of robot dynamics

In this subsection, a scheme is described that can decompose the robot dynamics into force, position, and redundant joint subspaces. The following identities will be used in the decomposition process

$$I = J^+ J + J^- \quad (10)$$

$$I = K^+ K + K^- \quad (11)$$

where

$$J^+ = J^T [J J^T]^{-1} \quad (12)$$

$$K^+ = K^T [K K^T]^{-1} \quad (13)$$

$$J^- = I - J^T [J J^T]^{-1} J \quad (14)$$

$$K^- = I - K^T [K K^T]^{-1} K \quad (15)$$

where  $J^+$  and  $K^+$  are Penrose pseudo inverse and  $J^-$  and  $K^-$  are projectors, that is, idempotent matrices.

Using Eqs. (10) and (7),  $\dot{q}$  can be decomposed as

$$\dot{q} = J^+ J \dot{q} + J^- \dot{q} = J^+ \dot{x} + J^- \dot{q} \quad (16)$$

Using Eqs. (9) and (11),  $\dot{x}$  can be decomposed as

$$\dot{x} = K^+ K \dot{x} + K^- \dot{x} = K^+ \dot{F} + K^- \dot{x} \quad (17)$$

Taking the first derivative of Eq. (7), we get

$$\ddot{x} = J \ddot{q} + \dot{J} \dot{q} \quad (18)$$

Using Eq. (10) together with Eq. (18),  $\ddot{q}$  is decomposed as follows

$$\ddot{q} = J^+ J \ddot{q} + J^- \ddot{q} = J^+ (\ddot{x} - \dot{J} \dot{q}) + J^- \ddot{q} \quad (19)$$

Differentiation of Eq. (17), we get

$$\ddot{x} = K^+ K \ddot{x} + K^- \ddot{x} = K^+ \ddot{F} + K^- \ddot{x} \quad (20)$$

Substituting Eqs. (16), (17) and (19) into Eq. (1), we get

$$M(J^+[K^+\ddot{F} + K^-\ddot{x} - \dot{J}\dot{q}] + J^-\ddot{q}) + V_m(J^+[K^+\dot{F} + K^-\dot{x}] + J^-\dot{q}) + G + F_e + T_d = \tau - \tau_e \quad (21)$$

This is exactly equivalent to Eq. (1), and there is a clear decomposition of force ( $F$ ), position ( $K^+x$ ), and redundant joint  $J^-q$ .

### 3 Feedforward neural network

Mathematically, a two-layer feed-forward neural network (Fig. 1) with  $n$  input units,  $m$  output units, and  $N$  units in the hidden layer is given as [13]

$$z_i = \sum_{j=1}^N \left[ w_{ij} \sigma \left( \sum_{k=1}^n v_{jk} y_k + \theta_{vj} \right) + \theta_{wi} \right] \quad i = 1, 2, \dots, m \quad (22)$$

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  $\theta_{vj}$ ,  $\theta_{wi}$ . There are many classes of activation functions, for example, sigmoid, hyperbolic tangent, and Gaussian. The sigmoid activation function used in this work is given by

$$\sigma(y) = \frac{1}{1 + e^{-y}}$$

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

$$z = W^T \sigma(V^T y)$$

with the vector of activation functions defined by  $\sigma(p) = [\sigma(p_1) \dots \sigma(p_n)]^T$  for a vector  $p \in R^n$ . The bias weights are included as the first column of the weight matrices. To accommodate bias weights, the vectors  $y$  and  $\sigma(\cdot)$  need to be augmented by replacing 1 as their first element, for example,  $y = [1 \ y_1 \ y_2 \ \dots \ y_n]^T$ . According to the neural network function approximation property [13], given a continuous function  $h$  of  $n$  variables on a compact set  $\Omega \subset R^n$ , there exists  $N$  hidden-layer neurons, weights matrices  $W$  and  $V$  such that

$$h(y) = W^T \sigma(V^T y) + \epsilon(y) \quad (23)$$

where  $\epsilon(y)$  denotes NN approximation error satisfying  $\|\epsilon(y)\| < \epsilon_N$  for some  $\epsilon_N > 0$ . Then, an estimate of  $h(y)$  can be given by

$$\hat{h}(y) = \hat{W}^T \sigma(\hat{V}^T y)$$

where  $\hat{W}$  and  $\hat{V}$  are the estimations of  $W$  and  $V$ , respectively, provided by some on-line weight-tuning algorithms.

### 4 Controller design and stability analysis

In this section, model-based and neural network-based control schemes are developed to achieve the stability in the sense of Lyapunov for desired interaction force between the end-effector and the environment as well as regulate robot tip position in cartesian space. Let us define

$$r = J^+[K^+(\dot{F} - v_f) + K^-(\dot{x} - v_x)] + J^-\dot{\tilde{q}} \quad (24)$$

where

$$v_f = \dot{F}_d - \lambda \tilde{F} \quad v_x = \dot{x}_d - \lambda \tilde{x} \quad (25)$$

$$\tilde{F} = F - F_d \quad \tilde{x} = x - x_d \quad \tilde{q} = q - q_d \quad (26)$$

Differentiating Eq. (24) and using Eqs. (25) and (26), the robot dynamics given by Eq. (21) can be written in terms of  $r$  as

$$M\dot{r} + V_m r = \tau - \tau_e - h(y) - T_d \quad (27)$$

where  $h(y) = M[J^+(K^+v_f + K^-v_x - \dot{J}\dot{q}) + J^-\ddot{q}_d - z] + V_m[J^+(K^+v_f + K^-v_x) + J^-\dot{\tilde{q}}] + G + F_e$  is termed as robot nonlinear function with  $y = [r^T \ \bar{F}^T \ q^T \ \dot{q}^T]^T$ . Using the property 3,  $h(y)$  can be written as

$$h(y) = Y(r, \bar{F}, q, \dot{q})\theta \quad (28)$$

[2], where  $Y(r, \bar{F}, q, \dot{q}) \in R^{n \times r}$  denotes the regression matrix and  $\theta \in R^r$  contains the constant system parameters (e.g., mass, inertia, friction coefficients). Also  $z = \frac{dJ^+}{dt} [K^+(\dot{F} - v_f) + K^-(\dot{x} - v_x)] + \frac{dJ^-}{dt} \dot{\tilde{q}}$ . During the controller development, we will make the assumption that the Jacobian matrix  $J(q)$  has full rank and the desired trajectories are bounded.

#### 4.1 Model-based robust control scheme with robust compensator

Based on the error dynamics system development in Eq. (27) and the subsequent stability analysis, we design the following robust controller  $\tau$  as

$$\tau = \tau_e + Y\hat{\theta} - K_d r + v \quad (29)$$

where  $\hat{\theta} \in R^n$  denotes the constant best guess estimation of the parameter vector  $\theta$  defined in Eq. (28),  $K_d = K_d^T$  is a positive definite gain matrix and  $v$  is a robust compensator term defined as

$$v = -\frac{r\rho^2}{\|r\|\rho + \delta}, \quad (30)$$

where  $\rho$  is designed to satisfy

$$\rho \geq \|Y\tilde{\theta}\| + \|T_d\|, \quad (31)$$

$\dot{\delta} = -\gamma\delta$  with  $\delta(0) > 0$ . With the controller in Eq. (29), the error dynamics system for  $r$  can be written in the following form

$$M\dot{r} = -V_m r - K_d r + Y\tilde{\theta} - T_d + v. \quad (32)$$

For stability analysis, we have the following theorem.

**Theorem 1** *If control input torque  $\tau$  and robust compensator  $v$  are designed as Eqs. (29) and (30), respectively, then the force error ( $\tilde{F}$ ), position error ( $K^-\tilde{x}$ ) and redundant joint velocity error ( $J^-\dot{\tilde{q}}$ ) converge to zero as  $t \rightarrow \infty$ .*

*Proof 1* Consider the following Lyapunov function candidate

$$L = \frac{1}{2}r^T M r + \frac{\delta}{\gamma} \quad (33)$$

The time derivative of the Lyapunov function gives

$$\dot{L} = \frac{1}{2}r^T \dot{M} r + r^T M \dot{r} + \frac{\dot{\delta}}{\gamma} \quad (34)$$

Using Eq. (32), we get

$$\dot{L} = \frac{1}{2}r^T \dot{M} r + r^T (-V_m r - K_d r + Y\tilde{\theta} - T_d + v) - \delta \quad (35)$$

Now using Eqs. (30) and (31), we get

$$\dot{L} \leq \frac{1}{2}r^T (\dot{M} - 2V_m) r - r^T K_d r + \|r\|\rho - \frac{\|r\|^2 \rho^2}{\|r\|\rho + \delta} - \delta \quad (36)$$

From property 2, we have

$$\dot{L} \leq -r^T K_d r + \frac{\|r\|\rho^2}{\|r\|\rho + \delta} - \delta \quad (37)$$

$$\dot{L} \leq -r^T K_d r \quad (38)$$

Since  $L > 0$  and  $\dot{L} \leq 0$ , this shows stability in the sense of Lyapunov, so that  $r$  is bounded [21]. Now from Eq. (24), using Eq. (9), we get [2]

$$r = J^+[K^+(K\dot{x} - v_f) + K^-(\dot{x} - v_x)] + J^-\dot{\tilde{q}}$$

Using Eq. (11), we get

$$\begin{aligned} r &= J^+[\dot{x} - K^+v_f - K^-v_x] + J^-\dot{\tilde{q}} \\ r &= J^+\dot{x} - J^+(K^+v_f + K^-v_x) + J^-\dot{\tilde{q}} - J^-\dot{\tilde{q}} \end{aligned}$$

Now using Eq. (16), we get

$$r = \dot{q} - J^+(K^+v_f + K^-v_x) - J^-\dot{\tilde{q}} \quad (39)$$

Since we consider revolute joints only,  $J^+$ ,  $J^-$ ,  $K^+$ ,  $K^-$ ,  $x$ ,  $F$  are all bounded due to physical constraints. Therefore, from Eq. (25)  $v_f$  and  $v_x$  are also bounded. Now,  $q_d$  is also bounded. Thus, we see that  $\dot{q}$  is bounded. Differentiating  $r$  in Eq. (39) and using Property 1, it can be similarly argued that, for a revolute joint robot,  $\dot{r}$  is also bounded. Hence,  $r \rightarrow 0$  as  $t \rightarrow \infty$ .

Now,  $K$  and  $J$  are bounded and are of full rank. Multiplying Eq. (24) by  $KJ$  we get

$$KJr = \dot{F} - v_f \rightarrow 0 \quad (40)$$

since  $r \rightarrow 0$ . Now multiplying Eq. (24) by  $J$ , we get

$$Jr = K^-(\dot{x} - v_x) \rightarrow 0 \quad (41)$$

Now using Eqs. (24), (40) and (41), we get

$$J^-\dot{\tilde{q}} \rightarrow 0 \quad (42)$$

Now from Eqs. (25), (26), (41) and (42), we have  $\dot{\tilde{F}} + \lambda\tilde{F} \rightarrow 0$  and  $K^-\dot{\tilde{x}} + \lambda K^-\tilde{x} \rightarrow 0$ . Now  $\tilde{F}$  and  $K^-\tilde{x}$  can be considered as the outputs of filters of the form  $1/(s + \lambda)$  with inputs of  $\dot{\tilde{F}} + \lambda\tilde{F} \rightarrow 0$  and  $K^-\dot{\tilde{x}} + \lambda K^-\tilde{x} \rightarrow 0$ , respectively. Hence, we see that the force error  $\tilde{F} \rightarrow 0$ , position error  $K^-\tilde{x} \rightarrow 0$  and redundant joint velocity error  $J^-\dot{\tilde{q}} \rightarrow 0$ .

#### 4.2 Robust adaptive neural network-based control scheme with adaptive compensator

Based on the error dynamics system development in Eq. (27) and the subsequent stability analysis, we design the following robust adaptive neural network-based controller  $\tau$  as

$$\tau = \tau_e + \hat{h}(y) - K_d r + v \quad (43)$$

where  $\hat{h}(y)$  is an estimate of robot nonlinear function. With the controller in Eq. (43), the closed loop error dynamics becomes

$$M\dot{r} = -V_m r - \tilde{h}(y) - K_d r - T_d + v \quad (44)$$

where  $\tilde{h}(y) = h(y) - \hat{h}(y)$  is the functional estimation error. The functional estimation  $\hat{h}(y)$  with a feedforward neural network (FFNN) may be given as

$$\hat{h}(y) = \hat{W}^T \sigma(\hat{V}^T y) \quad (45)$$

Using this FFNN functional approximation, the closed loop error dynamics becomes

$$\begin{aligned} M\dot{r} &= -V_m r - K_d r - W^T \sigma(V^T y) - \epsilon(y) + \hat{W}^T \sigma(\hat{V}^T y) \\ &\quad - T_d + v \end{aligned} \quad (46)$$



For notational convenience, define the matrix of all NN weights as  $Z \equiv \text{diag}\{W, V\}$  and the weight estimation errors as  $\tilde{W} = W - \hat{W}$ ,  $\tilde{V} = V - \hat{V}$  and  $\tilde{Z} = Z - \hat{Z}$ . The ideal NN weights are bounded so that  $\|Z\|_F \leq Z_M$  with known  $Z_M$  and  $Z_F^2 = \text{tr}(Z^T Z)$ .

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 Eq. (46), we get

$$\begin{aligned} M\dot{r} = & -V_m r - K_d r - W^T (\sigma(V^T y) - \sigma(\hat{V}^T y)) - \epsilon(y) \\ & - \tilde{W}^T \sigma(\hat{V}^T y) - T_d + v \end{aligned} \quad (47)$$

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

$$\sigma(V^T y) = \sigma(\hat{V}^T y) + \sigma'(\hat{V}^T y) \tilde{V}^T y + O(\tilde{V}^T y)^2 \quad (48)$$

with  $\sigma'(\hat{z}) = \frac{d\sigma(z)}{dz}|_{z=\hat{z}}$  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 Eq. (47), we get

$$M\dot{r} = -V_m r - K_d r - \hat{W}^T \hat{\sigma}' \tilde{V}^T y - \tilde{W}^T \hat{\sigma} - w_1 + v \quad (49)$$

where the disturbance terms are

$$w_1 = \tilde{W}^T \hat{\sigma}' \tilde{V}^T y + W^T O(\tilde{V}^T y)^2 + \epsilon(y) + T_d \quad (50)$$

Finally, adding and subtracting  $\tilde{W}^T \hat{\sigma}' \tilde{V}^T y$  in Eq. (49), we get

$$\begin{aligned} M\dot{r} = & -V_m r - K_d r - \hat{W}^T \hat{\sigma}' \tilde{V}^T y - \tilde{W}^T (\hat{\sigma} - \hat{\sigma}' \tilde{V}^T y) \\ & - w_1 - \tilde{W}^T \hat{\sigma}' \tilde{V}^T y + v \end{aligned} \quad (51)$$

$$\begin{aligned} M\dot{r} = & -V_m r - K_d r - \hat{W}^T \hat{\sigma}' \tilde{V}^T y - \tilde{W}^T (\hat{\sigma} - \hat{\sigma}' \tilde{V}^T y) \\ & - w + v \end{aligned} \quad (52)$$

where the modified disturbance terms are

$$w(t) = \tilde{W}^T \hat{\sigma}' V^T y + W^T O(\tilde{V}^T y)^2 + \epsilon(y) + T_d \quad (53)$$

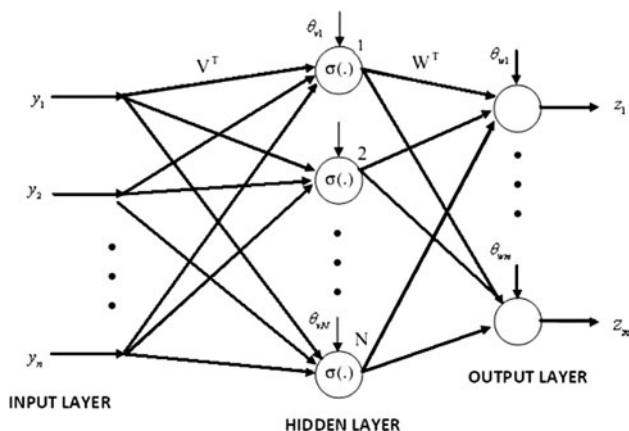


Fig. 1 Feedforward neural network

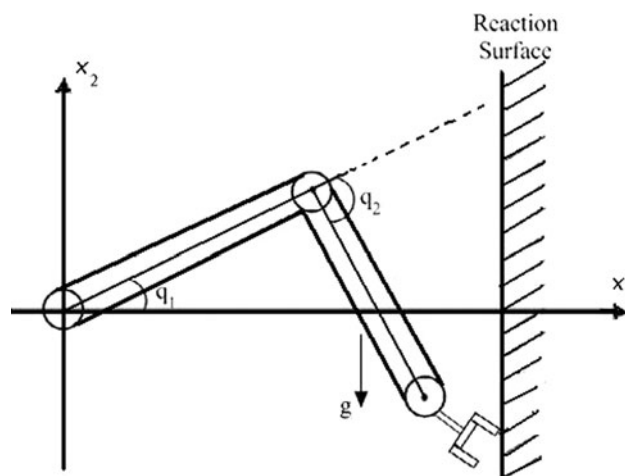


Fig. 2 Two-link robot manipulator contacting a surface

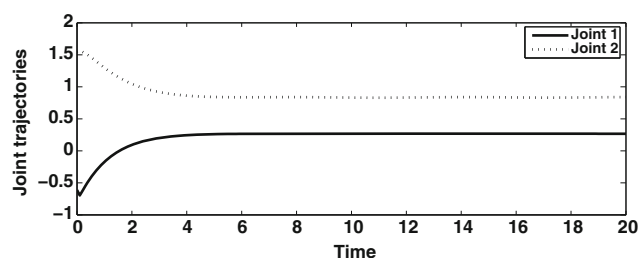


Fig. 3 Joint trajectories without FFNN

with

$$\|w(t)\| \leq \rho \quad (54)$$

where the positive scalar function  $\rho(t)$  is defined as [13]

$$\rho = C_o + C_1 \|\tilde{Z}\|_F + C_2 \|\tilde{Z}\|_F \|r\|. \quad (55)$$

In the error system neural network approximation error  $\epsilon(y)$ , external disturbance or unmodeled dynamics  $T_d$  and the higher-order terms in the Taylor series expansion for  $\tilde{\sigma}$  all have exactly the same influence as disturbances. To eliminate the effect of these disturbances, we choose the stabilizing adaptive compensator as follows [20]

$$v = -\frac{\hat{\rho}^2 r}{\hat{\rho} \|r\| + \delta} \quad (56)$$

where  $\hat{\rho}$  is the estimation of the positive bounding function  $\rho(\cdot)$  designed as

$$\hat{\rho} = Q^T \hat{\phi}$$

where

$$Q^T = [1 \ \|\tilde{Z}\|_F \ \|\tilde{Z}\|_F \|r\|] \quad (57)$$

and  $\hat{\phi}$  is the estimation of  $\phi = [C_0 \ C_1 \ C_2]^T$  with largest possible positive-bounding constants  $C_i$  for  $i = 0, 1, 2$ , that

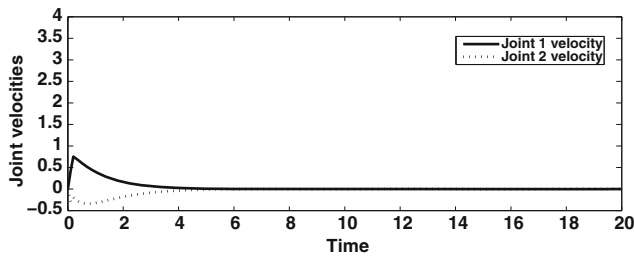


Fig. 4 Joint velocities without FFNN

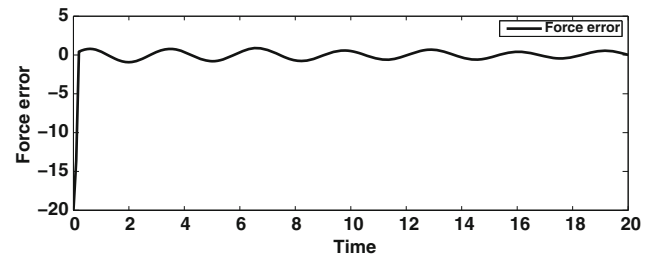


Fig. 7 Force tracking error without FFNN

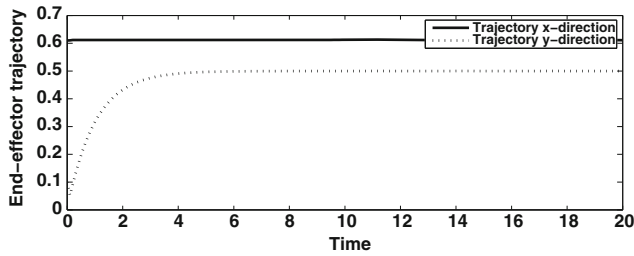


Fig. 5 End-effector trajectories without FFNN

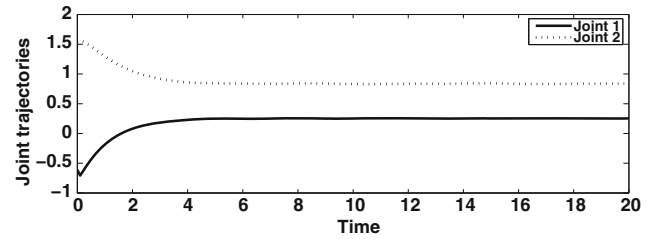


Fig. 8 Joint trajectories without FFNN

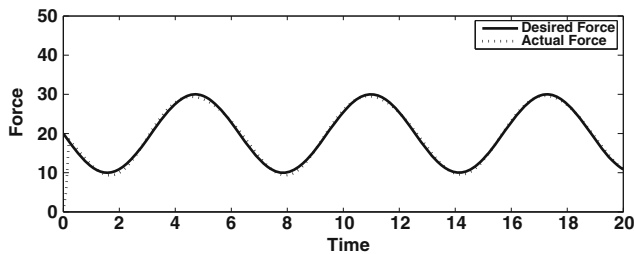


Fig. 6 Force tracking without FFNN

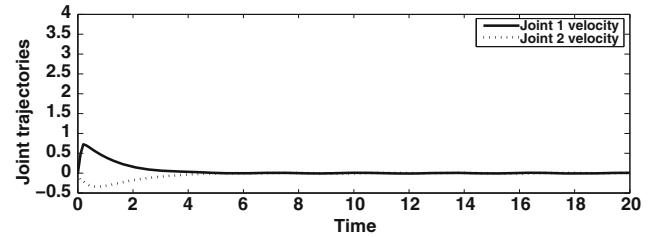


Fig. 9 Joint velocities without FFNN

are based on the desired trajectory, external disturbances, NN approximation error, and so on.

For stability analysis we have the following theorem.

**Theorem 2** If control input torque  $\tau$  and adaptive compensator  $v$  are designed as Eqs. (43) and (56), respectively, and adaptive laws are selected as

$$\begin{aligned}\dot{\hat{W}} &= -F_w \hat{\sigma} r^T + F_w \hat{\sigma}' \hat{V}^T y r^T, \quad \dot{\hat{V}} = -G_v y (\hat{\sigma}' \hat{W} y)^T, \\ \dot{\hat{\phi}} &= \Gamma_\phi Q \|r\| \end{aligned} \quad (58)$$

with some scalar positive definite symmetric matrices  $F_w$ ,  $G_v$ , and  $\Gamma_\phi$ , then the force error ( $\tilde{F}$ ), position error ( $K^{-1}\tilde{x}$ ) and redundant joint velocity error ( $J^{-1}\dot{\tilde{q}}$ ) converge to zero as  $t \rightarrow \infty$ .

**Proof 2** Consider the following Lyapunov function candidate

$$\begin{aligned}L &= \frac{1}{2} r^T M r + \frac{1}{2} \text{tr}(\tilde{W}^T F_w^{-1} \tilde{W}) + \frac{1}{2} \text{tr}(\tilde{V}^T G_v^{-1} \tilde{V}) \\ &\quad + \frac{1}{2} \text{tr}(\tilde{\phi}^T \Gamma_\phi^{-1} \tilde{\phi}) + \frac{\delta}{\gamma} \end{aligned} \quad (59)$$

The time derivative of the Lyapunov function gives

$$\begin{aligned}\dot{L} &= \frac{1}{2} r^T \dot{M} r + r^T M \dot{r} + \text{tr}(\tilde{W}^T F_w^{-1} \dot{\tilde{W}}) + \text{tr}(\tilde{V}^T G_v^{-1} \dot{\tilde{V}}) \\ &\quad + \text{tr}(\tilde{\phi}^T \Gamma_\phi^{-1} \dot{\tilde{\phi}}) + \frac{\dot{\delta}}{\gamma} \end{aligned} \quad (60)$$

Using Eq. (52), we get

$$\begin{aligned}\dot{L} &= \frac{1}{2} r^T \dot{M} r + r^T (-V_m r - K_d r - \hat{W}^T \hat{\sigma}' \tilde{V}^T y \\ &\quad - \tilde{W}^T (\hat{\sigma} - \hat{\sigma}' \hat{V}^T y) - w + v) + \text{tr}(\tilde{W}^T F_w^{-1} \dot{\tilde{W}}) \\ &\quad + \text{tr}(\tilde{V}^T G_v^{-1} \dot{\tilde{V}}) + \text{tr}(\tilde{\phi}^T \Gamma_\phi^{-1} \dot{\tilde{\phi}}) - \delta \end{aligned} \quad (61)$$

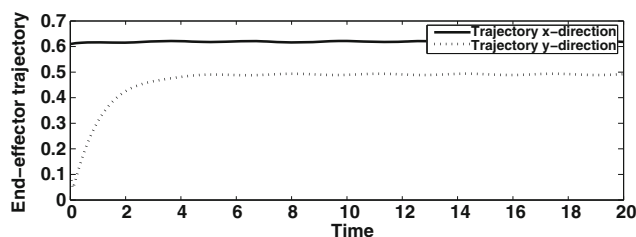


Fig. 10 End-effector trajectories without FFNN

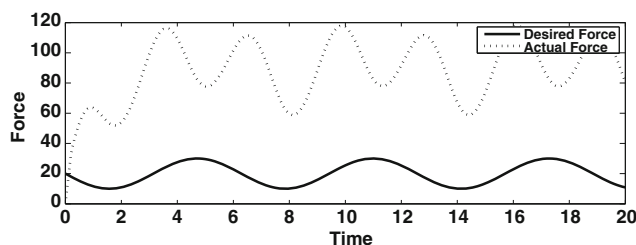


Fig. 11 Force tracking without FFNN

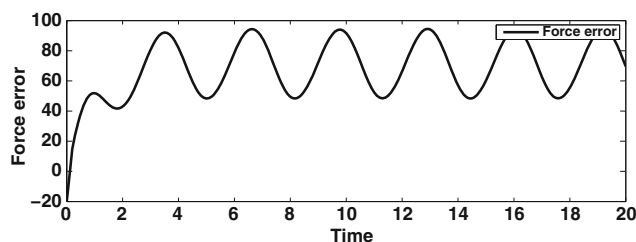


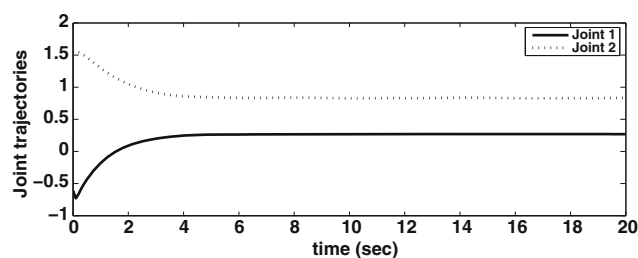
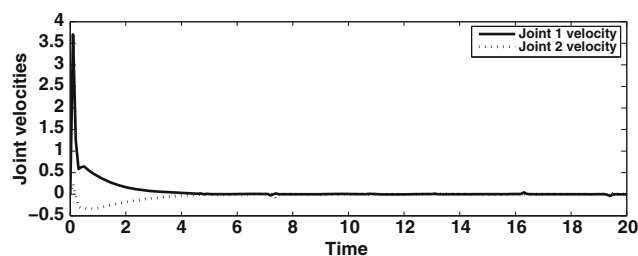
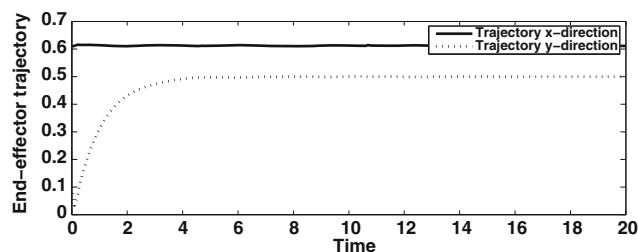
Fig. 12 Force Tracking error without FFNN

$$\begin{aligned} \dot{L} = & \frac{1}{2} r^T (\dot{M} - 2V_m) r - r^T K_d r - r^T \tilde{W}^T \hat{\sigma}' \tilde{V}^T y \\ & - r^T \tilde{W}^T (\hat{\sigma} - \hat{\sigma}' \tilde{V}^T y) + tr(\tilde{W}^T F_w^{-1} \dot{\tilde{W}}) + tr(\tilde{V}^T G_v^{-1} \dot{\tilde{V}}) \\ & + tr(\tilde{\phi}^T \Gamma_\phi^{-1} \dot{\tilde{\phi}}) - r^T w + r^T v - \delta \end{aligned} \quad (62)$$

Now using property 2 together with  $\tilde{\phi} = \phi - \hat{\phi}$ ,  $\dot{\tilde{W}} = -\dot{\tilde{W}}$ ,  $\dot{\tilde{V}} = -\dot{\tilde{V}}$ ,  $\dot{\tilde{\phi}} = -\dot{\tilde{\phi}}$  and adaptive learning laws in Eq. (58), we have

$$\begin{aligned} \dot{L} \leq & -r^T K_d r - r^T \tilde{W}^T \hat{\sigma}' \tilde{V}^T y - r^T \tilde{W}^T \hat{\sigma} + r^T \tilde{W}^T \hat{a}' \tilde{V}^T y \\ & + tr(\tilde{W}^T \hat{\sigma} r^T - \tilde{W}^T \hat{a}' \tilde{V}^T y r^T) + tr(\tilde{V}^T y r^T \tilde{W}^T \hat{\sigma}'^T) \\ & + tr(-\tilde{\phi}^T \Gamma_\phi^{-1} \tilde{\phi} Q \|r\|) + \|r\| \rho + \|r\| \hat{\rho} - \|r\| \hat{\rho} \\ & + r^T v - \delta \end{aligned} \quad (63)$$

$$\begin{aligned} \dot{L} \leq & -r^T K_d r + tr(\tilde{W}^T \hat{\sigma} r^T - \tilde{W}^T \hat{a}' \tilde{V}^T y r^T - \tilde{W}^T \hat{\sigma} r^T \\ & + \tilde{W}^T \hat{a}' \tilde{V}^T y r^T) + tr(\tilde{V}^T y r^T \tilde{W}^T \hat{\sigma}'^T - \tilde{V}^T y r^T \tilde{W}^T \hat{\sigma}'^T) \\ & + tr(-\tilde{\phi}^T Q \|r\| + \tilde{\phi}^T Q \|r\|) + \|r\| \hat{\rho} + r^T v - \delta \end{aligned} \quad (64)$$

Fig. 13 Joint trajectories with FFNN but  $v = 0$ Fig. 14 Joint velocities with FFNN but  $v = 0$ Fig. 15 End-effector trajectories with FFNN but  $v = 0$ 

Using Eq. (56), we get

$$\dot{L} \leq -r^T K_d r + \|r\| \hat{\rho} - \frac{\|r\|^2 \hat{\rho}^2}{\|r\| \hat{\rho} + \delta} - \delta \quad (65)$$

$$\dot{L} \leq -r^T K_d r \quad (66)$$

Since  $L > 0$  and  $\dot{L} \leq 0$ , this shows the stability in the sense of Lyapunov so that  $r(t)$ ,  $\tilde{V}$  and  $\tilde{W}$  (and hence  $\hat{V}$ ,  $\hat{W}$ ) are bounded [21]. Similar to proof 1, we see that the force error  $\tilde{F} \rightarrow 0$ , position error  $K^{-1} \tilde{x} \rightarrow 0$  and redundant joint velocity error  $J^{-1} \dot{\tilde{q}} \rightarrow 0$ .

## 5 Simulation studies

The simulation has been performed for a two-link rigid robotic manipulator. The mathematical model of the manipulator dynamics is expressed as follows [21] (Fig. 2)



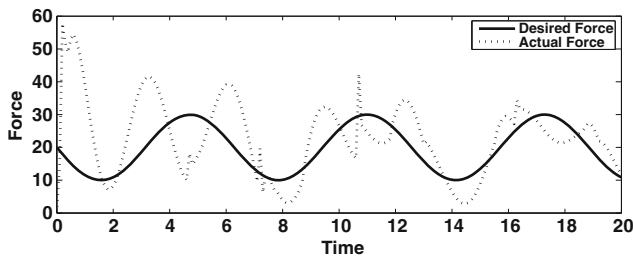
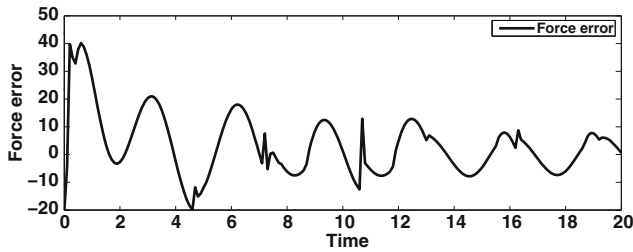
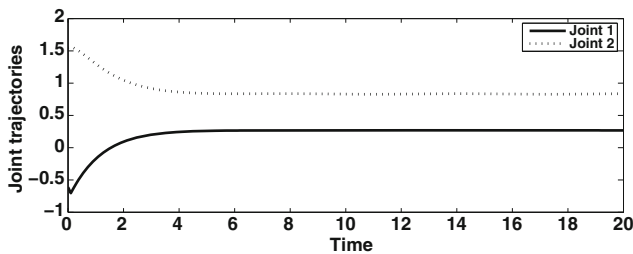
Fig. 16 Force tracking with FFNN but  $v = 0$ Fig. 17 Force tracking error with FFNN but  $v = 0$ 

Fig. 18 Joint trajectories with FFNN and adaptive compensator given by Eq. (56)

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} V_{m11} & V_{m12} \\ V_{m21} & V_{m22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} + \begin{bmatrix} F_{e1} \\ F_{e2} \end{bmatrix} + \begin{bmatrix} T_{d1} \\ T_{d2} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} - J^T(q)\bar{F} \quad (67)$$

where the mass matrix and gravity terms are given as follows

$$M_{11} = a + bC_2, \quad M_{12} = c + \frac{b}{2}C_2, \quad M_{21} = c + \frac{b}{2}C_2,$$

$$M_{22} = c$$

$$G_1 = dC_1 + eC_{12}, \quad G_2 = eC_{12}$$

$$F_{e1} = f_{e1}\dot{q}_1, \quad F_{e2} = f_{e2}\dot{q}_2$$

where  $a = l_2^2 m_2 + l_1^2(m_1 + m_2)$ ,  $b = 2l_1 l_2 m_2$ ,  $c = l_2^2 m_2$ ,  $d = (m_1 + m_2)l_1 g$ ,  $e = m_2 l_2 g$ ,  $C_1 = \cos q_1$ ,  $C_2 = \cos q_2$ ,  $C_{12} = \cos(q_1 + q_2)$ . Parameters  $a, b, c, d, e, f_{e1}$  and  $f_{e2}$  are

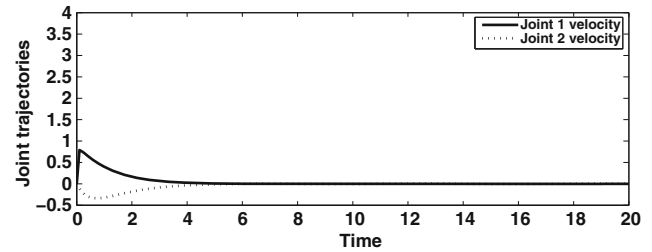


Fig. 19 Joint velocities with FFNN and adaptive compensator given by Eq. (56)

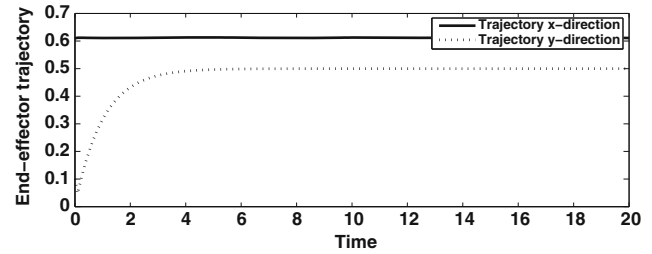


Fig. 20 End-effector trajectories with FFNN and adaptive compensator given by Eq. (56)

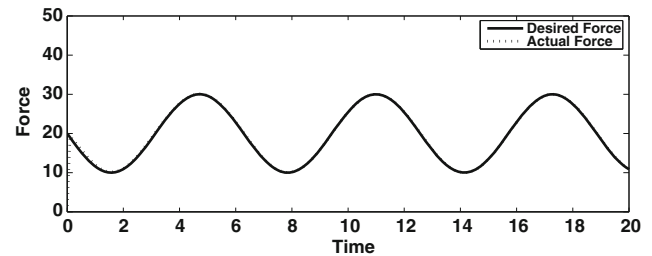
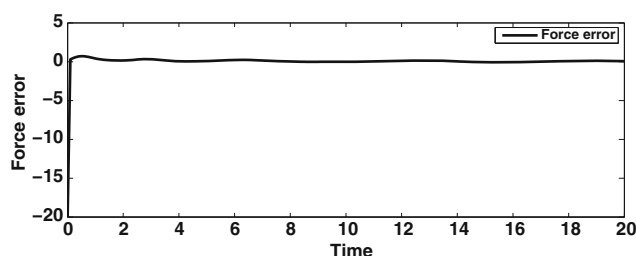


Fig. 21 Force tracking with FFNN and adaptive compensator given by Eq. (56)

unknown constants. Parameter values for two-link robot manipulator are set as given in [15] i.e.  $m_1 = 15.61$  kg,  $m_2 = 11.36$  kg,  $l_1 = 0.432$  m,  $l_2 = 0.432$  m,  $f_{e1} = 5.3$  Nm s,  $f_{e2} = 2.4$  Nm s,  $g = 9.8$  m/sec<sup>2</sup>, where  $m_i$ ,  $l_i$  denote the mass and link length, respectively of the link  $i = 1, 2$ .  $f_{e1}$  and  $f_{e2}$  denote the friction coefficient. External disturbance term  $T_d = [T_{d1} \ T_{d2}]^T$  is defined as  $T_d = 80[\cos(2t) \ \sin(2t)]^T$ . The constant parameter vector defined in robust controller in Eq. (29) is given as  $\hat{\theta} = [a \ b \ c \ d \ e \ f_{e1} \ f_{e2}]^T$  and we choose  $\rho = 100$ . The contact force only occurs in the  $x_1$  direction and is given by  $F = \kappa(x - x_{1e})$  where  $\kappa = 10^4$  and  $x_{1e} = 0.61$ . Also  $K^+ = 1/\kappa$ ,  $K^- = [0 \ 0; 0 \ 1]$ ,  $J^+ = J^{-1}$  and  $J^- = 0$ . The desired force  $F_d = 20(1 - 0.5\sin t)$ . The desired trajectory  $x_{2d}$  in  $x_2$  direction is  $0.5[1 - \exp(-t)]$ . The controller parameters are  $\lambda = 100$ ,  $K_d = 200I$ . To compare the



**Fig. 22** Force tracking error with FFNN and adaptive compensator given by Eq. (56)

performance of the controllers, we use scalar valued  $L^2$  norm as an performance index of an entire force error curve. The  $L^2$  norm measures the root-mean-square (RMS) average of the force error, which is given by

$$L^2(\tilde{F}) = \sqrt{\frac{1}{t_f - t_0} \int_{t_0}^{t_f} \|\tilde{F}\|^2 dt} \quad (68)$$

where  $t_0, t_f \in R_+$  are the initial and final times, respectively. A smaller  $L^2(\tilde{F})$  represents lesser force error and it indicates the better performance of the controller. For simulation studies, we have the following cases:

Case (i): Performance of the model-based controller given by Eq. (29) is shown in Figs. 3, 4, 5, 6, 7.

Case (ii): When we choose different robot parameter vector  $\hat{\theta} = [a \ b \ c \ d/2 \ e/2 \ f_{e1} \ f_{e2}]^T$ , the performance of the model-based controller given by Eq. (29) is shown in Figs. 8, 9, 10, 11, 12.

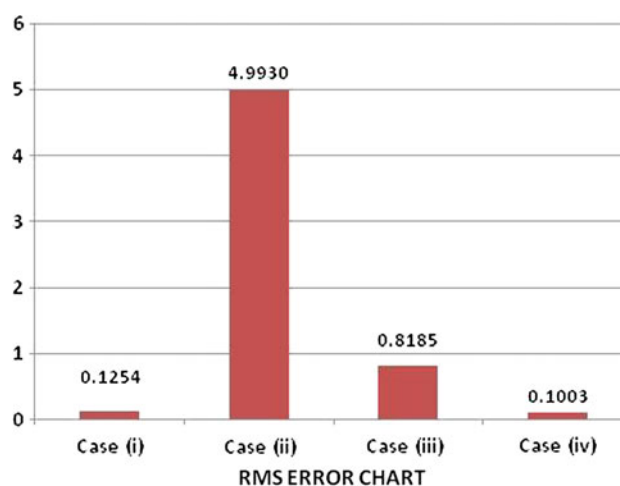
Case (iii): Performance of the neural network based controller given by Eq. (43) with adaptive compensator  $v = 0$  in Eq. (56) and disturbance term  $w \neq 0$  in Eq. (53) is shown in Figs. 13, 14, 15, 16, 17.

Case (iv): Performance of the neural network based controller given by Eq. (43) with adaptive compensator  $v \neq 0$  in Eq. (56) and disturbance term  $w \neq 0$  in Eq. (53) is shown in Figs. 18, 19, 20, 21, 22.

Root mean square of the force tracking errors for above cases is shown in Fig. 23.

### 5.1 Neural network architecture and simulation discussion

The simulation of the whole system is shown for 20 s. The architecture of the FFNN is composed of 7 input units and 1 bias unit, 5 hidden sigmoidal units, and 1 bias unit and 2 output units. For the simulation, we choose the parameters as  $F_w = 100I_6$ ,  $G_v = 100I_8$ ,  $\Gamma_\phi = 150I_3$ ,  $\gamma = 0.05$ ,  $\delta(0) = 0.5$ . Initial FFNN weights are selected at random and initial values for  $\hat{\phi}$  are  $[0 \ 0 \ 0]^T$ . The performance of the model-based controller (known dynamics) and neural



**Fig. 23** Root mean square of the force tracking errors

network-based controller (unknown dynamics) to produce the desired system response are shown from Figs. 3, 4, 5, 6, 7 to Figs. 18, 19, 20, 21, 22, respectively. The effect of the disturbance term  $w$  on the robot manipulator performance can be easily seen by Figs. 13, 14, 15, 16, 17. Figures 8, 9, 10, 11, 12 show that variation in robot parameter vector  $\hat{\theta} = [a \ b \ c \ d/2 \ e/2 \ f_{e1} \ f_{e2}]^T$  from its best guess estimation vector  $\hat{\theta} = [a \ b \ c \ d \ e \ f_{e1} \ f_{e2}]^T$ , adversely affects the performance of the model-based controller given by Eq. (29). When we change the robot parameter vector, the tracking error becomes large, that is, we can say the performance of the controller given by Eq. (29) depend on the model of robot manipulator. But, in neural network-based controller given by Eq. (43) any prior knowledge of the robot manipulator is not required. This is the sharp contrast to the model-based controller given by Eq. (29). From Fig. 23 can be seen that the root mean square error (RMSE) of employing the neural network-based controller with adaptive compensator [case (iv)] is less than the other three cases. Overall, Figs. 17, 18, 19, 20, 21, 22 clearly indicate that the FFNN-based controller given by Eq. (43) is quite effective in comparison to the controller given by Eq. (29).

## 6 Conclusion

In this paper, we designed a robust adaptive neural network-based hybrid position/force control scheme for robot manipulators in the presence of model uncertainties and external disturbance. It can be applied to any rigid-link robot manipulator without any prior knowledge about the robot such as masses and link lengths. A feedforward neural network (FFNN) is used to approximate the unknown dynamics of robot manipulator, which requires

no preliminary learning. The control purpose is to achieve the stability in the sense of Lyapunov generated by weighting matrices for desired interaction force between the end-effector and the environment and to regulate robot tip position in cartesian space. Moreover, an adaptive compensator  $v$  in Eq. (56) is also developed to be used as a controller to guarantee that the effect of the disturbance term  $w$  can be eliminated without requirement of known bounds. Finally, simulation results assure the validity of the proposed neural network-based controller and adaptive compensator with on-line updating laws to enhance the tracking performance of a 2-link rigid robot manipulator.

**Acknowledgments** This work is financially supported by the Council of Scientific and Industrial Research (CSIR), New Delhi, India, through file No. 09/143(0624)/2008-EMR-I.

## References

1. Raibert MH, Craig JJ (1981) Hybrid position/force control of manipulators. *ASME J Dyn Syst Measure Control* 103(2):126–133
2. Lozano R, Brogliato B (1992) Adaptive hybrid position/force control of redundant manipulators. *IEEE Trans Autom Control* 37(10):1501–1505
3. Yoshikawa T, Sudou A (1993) Dynamic hybrid position/force control of robot manipulators: online-estimation of unknown constraint. *IEEE Trans Robot Autom* 9(2):220–226
4. Kwan CM (1996) Robust adaptive force/motion control of constrained robots. *IEE Proc Control Theor Appl* 143(1):103–109
5. Xiao D, Ghosh BK, Ning X, Tarn TJ (2000) Dynamic hybrid position/force control of a robot manipulator in an uncalibrated environment. *IEEE Trans Control Syst Technol* 8(4):635–645
6. Kouya DN, Saad M, Lamarche L (2002) Backstepping adaptive hybrid force/position control of robotic manipulators. *Proceeding of the American control conference*, pp 4595–4600
7. Roy J, Whitcomb LL (2002) Adaptive force control of position/velocity controlled robots: theory and experiment. *IEEE Trans Robot Autom* 18(2):121–137
8. Cheah CC, Zhao Y, Slotine JJE (2006) Adaptive Jacobian motion and force control for constrained robots with uncertainties, *Proceeding of the international conference on robotics and automation*, pp 2226–2231
9. Kouya DN (2008) Adaptive hybrid force-position control of a robotic manipulator. *2nd European computing conference*, pp 323–331
10. Filaretov VF, Zuev AV (2008) Adaptive force/position control of robot manipulators. *Proceedings of the 2008 IEEE/ASME international conference on advanced intelligent mechatronics*, pp 96–101
11. Wnag H, Xie Y (2009) Adaptive Jacobian position/force tracking control of free-flying manipulators. *Robot Auton Syst* 57:173–181
12. Huang Q, Enomoto R (2008) Hybrid position, posture, force and moment control of robot manipulators. *Proceedings of the IEEE international conference on robotics and biomimetics*, pp 1444–1450
13. Lewis FL, Jagannathan S, Yesildirek A (1999) *Neural network control of robot manipulators and nonlinear systems*. Taylor and Francis, UK
14. Lewis FL, Yesildirek A, Liu K (1996) Multilayer neural net robot controller with guaranteed tracking performance. *IEEE Trans Neural Netw* 7(2):388–399
15. Kumar N, Panwar V, Sukavanam N, Sharma SP, Borm JH (2011) Neural network based hybrid force/position control for robot manipulators. *Int J Precis Eng Manuf* 12(3):419–426
16. Panwar V, Sukavanam N (2007) Design of optimal hybrid position/force controller for a robot manipulator using neural networks. *Math Probl Eng* 23:ID 65028
17. Karayiannidis Y, Rovithakis G, Doulgeri Z (2007) Force/position tracking for a robotic manipulator in compliant contact with a surface using neuro-adaptive control. *Automatica* 43:1281–1288
18. Ferguene F, Toumi R (2009) Dynamic external force feedback loop control of a robot manipulator using a neural network compensator-application to the trajectory following in an unknown environment. *Int J Appl Math Comput Sci* 19(1):113–126
19. Bechlioulis CP, Doulgeri Z, Rovithakis GA (2010) Neuro-adaptive force/position control with prescribed performance and guaranteed contact maintenance. *IEEE Trans Neural Netw* 21(12):1857–1868
20. Singh HP, Sukavanam N (2012) Neural network based control scheme for redundant robot manipulators subject to multiple self-motion criteria. *Math Comput Model* 55:1275–1300
21. Kwan C, Dawson DM, Lewis FL (2001) Robust adaptive control of robots using neural network: global stability. *Asian J Control* 3(2):111–121