Nonlinear Dyn (2011) 65:421–431
DOI 10.1007/s11071-010-9902-4

ORIGINAL PAPER

Neural network based robust hybrid control for robotic system: an H_∞ approach

Jinzhu Peng · Jie Wang · Yaonan Wang

Received: 22 February 2010 / Accepted: 28 November 2010 / Published online: 29 December 2010

Abstract A novel robust hybrid tracking control for robotic system is proposed. This hybrid control scheme combines computed torque control (CTC) with neural network, variable structure control (VSC) and nonlinear H_∞ control methods. It is assumed that the nominal system of robotic system is completely known, which is controlled by using CTC method. Neural network is designed to approximate parameter uncertainties, VSC is used to eliminate the effect of approximation error, and H_∞ control is employed to achieve a desired robust tracking performance. Based on Lyapunov stability theorem, it can be guaranteed that all signals in closed loop are bounded and a specified H_∞ tracking performance is achieved by employing the proposed robust hybrid control. The validity of the control scheme is shown by computer simulation of a two-link robotic manipulator.

Keywords Robotic system · Computed torque control · Neural network · Variable structure control · H_∞ control · Lyapunov stability

J. Peng (✉) · J. Wang
School of Electrical Engineering, Zhengzhou University, Zhengzhou, 450001, Henan, China
e-mail: jzpeng@zzu.edu.cn

Y. Wang
College of Electrical and Information Engineering, Hunan University, Changsha, 410082, Hunan, China

1 Introduction

Tracking control for robotic system always is a challenging problem since the uncertainties, disturbances and nonlinear system dynamics [1–3]. In the past decades, many control approaches have been proposed for dealing with these issues.

Computed torque control (CTC) [4] is a linearization method to linearize and decouple robotic dynamics, so that motion of each joint can be controlled individually and many well-developed linear control strategies can be used. However, the algorithm requires linearization of the nonlinear dynamical model of the robot and perfect knowledge of the system dynamics. Unfortunately, uncertainties always exist in practice, such as plant parameter variations, modeling errors, and unknown disturbances. These uncertainties will degrade the control performance. Even though, as a traditional control method, CTC is easily understood and of good performances and could not be neglected in designing controller for complex robotic system [5].

Variable structure control (VSC) has been investigated for controlling the robotic system [6–10]. The VSC provides a robust control method, which is insensitive to parameter uncertainties, modeling errors and external disturbances. However, there always exists high frequency oscillation in the control input, which is called 'chattering.' The high speed switching necessary for the establishment of a sliding mode causes the oscillations. In addition, the traditional VSC needs a plant model in the design, but it is difficult to obtain precise parameters of the system.

Neural networks have been widely used in modeling and controlling of nonlinear systems because of their capabilities of nonlinear function approximation, learning, and fault tolerance [11, 12]. Considering the robustness and trajectory tracking performance of dynamics, many adaptive neural network based control schemes have been developed to solve highly nonlinear control problems for robotic systems with uncertainties [13–17]. In general, a neural network system is employed to approximate uncertain nonlinearity of robotic systems, and applying the adaptive training algorithm to diminish the approximate errors, at the same time, using a robustifying controller (e.g. VSC controller) to compensate the approximate errors.

As robustness and its capability of disturbance attenuation in nonlinear systems, the approach of H_∞ optimal control has been widely discussed [19, 20], also applied in robotic systems. In conventional H_∞ controls, the plant models must be known, perhaps allowing a small perturbation. However, if the plant models have large uncertainties, the conventional H_∞ tracking control will meet additional difficulties. Recently, some attention has been paid to treating the robust H_∞ control for uncertain systems with the perturbation of dynamic uncertainties [14, 21–24].

Many researchers have tried to combine two or more above methods for controlling robotic system. Typically, Song et al. [5] proposed an approach of CTC combined fuzzy systems. The nominal system is controlled by CTC and for uncertain system, a fuzzy controller acts as a compensator. Zuo et al. [24] used a neural network robust controller acting as compensator and assumed that the external disturbance is finite energy. But they assumed that the system accelerations are measurable. Although the accelerations can be obtained through installing accelerometers on the robotic systems, the measurement noises and weight of these extra utilities would both sacrifice the tracking performance of robotic systems. Liu and Li [18] assumed that the dynamic parameters of robotic system can be expressed by the nominal parameters and the parametric uncertainties. Then, a fuzzy CMAC neural network was used to model the nominal system. Furthermore, the bounds of uncertainty parameters are assumed to be known and used to design the VSC, and an H_∞ controller is designed to achieve a certain tracking performance. In practice, in order to sufficiently consider system parameters' variety, these bounded values should be large enough; however, the larger bounded values would cause a more severe chattering in VSC design.

In this paper, without measuring the accelerations of robotic system, a novel neural network based robust hybrid tracking control scheme, combining CTC, neural network, VSC and H_∞ control for robotic system, is proposed. We assume that the constant parameters of nominal system, which is controlled by using CTC method matrix, are known. To avoid the problem of conservative bounds estimation in [18], a neural network is designed to approximate the uncertainties, VSC can eliminate the effect of approximation error, and H_∞ control is employed to achieve a certain robust tracking performance and to attenuate the effect of external disturbances to a prescribed level. The proposed robust hybrid controller can guarantee stability of closed-loop systems and H_∞ tracking performance by using Lyapunov stability theorem.

This paper is organized as follows. In Sect. 2, some preliminaries are addressed, which consist of mathematical notations, neural networks, dynamical models of robotic system with uncertainties, and detailed explanation related to CTC for robotic system. The design of the neural network based robust hybrid controller is given in Sect. 3, and the robust stability is analyzed. The Matlab simulation results are given in Sect. 4, and the conclusions are drawn in Sect. 5.

2 Preliminaries

Standard notations are used in this paper. Let $\Re$ be the real number set, $\Re^n$ be the n-dimensional vector space, $\Re^{n\times n}$ be the $n\times n$ real matrix space. The norm of vector $x\in\Re^n$ and that of matrix $A\in\Re^{n\times n}$ are defined, respectively, as $\|x\| = \sqrt{x^{\mathrm{T}}x}$ and $\|A\| = \mathrm{tr}(A^{\mathrm{T}}A)$. If y is a scalar, then $\|y\|$ denotes the absolute value. $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ are the minimum and the maximum eigenvalues of matrix A, respectively. $I_{n\times n}$ is an $n\times n$ identity matrix. And $\mathrm{sgn}(\cdot)$ is a standard sign function.

2.1 Neural network

In general, a three-layer neural network with multi-outputs $\hat{f}(x,W) = [\hat{f}_1(x,w_1),\ldots,\hat{f}_n(x,w_n)]^{\mathrm{T}}$ is usually used to approximate a continuous function

$f(x) = [f_1(x), \ldots, f_n(x)]^{\mathrm{T}}$, where x denotes the input vector and W contains the tunable network parameters. The neural network can be represented in the following form:

$$\hat{f}_k(x, w_k) = \sum_{j=1}^{p} \omega_{kj} \phi\left(\sum_{i=1}^{m} v_{ji} x_i + \theta_j\right) = w_k^{\mathrm{T}} \phi_k(\cdot) \tag{1}$$

with

$$w_k = \begin{bmatrix} \omega_{k1} \\ \vdots \\ \omega_{kp} \end{bmatrix}, \qquad \phi_k(\cdot) = \begin{bmatrix} \phi(\sum_{i=1}^{m} v_{1i} x_i + \theta_1) \\ \vdots \\ \phi(\sum_{i=1}^{m} v_{pi} x_i + \theta_p) \end{bmatrix},$$

where m, p, n denote the number of input, hidden and output neurons, respectively. The weights v_{ji} for $j = 1, \ldots, p$ and $i = 1, \ldots, m$ are the input layer to hidden layer interconnection weights, which are assumed to be fixed and selected randomly. The threshold offsets are denoted by θ_j for $j = 1, \ldots, p$ and ω_{kj} for $k = 1, \ldots, n$ and $j = 1, \ldots, p$ are the hidden layer to output layer interconnection weights, which are the tunable network parameters. $\phi(\cdot)$ denotes the activation function which should be a nonconstant, bounded and monotonically increasing continuous function, here is selected to be a bipolar sigmoid function

$$\phi(x) = \frac{1 - e^{-x}}{1 + e^{-x}}. \tag{2}$$

Then, the complete neural network can be denoted by

$$\hat{f}(x, W) = \begin{bmatrix} \hat{f}_1(x, w_1) \\ \vdots \\ \hat{f}_n(x, w_n) \end{bmatrix} = \begin{bmatrix} w_1^{\mathrm{T}} \phi_1(\cdot) \\ \vdots \\ w_n^{\mathrm{T}} \phi_n(\cdot) \end{bmatrix} = W^{\mathrm{T}} \Phi(\cdot), \tag{3}$$

where

$$W = \begin{bmatrix} w_1^{\mathrm{T}} \\ w_2^{\mathrm{T}} \\ \vdots \\ w_n^{\mathrm{T}} \end{bmatrix},$$

$$\Phi(\cdot) = \begin{bmatrix} \phi_1(\cdot) & 0 & \cdots & 0 \\ 0 & \phi_2(\cdot) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \phi_n(\cdot) \end{bmatrix}.$$

Here we use the smooth projection idea of Khalil [25]: consider $\Omega_0 = \{W : W^{\mathrm{T}} W \le M_W\}$ and $\Omega_W = \{W : W^{\mathrm{T}} W \le M_W + \delta\}$ for some $M_W > 0$ and $\delta > 0$, which can be arbitrarily pre-assigned by the designer. Define the smooth projection mapping as

$$\mathrm{Proj}[W, \Xi] = \begin{cases} \Xi, & \text{if } W^{\mathrm{T}} W \le M_W \\ & \text{or } (W^{\mathrm{T}} W > M_W \text{ and} \\ & W^{\mathrm{T}} \Xi \le 0) \\ \Xi - \frac{(W^{\mathrm{T}} W - M_W) W^{\mathrm{T}} \Xi}{\delta W^{\mathrm{T}} W} W, & \\ \text{otherwise.} & \end{cases} \tag{4}$$

The following assumptions are used throughout this study [22, 23]:

Assumption 1 *There exists an optimal parameter value $W^* \in \Omega_W$ such that $\hat{f}(x, W^*)$ can approximate $f(x)$ as closely as possible.*

Assumption 2 *Assume that the approximation error, $\varepsilon(x) = f(x) - \hat{f}(x, W^*)$, is bounded by a state-dependent function, that is, there exists a function $\kappa(x) > 0$ such that $|\varepsilon(x)| \le \kappa(x)$.*

2.2 Robotic system dynamics

The dynamic equation of robotic system is usually obtained by the Euler–Lagrange equation. Consider a general n-link rigid robot, which takes into account the external disturbances, with the equation of motion given by [2]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + \tau_{\mathrm{d}}, \tag{5}$$

where $q, \dot{q}, \ddot{q} \in \Re^n$ are the joint angular position, velocity and acceleration vectors of the robot, respectively; $M(q) \in \Re^{n \times n}$ is the symmetric and positive definite inertia matrix; $C(q, \dot{q}) \in \Re^{n \times n}$ is the effect of Coriolis and centrifugal forces; $G(q) \in \Re^n$ is the gravity vector; $\tau \in \Re^n$ is the torque input vector, and $\tau_{\mathrm{d}} \in \Re^n$ denotes unknown external disturbance.

For convenience, dynamical model (5) can be rewritten in the following compact form:

$$M(q)\ddot{q} + H(q, \dot{q}) = \tau + \tau_{\mathrm{d}}, \tag{6}$$

where $H(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q)$. The parameters $M(q)$, $C(q, \dot{q})$ and $G(q)$ in dynamical model (5) are functions of physical parameters of systems like links' masses, links' lengths, moments of inertial, and so on. The precise values of these parameters are difficult to acquire due to measuring errors, environment and payloads variations. Therefore, here it is assumed that actual values $M(q)$, $C(q,\dot{q})$ and $G(q)$ can be separated as nominal parts denoted by $M_0(q)$, $C_0(q,\dot{q})$ and $G_0(q)$ and uncertain parts denoted by $\Delta M(q)$, $\Delta C(q,\dot{q})$ and $\Delta G(q)$, respectively. These variables satisfy the following relationships:

$$\begin{aligned} M(q) &= M_0(q) + \Delta M(q), \\ C(q,\dot{q}) &= C_0(q,\dot{q}) + \Delta C(q,\dot{q}), \\ G(q) &= G_0(q) + \Delta G(q). \end{aligned}$$

Suppose that dynamical model of robotic system is known precisely and unmodeled dynamics and disturbances are excluded, i.e., $\Delta M(q)$, $\Delta C(q,\dot{q})$, $\Delta G(q)$ and τ_{d} are all zeros. Dynamical model (6) can be converted into the following nominal model:

$$M_0(q)\ddot{q} + H_0(q,\dot{q}) = \tau, \tag{7}$$

where $H_0(q,\dot{q}) = C_0(q,\dot{q})\dot{q} + G_0(q)$.

2.3 Computed torque control method for robotic system

The perfect knowledge of the robotic system dynamics should be known in the CTC design. The corresponding CTC control law for the nominal model (7) of robotic system can be chosen as

$$\tau = M_0(q)(\ddot{q}_{\mathrm{d}} - K_v\dot{e} - K_p e) + H_0(q,\dot{q}), \tag{8}$$

where e is the tracking error defined by $e = q - q_{\mathrm{d}}$, q and q_{d} are the actual and desired joint trajectories, respectively. The coefficients K_v and K_p should be chosen such that all the roots of the polynomial $h(s) = s^2 + K_v s + K_p$ are in the open left-half plane.

Assumption 3 *The desired trajectory q_{d} is continuous and bounded known function of time with bounded known derivatives up to the second order.*

Substituting (8) into (7) yields

$$\ddot{e} + K_v\dot{e} + K_p e = 0. \tag{9}$$

Remark 1 Obviously, CTC approach is based on feedback linearization technique, which results in a linear time-invariant closed-loop system (9), implying acquirement of globally asymptotical stability. Furthermore, explicit conditions for performance indices may be obtained in terms of controller gain matrices. More specifically, globally asymptotical stability is guaranteed provided that K_v and K_p in (9) are symmetric and positive definite constant matrices.

According to the analysis above, CTC strategy relies on strong assumptions that exact knowledge of robotic dynamics is precisely known and unmodeled dynamics has to be ignored. However, these assumptions are impossible in practical engineering. Therefore, applying (8) to the practical robotic system (6) yields

$$\begin{aligned} &M(q)(\ddot{e} + K_v\dot{e} + K_p e) \\ &\quad = -\Delta M(q)(\ddot{q}_{\mathrm{d}} - K_v\dot{e} - K_p e) \\ &\qquad - \Delta H(q,\dot{q}) + \tau_{\mathrm{d}}. \end{aligned} \tag{10}$$

Remark 2 All the parameters in the proposed scheme may be uncertain, which is true in practical situations.

Differently from the approaches proposed by Song et al. [5] and Zuo et al. [24] who assumed that the system accelerations are measurable, in this paper we assume that actual system accelerations are unmeasurable and a neural network based robust hybrid tracking control scheme is designed.

3 Robust hybrid control design

In this section, we will design a neural network based robust hybrid control law such that joint motions of robotic systems (5) can follow the desired trajectories. This hybrid controller is defined as

$$\tau = \tau_0 + u \tag{11}$$

where τ_0 is CTC defined like (8) and u is the neural network based hybrid compensator to be determined later.

Let $x_1 = e$, $x_2 = \dot{e}$, then the robotic system error dynamic equation (10) can be written as

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = M^{-1}(q)\{u - [M_0(q)K_v + \Delta C(q,\dot{q})] \\ \qquad - M_0(q)K_p x_1 - \Delta M(q)\ddot{q}_{\mathrm{d}} \\ \qquad - \Delta C(q,\dot{q})\dot{q}_{\mathrm{d}} - \Delta G(q) + \tau_{\mathrm{d}}\}. \end{cases} \tag{12}$$

Define $f(x_e) = \Delta M(q)\ddot{q}_\mathrm{d} + \Delta C(q,\dot{q})\dot{q}_\mathrm{d} + \Delta G(q)$ as a function of joint variables and parameter variations, where $x_e = [q^\mathrm{T}, \dot{q}^\mathrm{T}, \dot{q}_\mathrm{d}^\mathrm{T}, \ddot{q}_\mathrm{d}^\mathrm{T}]^\mathrm{T}$. Then, the error dynamic state-space equation has a form

$$\dot{x} = Ax + B\big(u - f(x_e) + \tau_\mathrm{d}\big), \tag{13}$$

where $x = [x_1^\mathrm{T}, x_2^\mathrm{T}]^\mathrm{T}$, $A = A_0 + \Delta A$, $B = (B_0 + \Delta B)M_0^{-1}(q)$, with

$$A_0 = \begin{bmatrix} 0_{n\times n} & I_{n\times n} \\ -K_p & -K_v \end{bmatrix},$$

$$\Delta A = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ M^{-1}(q)\Delta M(q)K_p & M^{-1}(q)(\Delta M(q)K_v - \Delta C(q,\dot{q})) \end{bmatrix},$$

$$B_0 = \begin{bmatrix} 0_{n\times n} \\ I_{n\times n} \end{bmatrix}, \qquad \Delta B = \begin{bmatrix} 0_{n\times n} \\ -M^{-1}(q)\Delta M(q) \end{bmatrix}.$$

Assumption 4 *ΔA and ΔB represent the time-varying parametric uncertainties having the following structure*:

$$[\Delta A, \Delta B] = DF[E_a, E_b],$$

where D, E_a, E_b are known constant matrix appropriate dimensions, $F \in \Re^{2n\times 2n}$ is unknown Lebesgue measurable matrix which is bounded as follows:

$$F^\mathrm{T}F \le I_{2n\times 2n}.$$

In this paper, a three-layer neural network described in Sect. 2.1 is used to approximate $f(x_e)$ in (13), which can be represented as

$$f(x_e) = W^{*\mathrm{T}}\Phi(\cdot) + \varepsilon(x_e), \tag{14}$$

and the neural network based hybrid compensator u in (11) is assumed to take the following form:

$$u = u_n + u_r, \tag{15}$$

where $u_n = W^\mathrm{T}\Phi(\cdot)$ is the neural network controller to approximate the unknown parameter uncertainties $f(x_e)$ online, and u_r is the robust controller to eliminate the approximation errors of neural network and achieve a certain robust performance.

Substituting (14) and (15) into the state-space equation (13) yields

$$\dot{x} = Ax + B\big(u_r - \tilde{W}^\mathrm{T}\Phi(\cdot) - \varepsilon(x_e) + \tau_\mathrm{d}\big), \tag{16}$$

where $\tilde{W} = W^* - W$.

Theorem 1 *Considering the robotic system dynamic equation (5), suppose that Assumptions 1–4 are satisfied, and $\bar{\tau}_\mathrm{d} \in L_2[0, +\infty)$, where $\bar{\tau}_\mathrm{d} = M_0^{-1}(q)\tau_\mathrm{d}$.*

If there exists a matrix $P = P^\mathrm{T} > 0$ and positive numbers γ, ξ, μ, ζ such that the following matrix inequality holds:

$$\begin{aligned} &PA_0 + A_0^\mathrm{T}P + Q + PB_0\Big(\frac{1}{\gamma^2}I_{n\times n} - 2R^{-1}\Big)B_0^\mathrm{T}P \\ &\quad + \Big(\frac{1}{\xi^2} + \mu^2\Big)PDD^\mathrm{T}P + \frac{1}{\mu^2}E_a^\mathrm{T}E_a \\ &\quad + \zeta^2 PB_0R^{-1}E_b^\mathrm{T}E_bR^{-1}B_0^\mathrm{T}P \\ &\quad + \frac{1}{\zeta^2}PD^\mathrm{T}DP \le 0 \end{aligned} \tag{17}$$

and $\operatorname{sgn}(B_0^\mathrm{T}Px) = \operatorname{sgn}(B^\mathrm{T}Px)$, where $Q = Q^\mathrm{T} > 0$ is a prescribed weighting matrix and R is some positive gain;

And, the control law τ is provided by (11), where τ_0 is CTC like (8) and u is the neural network based hybrid compensator designed as (15), in which the neural network controller u_n and its adaptive training law are designed as

$$u_n = W^\mathrm{T}\Phi(\cdot), \tag{18}$$

$$\dot{W} = -\eta\mathrm{Proj}[W, \Xi], \tag{19}$$

where η is the learning rate of neural network and $\Xi = \Phi(\cdot)^\mathrm{T}B_0^\mathrm{T}Px$;

And the robust controller u_r in (15) is defined as

$$u_r = u_h + u_s + u_w \tag{20}$$

with

$$u_h = -M_0(q)R^{-1}B_0^\mathrm{T}Px, \tag{21}$$

$$u_s = -\kappa(x_e)\operatorname{sgn}\big(B_0^\mathrm{T}Px\big), \tag{22}$$

$$u_w = -2M_W\|\Phi(\cdot)\|\operatorname{sgn}\big(B_0^\mathrm{T}Px\big); \tag{23}$$

Then, controllers can guarantee that (i) all the variables of the closed-loop system are bounded; (ii) the following H_∞ tracking performance is achieved:

$$\int_0^T \|x(t)\|_Q^2\,\mathrm{d}t \le \rho^2\int_0^T \|\bar{\tau}_\mathrm{d}\|^2\,\mathrm{d}t + \beta, \tag{24}$$

where ρ is an attenuation level and defined thereafter, $\beta \in \Re$.

Proof Let us select a Lyapunov function candidate,

$$V=\frac{1}{2}x^{\mathrm{T}}Px+\frac{1}{2\eta}\tilde{W}^{\mathrm{T}}\tilde{W}. \tag{25}$$

Differentiating the above equation and substituting the state-space equation (16) yields:

$$\begin{aligned}\dot{V}&=\frac{1}{2}\dot{x}^{\mathrm{T}}Px+\frac{1}{2}x^{\mathrm{T}}P\dot{x}+\frac{1}{\eta}\tilde{W}^{\mathrm{T}}\dot{\tilde{W}}\\&=\frac{1}{2}\big[Ax+B\big(u_r-\tilde{W}^{\mathrm{T}}\Phi(\cdot)-\varepsilon(x_e)+\tau_{\mathrm{d}}\big)\big]^{\mathrm{T}}Px\\&\quad+\frac{1}{2}x^{\mathrm{T}}P\big[Ax+B\big(u_r-\tilde{W}^{\mathrm{T}}\Phi(\cdot)-\varepsilon(x_e)+\tau_{\mathrm{d}}\big)\big]\\&\quad+\frac{1}{\eta}\tilde{W}^{\mathrm{T}}\dot{\tilde{W}}\\&=\frac{1}{2}x^{\mathrm{T}}\big(PA_0+A_0^{\mathrm{T}}P\big)x+x^{\mathrm{T}}PDFE_ax\\&\quad+\big(u_r-\tilde{W}^{\mathrm{T}}\Phi(\cdot)-\varepsilon(x_e)\big)^{\mathrm{T}}B^{\mathrm{T}}Px+\tau_{\mathrm{d}}^{\mathrm{T}}B^{\mathrm{T}}Px\\&\quad+\frac{1}{\eta}\tilde{W}^{\mathrm{T}}\dot{\tilde{W}}.\end{aligned} \tag{26}$$

Considering the matrix inequality (17) and the robust controller u_r, (26) can be bounded as

$$\begin{aligned}\dot{V}&\le-\frac{1}{2}x^{\mathrm{T}}\bigg\{Q+PB_0\bigg(\frac{1}{\gamma^2}I_{n\times n}-2R^{-1}\bigg)B_0^{\mathrm{T}}P\\&\quad+\bigg(\frac{1}{\xi^2}+\mu^2\bigg)PDD^{\mathrm{T}}P+\frac{1}{\mu^2}E_a^{\mathrm{T}}E_a\\&\quad+\zeta^2PB_0R^{-1}E_b^{\mathrm{T}}E_bR^{-1}B_0^{\mathrm{T}}P+\frac{1}{\zeta^2}PD^{\mathrm{T}}DP\bigg\}x\\&\quad+x^{\mathrm{T}}PDFE_ax+u_h^{\mathrm{T}}B^{\mathrm{T}}Px\\&\quad+\big(u_s-\varepsilon(x_e)\big)^{\mathrm{T}}B^{\mathrm{T}}Px+\tau_{\mathrm{d}}^{\mathrm{T}}B^{\mathrm{T}}Px\\&\quad+\big(u_w-\tilde{W}^{\mathrm{T}}\Phi(\cdot)\big)^{\mathrm{T}}B^{\mathrm{T}}Px+\frac{1}{\eta}\tilde{W}^{\mathrm{T}}\dot{\tilde{W}}.\end{aligned} \tag{27}$$

According to (4), the proof of $\|W\|\le M_W$ can be found in [25]. Since $\|W^*\|\le M_W$, there is $\|\tilde{W}\|\le 2M_W$ [18]. And considering the fact $\dot{\tilde{W}}=-\dot{W}$ and the controller u_w in (23), the training law (19) of neural network implies that

$$\big(u_w-\tilde{W}^{\mathrm{T}}\Phi(\cdot)\big)^{\mathrm{T}}B^{\mathrm{T}}Px+\frac{1}{\eta}\tilde{W}^{\mathrm{T}}\dot{\tilde{W}}\le 0. \tag{28}$$

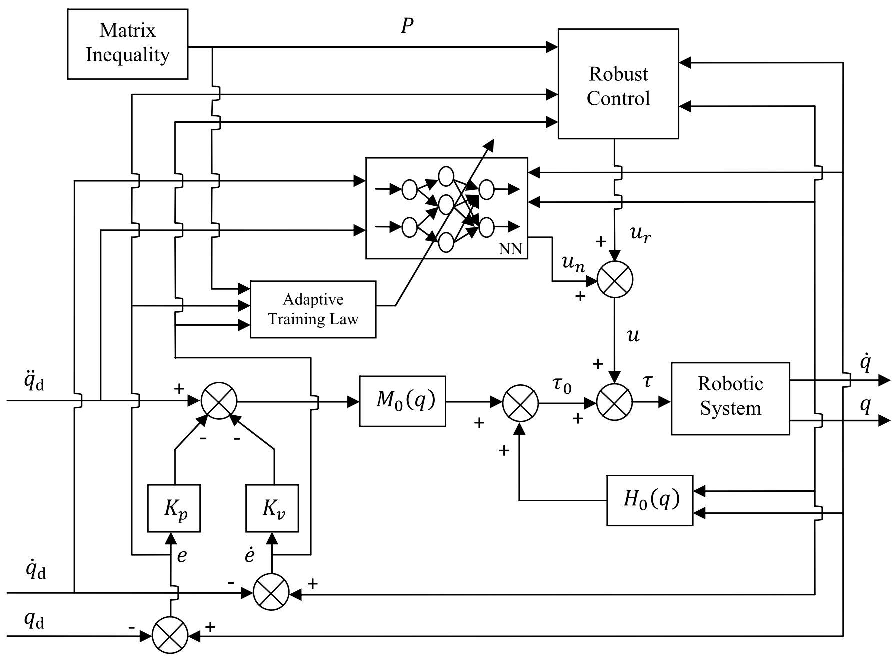
Then, considering the controller u_s in (22) and Assumption 2, we obtain

$$\begin{aligned}&\big(u_s-\varepsilon(x_e)\big)^{\mathrm{T}}B^{\mathrm{T}}Px\\&\quad\le-\kappa(x_e)\big|B^{\mathrm{T}}Px\big|+\big|\varepsilon(x_e)\big|\big|B^{\mathrm{T}}Px\big|\\&\quad\le 0.\end{aligned} \tag{29}$$

Considering the H_∞ controller u_h in (21) and substituting (28) and (29) into (27) yields:

$$\begin{aligned}\dot{V}&\le-\frac{1}{2}x^{\mathrm{T}}\bigg\{Q+PB_0\bigg(\frac{1}{\gamma^2}I_{n\times n}-2R^{-1}\bigg)B_0^{\mathrm{T}}P\\&\quad+\bigg(\frac{1}{\xi^2}+\mu^2\bigg)PDD^{\mathrm{T}}P+\frac{1}{\mu^2}E_a^{\mathrm{T}}E_a\\&\quad+\zeta^2PB_0R^{-1}E_b^{\mathrm{T}}E_bR^{-1}B_0^{\mathrm{T}}P+\frac{1}{\zeta^2}PD^{\mathrm{T}}DP\bigg\}x\\&\quad+x^{\mathrm{T}}PDFE_ax+u_h^{\mathrm{T}}B^{\mathrm{T}}Px+\tau_{\mathrm{d}}^{\mathrm{T}}B^{\mathrm{T}}Px\\&=-\frac{1}{2}x^{\mathrm{T}}\bigg\{Q+PB_0\bigg(\frac{1}{\gamma^2}I_{n\times n}-2R^{-1}\bigg)B_0^{\mathrm{T}}P\\&\quad+\bigg(\frac{1}{\xi^2}+\mu^2\bigg)PDD^{\mathrm{T}}P+\frac{1}{\mu^2}E_a^{\mathrm{T}}E_a\\&\quad+\zeta^2PB_0R^{-1}E_b^{\mathrm{T}}E_bR^{-1}B_0^{\mathrm{T}}P+\frac{1}{\zeta^2}PD^{\mathrm{T}}DP\bigg\}x\\&\quad+x^{\mathrm{T}}PDFE_ax-x^{\mathrm{T}}PB_0R^{-1}B_0^{\mathrm{T}}Px\\&\quad-x^{\mathrm{T}}PB_0R^{-1}E_b^{\mathrm{T}}F^{\mathrm{T}}D^{\mathrm{T}}Px\\&\quad+\bar{\tau}_{\mathrm{d}}^{\mathrm{T}}B_0^{\mathrm{T}}Px+\bar{\tau}_{\mathrm{d}}^{\mathrm{T}}E_b^{\mathrm{T}}F^{\mathrm{T}}D^{\mathrm{T}}Px\\&=-\frac{1}{2}\bigg(\frac{1}{\gamma}B_0^{\mathrm{T}}Px-\gamma\bar{\tau}_{\mathrm{d}}\bigg)^{\mathrm{T}}\bigg(\frac{1}{\gamma}B_0^{\mathrm{T}}Px-\gamma\bar{\tau}_{\mathrm{d}}\bigg)\\&\quad-\frac{1}{2}\bigg(\frac{1}{\xi}F^{\mathrm{T}}D^{\mathrm{T}}Px-\xi E_b\bar{\tau}_{\mathrm{d}}\bigg)^{\mathrm{T}}\\&\quad\cdot\bigg(\frac{1}{\xi}F^{\mathrm{T}}D^{\mathrm{T}}Px-\xi E_b\bar{\tau}_{\mathrm{d}}\bigg)\\&\quad-\frac{1}{2}\bigg(\mu D^{\mathrm{T}}Px-\frac{1}{\mu}FE_ax\bigg)^{\mathrm{T}}\\&\quad\cdot\bigg(\mu D^{\mathrm{T}}Px-\frac{1}{\mu}FE_ax\bigg)\\&\quad-\frac{1}{2}\bigg(\zeta FE_bR^{-1}B_0^{\mathrm{T}}Px-\frac{1}{\zeta}D^{\mathrm{T}}Px\bigg)^{\mathrm{T}}\\&\quad\cdot\bigg(\zeta FE_bR^{-1}B_0^{\mathrm{T}}Px-\frac{1}{\zeta}D^{\mathrm{T}}Px\bigg)\end{aligned}$$

Fig. 1 Architecture of closed-loop system



$$
\begin{aligned}
&-\frac{1}{2}\frac{1}{\xi^2}x^{\mathrm{T}}PD^{\mathrm{T}}\big(I_{2n\times 2n}-F^{\mathrm{T}}F\big)DPx \\
&-\frac{1}{2}\frac{1}{\mu^2}x^{\mathrm{T}}E_a^{\mathrm{T}}\big(I_{2n\times 2n}-F^{\mathrm{T}}F\big)E_a x \\
&-\frac{1}{2}\zeta^2 x^{\mathrm{T}}PB_0R^{-1}E_b^{\mathrm{T}}\big(I_{2n\times 2n} \\
&-F^{\mathrm{T}}F\big)E_bR^{-1}B_0^{\mathrm{T}}Px \\
&+\frac{1}{2}\gamma^2\bar{\tau}_{\mathrm{d}}^{\mathrm{T}}\bar{\tau}_{\mathrm{d}}+\frac{1}{2}\xi^2\bar{\tau}_{\mathrm{d}}^{\mathrm{T}}E_b^{\mathrm{T}}E_b\bar{\tau}_{\mathrm{d}}-\frac{1}{2}x^{\mathrm{T}}Qx. \qquad (30)
\end{aligned}
$$

Then, according to Assumption 4, we have

$$
\begin{aligned}
\dot{V} &\le \frac{1}{2}\gamma^2\bar{\tau}_{\mathrm{d}}^{\mathrm{T}}\bar{\tau}_{\mathrm{d}}+\frac{1}{2}\xi^2\bar{\tau}_{\mathrm{d}}^{\mathrm{T}}E_b^{\mathrm{T}}E_b\bar{\tau}_{\mathrm{d}}-\frac{1}{2}x^{\mathrm{T}}Qx \\
&=\frac{1}{2}\rho^2\bar{\tau}_{\mathrm{d}}^{\mathrm{T}}\bar{\tau}_{\mathrm{d}}-\frac{1}{2}x^{\mathrm{T}}Qx, \qquad (31)
\end{aligned}
$$

where $\rho^2=\gamma^2+\xi^2\|E_b\|^2$.

Integrating the above inequality from $t=0$ to $t=T$ yields

$$
V(T)-V(0)\le\frac{1}{2}\rho^2\int_0^T\bar{\tau}_{\mathrm{d}}^{\mathrm{T}}\bar{\tau}_{\mathrm{d}}\,\mathrm{d}t-\frac{1}{2}\int_0^T x^{\mathrm{T}}Qx\,\mathrm{d}t. \qquad (32)
$$

Since $V(T)\ge 0$, the above inequality leads to

$$
\int_0^T\|x(t)\|_Q^2\,\mathrm{d}t\le 2V(0)+\rho^2\int_0^T\|\bar{\tau}_{\mathrm{d}}\|^2\,\mathrm{d}t. \qquad (33)
$$

Defining $\beta=2V(0)=x^{\mathrm{T}}(0)Px(0)+\frac{1}{\eta}\tilde{W}^{\mathrm{T}}(0)\tilde{W}(0)$, then the H_∞ performance is achieved.

Since $\bar{\tau}_{\mathrm{d}}\in L_2[0,+\infty)$, there is a finite constant $M_d>0$ such that $\int_0^\infty\|\bar{\tau}_{\mathrm{d}}\|^2\,\mathrm{d}t\le M_d$, and then we have

$$
\|x\|\le\sqrt{\frac{\beta+\rho^2M_d}{\lambda_{\min}(Q)}}. \qquad (34)
$$

It can be concluded that all signals of the closed-loop system are bounded. □

According to the above analysis, the architecture of closed-loop system is shown in Fig. 1.

Remark 3 In this paper, the VSC u_s and u_w are designed to eliminate the effect of the approximation errors of neural network and ensure system stability, and the robust H_∞ controller u_h is employed to achieve the desired H_∞ tracking performance. In this way, global stability and H_∞ tracking performance of closed-loop systems are achieved.

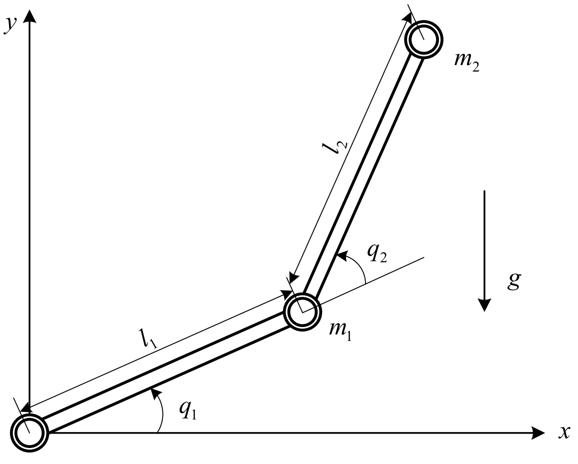


Fig. 2 Diagram of a two-link robot manipulator

4 Simulation example

To verify the theoretical results, simulations were carried out in two degrees of freedom robotic manipulator as shown in Fig. 2 and described by [2]:

$$M(q)=\begin{bmatrix} m_1l_1^2+m_2(l_1^2+l_2^2+2l_1l_2c_2) & m_2l_2^2+m_2l_1l_2c_2 \\ m_2l_2^2+m_2l_1l_2c_2 & m_2l_2^2 \end{bmatrix},$$

$$C(q,\dot{q})=\begin{bmatrix} -2m_2l_1l_2s_2\dot{q}_2 & m_2l_1l_2s_2\dot{q}_2 \\ m_2l_1l_2s_2\dot{q}_2 & 0 \end{bmatrix},$$

$$G(q)=\begin{bmatrix} m_2l_2gc_{12}+(m_1+m_2)l_1gc_1 \\ m_2l_2gc_{12} \end{bmatrix},$$

where m_1 and m_2 are the masses of link 1 and link 2, respectively; l_1 and l_2 are the lengths of link 1 and link 2, respectively; s_i denotes $\sin(q_i)$, c_i denotes $\cos(q_i)$, and c_{ij} denotes $\cos(q_i+q_j)$, for $i=1,2$ and $j=1,2$; and g is acceleration of gravity.

4.1 Design procedure

To summarize the analysis in Sect. 3, the step-by-step procedures of the neural network based robust hybrid control for robotic system are outlined as follows.

Step 1. Select controller gains $K_p=100I_{2\times 2}$ and $K_v=50I_{2\times 2}$ such as

$$A_0=\begin{bmatrix} 0_{2\times 2} & I_{2\times 2} \\ -100I_{2\times 2} & -50I_{2\times 2} \end{bmatrix}$$

is a Hurwitz matrix.

Step 2. Choose proper parameters:

$$Q=20I_{4\times 4}, \qquad R=10I_{2\times 2}, \qquad \gamma=0.4,$$

$$\xi=1, \qquad \mu=1, \qquad \zeta=1.$$

$$D=[0,0,5,5]^{\mathrm{T}}_{1\times 4}, \qquad E_a=[0,0,10,10]_{1\times 4},$$

$$E_a=[1,1]_{1\times 4},$$

solve P from matrix inequality (17):

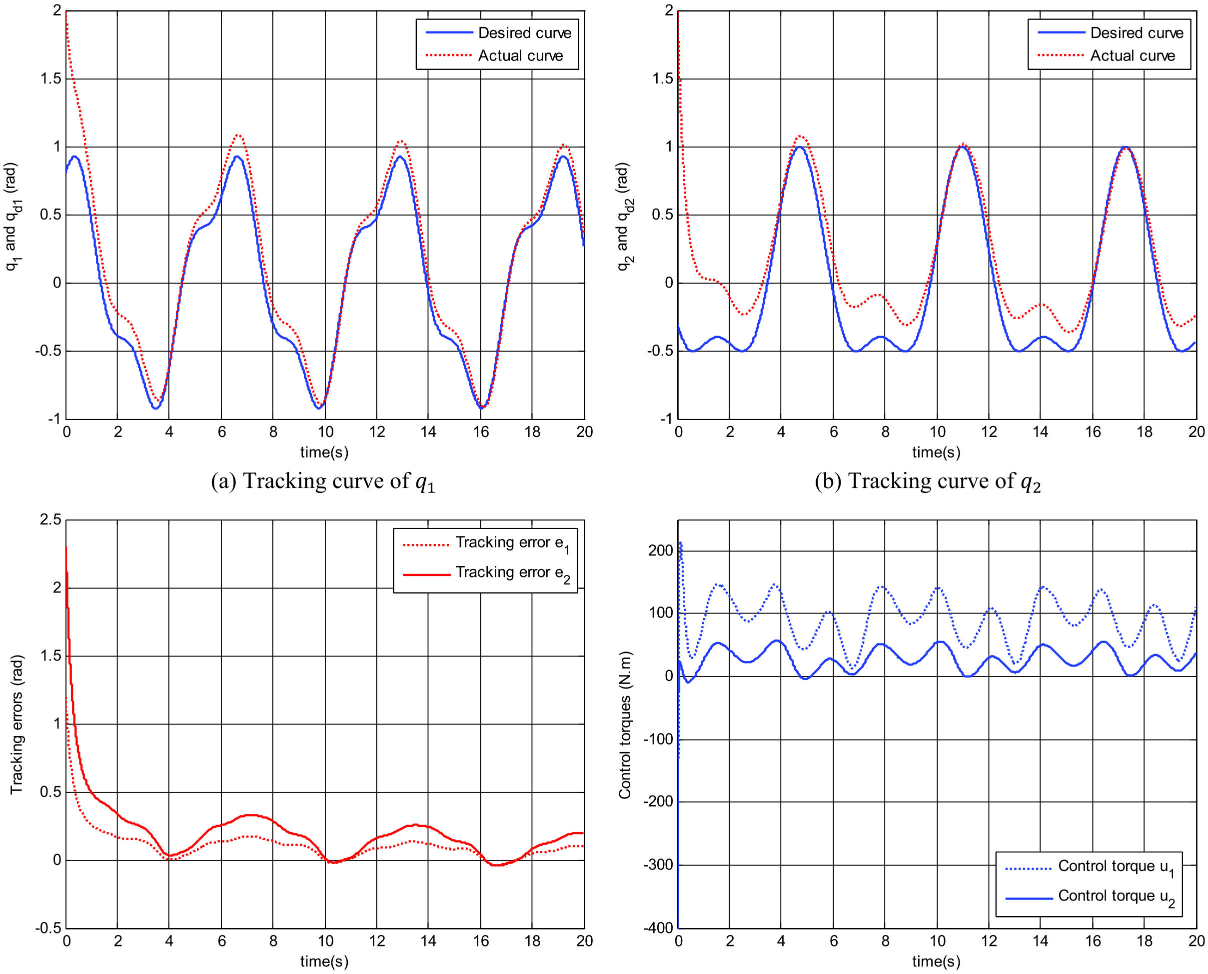
$$P=\begin{bmatrix} 12.7806 & -12.3229 & 0.0124 & -0.0874 \\ -12.3229 & 12.7806 & -0.0874 & 0.0124 \\ 0.0124 & -0.0874 & 0.1495 & -0.0508 \\ -0.0874 & 0.0124 & -0.0508 & 0.1495 \end{bmatrix}.$$

Step 3. Select initial weight $W(0)=0_{12\times 2}$, design the neural network structure as 8–12–2, and select the learning rate as $\eta=0.2$ in (19), set $M_W=1000$ and $\delta=10$, then the neural network controller u_n can be obtained from (18).

Step 4. According to (20)–(23) with the function $\kappa(x_e)=2\sqrt{\|e\|^2+\|\dot{e}\|^2}$ the robust controller u_r can be calculated; by using (15) we can obtain the neural network based robust hybrid compensator u. The corresponding neural network based robust hybrid control law can be obtained from Theorem 1.

4.2 Simulation results

In this section, the proposed control approach is applied to the position control of a two-link manipulator. The dynamic equation and parameters of the manipulator are similar to those in [5]. The nominal parameters of the robot used for simulation are $m_1=m_2=4$ kg and $l_1=l_2=1$ m, $g=9.8\ \mathrm{m/s^2}$, while actual parameters of robot are chosen as $m_1=m_2=8$ kg and $l_1=l_2=1.2$ m to introduce the parameter uncertainties. The desired trajectories to be tracked are $q_{\mathrm{d}1}=0.8\cos(t)+0.2\sin(3t)$, $q_{\mathrm{d}2}=-0.3\cos(2t)-0.7\sin(t)$. The initial conditions are $q_1(0)=q_2(0)=2$ rad and $\dot{q}_1(0)=\dot{q}_2(0)=0$ rad/s. The CTC for controlling robotic system without model uncertainties and disturbances can be found in [5, 24]. It is obvious that CTC can track the given trajectories perfectly under precise known model parameters. For the purpose of comparison, simulation studies in two cases are conducted. To show the robustness of the controller, we choose the exogenous disturbances $\tau_{\mathrm{d}}=[5e^{-t},-5e^{-t}]^{\mathrm{T}}$.



(a) Tracking curve of q_1

(b) Tracking curve of q_2

(c) Tracking errors

(d) Control torques

Fig. 3 CTC for robotic manipulator with uncertainties and disturbances

Case 1. The conventional CTC (8) for controlling robotic system under model uncertainties and disturbances is demonstrated. Figure 3 shows the results. It can be seen that the controller cannot drive the joints to reach the desired positions and steady-state tracking error exists.

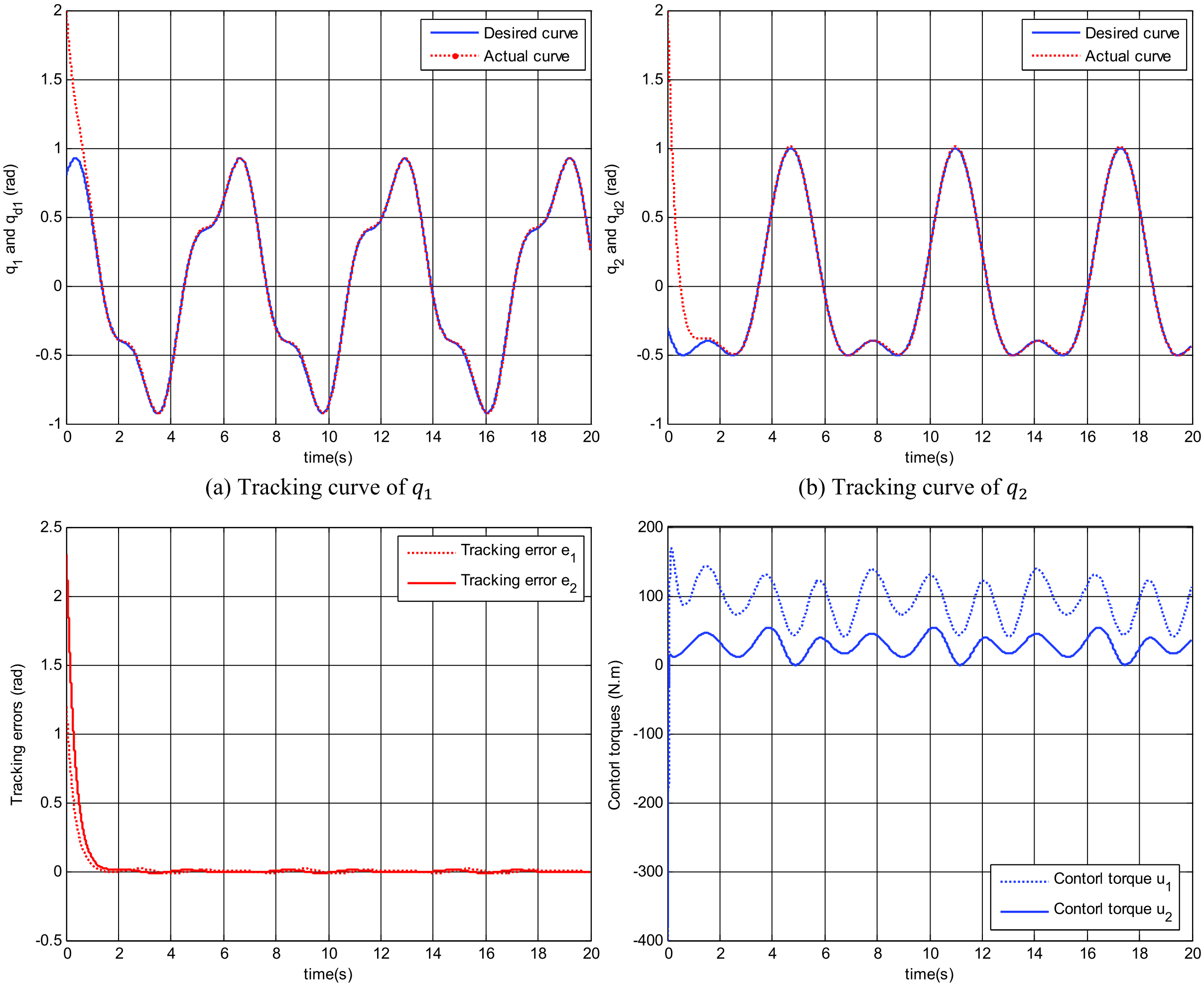
Case 2. Under the same conditions as in Case 1, the proposed neural network based robust hybrid tracking controller is used to control robotic manipulator. The control procedure is described in foregoing subsection. Figure 4 shows the tracking results. It is observed that the tracking error decreases quickly, and the effects of uncertainties are successfully compensated by the robust hybrid control term.

To quantify the control performance, the root-mean square average of tracking error (based on the L_2 norm of the tracking errors e) is given as follows [24]:

$$L_2(e) = \sqrt{\frac{1}{T}\int_0^T e^{\mathrm{T}} e \, \mathrm{d}t}, \tag{35}$$

where T represents the total simulation time. Table 1 shows the L_2 error norms for CTC method and the proposed robust control method. Note that the smaller L_2 norm represents the better performance.

From Fig. 4 and Table 1, the results demonstrate that the proposed robust hybrid tracking control can effectively control the rigid robotic system with model uncertainties and disturbances, and the proposed con-



(a) Tracking curve of q_1
(b) Tracking curve of q_2
(c) Tracking errors
(d) Control torques

Fig. 4 The proposed hybrid tracking control for robotic manipulator with uncertainties and disturbances

Table 1 L_2 norm for tracking errors

Controller	$L_2(e_1)$	$L_2(e_2)$
CTC	5.2152	10.0253
Proposed	3.5464	6.8693

troller presents a better transient response and a smaller tracking error norm than the CTC method.

5 Conclusions

In this paper, a neural network based robust hybrid tracking control scheme, combining CTC, neural network, VSC and H_∞ control for robotic manipulator without measuring the system accelerations, is proposed. In this scheme, CTC is employed to control the completely known nominal system of robotic manipulator; neural network is designed to approximate parameter uncertainties, the VSC and H_∞ controller can guarantee robustness to approximation errors and also attenuate the effect of finite-energy, immeasurable external disturbances entering the system. It can be guaranteed that all signals in the closed loop are bounded and ensured H_∞ tracking performance by employing the proposed robust hybrid control. The validity of the control scheme is shown by computer simulation of a two-link robotic manipulator.

Acknowledgements This work was partially supported by National Natural Science Foundation of China (Nos. 60775047 and 60835004), and National High Technology Research and Development Program ('863 Program') of China (Nos. 2007-AA04Z244 and 2008AA04Z214).

References

1. Book, W.J.: Modelling design and control of flexible manipulator arms: a tutorial review. In: Proceedings of 29th IEEE Conference on Decision and Control, pp. 500–506. IEEE Press, San Francisco (1990)
2. Lewis, F.L., Jagannathan, S., Yesildirek, A.: Neural Network Control of Robot Manipulators and Nonlinear Systems. SIAM Press, Philadelphia (2002)
3. Luh, J.Y.S.: Conventional controller design for industrial robots—a tutorial. IEEE Trans. Syst. Man Cybern. **13**, 298–316 (1983)
4. Middleton, R.H., Goodwin, G.C.: Adaptive computed torque control for rigid link manipulators. Syst. Control Lett. **10**(1), 9–16 (1988)
5. Song, Z., Yi, J., Zhao, D., Li, X.: A computed torque controller for uncertain robotic manipulator systems: fuzzy approach. Fuzzy Sets Syst. **154**, 208–226 (2005)
6. Kaynak, O., Erbatur, K., Ertugrul, M.: The fusion of computationally intelligent methodologies and sliding-mode control – a survey. IEEE Trans. Ind. Electron. **48**, 4–17 (2001)
7. Yeung, K.S., Chen, Y.P.: A new controller design for manipulators using the theory of variable structure systems. IEEE Trans. Autom. Control **33**, 200–209 (1988)
8. Hu, H., Woo, P.Y.: Fuzzy supervisory sliding-mode and neural-network control for robotic manipulators. IEEE Trans. Ind. Electron. **53**(3), 929–940 (2006)
9. Sun, F., Sun, Z., Feng, G.: An adaptive fuzzy controller based on sliding mode for robot manipulator. IEEE Trans. Syst. Man Cybern. **29**, 661–667 (1999)
10. Hsu, Y.C., Chen, G., Li, H.X.: A fuzzy adaptive variable structure controller with applications to robot manipulators. IEEE Trans. Syst. Man Cybern. **31**(3), 331–340 (2001)
11. Chen, M., Jiang, C., Wu, Q.: Backstepping control for a class of uncertain nonlinear systems with neural network. Int. J. Nonlinear Sci. **2**(3), 137–143 (2007)
12. Wai, R.: Hybrid fuzzy neural-network control for nonlinear motor-toggle servomechanism. IEEE Trans. Control Syst. Technol. **10**(4), 519–532 (2002)
13. Behera, L., Chaudhury, S., Gopal, M.: Neuro-adaptive hybrid controller for robot manipulator tracking control. IEE Proc., Control Theory Appl. **143**(4), 270–275 (1996)
14. Chang, Y., Chen, B.: A nonlinear adaptive H_∞ tracking control design in robotic systems via neural networks. IEEE Trans. Control Syst. Technol. **5**(2), 13–29 (1997)
15. Lewis, F.L., Liu, K., Yesildirek, A.: Neural net robot controller with guaranteed tracking performance. IEEE Trans. Neural Netw. **6**, 703–715 (1995)
16. Lewis, F.L., Yesildirek, A., Liu, K.: Multilayer neural-net robot controller with guaranteed tracking performance. IEEE Trans. Neural Netw. **7**, 388–399 (1996)
17. Kim, Y., Lewis, F.L.: Neural network output feedback control of robot manipulators. IEEE Trans. Robot. Autom. **15**(2), 301–309 (1999)
18. Liu, Z., Li, C.: Fuzzy neural networks quadratic stabilization output feedback control for biped robots via H_∞ approach. IEEE Trans. Syst. Man Cybern. **33**(1), 67–84 (2003)
19. Shen, T., Tamura, K.: Robust H_∞ control of uncertain nonlinear system via state feedback. IEEE Trans. Autom. Control **40**(5), 766–768 (1995)
20. Kung, C., Chen, T.: H_∞ tracking-based adaptive fuzzy sliding mode controller design for nonlinear systems. IET Control Theory Appl. **1**(1), 82–89 (2007)
21. Chen, B.S., Lee, T.S., Feng, J.H.: A nonlinear H_∞ control design in robotic systems under parameter perturbation and external disturbance. Int. J. Control **59**(2), 439–461 (1994)
22. Siqueira, A.A.G., Terra, M.H.: Neural network-based H_∞ control for fully actuated and underactuated cooperative manipulators. Control Eng. Pract. **17**, 418–425 (2009)
23. Chang, Y.C.: Neural network-based H_∞ tracking control for robotic systems. IEE Proc., Control Theory Appl. **147**(3), 303–311 (2000)
24. Zuo, Y., Wang, Y., Liu, X.: Neural network robust H_∞ tracking control strategy for robot manipulators. Appl. Math. Model. **34**(7), 1823–1838 (2010)
25. Khalil, H.: Adaptive output feedback control of nonlinear systems represented by input–output models. IEEE Trans. Autom. Control **41**(3), 177–188 (1996)