J Control Theory Appl 2010 8 (2) 249–256
DOI 10.1007/s11768-010-8038-x

Adaptive RBF neural network control of robot with actuator nonlinearities

Jinkun LIU, Yu LU
(School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: In this paper, an adaptive neural network control scheme for robot manipulators with actuator nonlinearities is presented. The control scheme consists of an adaptive neural network controller and an actuator nonlinearities compensator. Since the actuator nonlinearities are usually included in the robot driving motor, a compensator using radial basis function (RBF) network is proposed to estimate the actuator nonlinearities and eliminate their effects. Subsequently, an adaptive neural network controller that neither requires the evaluation of inverse dynamical model nor the time-consuming training process is given. In addition, GL matrix and its product operator are introduced to help prove the stability of the closed control system. Considering the adaptive neural network controller and the RBF network compensator as the whole control scheme, the closed-loop system is proved to be uniformly ultimately bounded (UUB). The whole scheme provides a general procedure to control the robot manipulators with actuator nonlinearities. Simulation results verify the effectiveness of the designed scheme and the theoretical discussion.

Keywords: Adaptive control; RBF neural network; Actuator nonlinearity; Robot manipulator; Deadzone

1 Introduction

Actuator nonlinearities, including deadzone, backlash, saturation, etc. [1], are quite common in the actual robot manipulators. Because of the non-analytical nature of the actuator nonlinearities, their exact nonlinear functions are unknown. These nonlinearities often have a great influence on system performances. The problems are particularly exacerbated when the required accuracy is high. For example, the deadzone in the actuator may cause limit cycles or poor performance that cannot satisfy the high accuracy-tracking requirement. Adaptive control for output backlash and deadzone compensator using fuzzy logic control are given in [2∼4]. Compensation of hysteresis is given in [5]. The method presented by Lewis et al. [6] is very attractive, because it is a general procedure to compensate the unknown actuator nonlinearities. The compensator uses two neural networks, one to estimate the unknown actuator nonlinearities and the other to provide adaptive compensation in the feedforward path. However, the controller of the system in [6] is not given exactly and it is needed to know the upper bounded error of the nonlinear plant estimation.

Unlike many other neural networks, for example, the BP neural network [7∼9], the RBF neural network has the universal approximation property and can avoid the local minima problem; the network can not only reduce the parameters of neural netwok tuning but also make the initialization of neural network much easier [10], so it is very suitable for realtime control applications. In the past few years, RBF neural network has been used for the on-line estimation of unknown functions in nonlinear adaptive control systems, and many researchers have used the neural network for a wide range of applications in nonlinear adaptive control [11∼13].

Our main efforts in this paper are highlighted as follows:

1) Instead of BP neural network in [6], three RBF networks with parameter adaptation law are used to emulate the inertia matrix, centripetal matrix and gravitational vector of robot directly. Nolinear robot function $f(x)$ estimate is not necessary. Chattering caused by switch gain can be decreased effectively.

2) The neural network used in [6] is designed by modified multilayer BP-network, which has too much parameters that need to be tuned and is not suitable for real-time control. In this paper, the RBF neural network is adopted to solve the problem.

3) GL matrix and its product operator are introduced to help prove the stability of n-link robot manipulator, and the closed-loop system is proved to be uniformly ultimately bounded (UUB).

The compensator not only provides compensation signals in the feedforward path, but also takes part in the controller. The remaining part of this paper will be organized as follows. In Section 2, the dynamics of the robot with actuator nonlinearity (deadzone) is described. In Section 3, an NN controller with parameter adaptation laws is given. In Section 4, the authors propose an RBF neural network deadzone compensator and describe the idea underlying the approach. In Section 5, the closed-loop system is proved to be uniformly ultimately bounded (UUB). In Section 6, the simulations are provided to illustrate the performance of the whole tracking system. Finally, we give the conclusion in Section 7.

2 Dynamics of robot system and deadzone description

2.1 Dynamics of robot system description

The dynamics of an n-link robot manipulator may be described by the nonlinear system [14]:

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) = \tau, \tag{1}$$

Received 13 March 2008; revised 13 November 2008.

where $q(t) \in \mathbb{R}^n$ is a vector of the link positions, $M(q)$ is the symmetric positive definite inertia matrix of robots, $V_m(q, \dot{q})$ represents coriolis and centrifugal forces, $G(q)$ is a vector that represents the gravitational forces, and τ is a vector of the torques (or forces) acted on the joints.

2.2 Deadzone nonlinearity

The deadzone usually appears as one form of the actuator nonlinearities. A mathematical model for the deadzone characteristics is given by:

$$D(u) = \begin{cases} g(u) < 0, & u \leqslant d_-, \\ 0, & d_- < d_+, \\ h(u) > 0, & u \geqslant d_+. \end{cases} \tag{2}$$

Function $h(u)$ and $g(u)$ are smooth, nonlinear functions, so this describes a very general class of deadzone. All of $h(u),\ g(u), d_+$ and d_- are unknown, so compensation is difficult. Here we assume that the functions $h(u)$ and $g(u)$ are monotonically increasing, smooth and invertible. Therefore, there exists a deadzone pre-inverse $D^{-1}(w)$,

$$D^{-1}(w) = \begin{cases} g^{-1}(w) < 0, & u < 0, \\ 0, & u = 0, \\ h^{-1}(w) > 0, & u > 0 \end{cases} \tag{3}$$

such that

$$D(D^{-1}(w)) = w. \tag{4}$$

To offset the deleterious effects of the unknown deadzone nonlinearity, a neural network compensator may be placed as illustrated in Fig.1, where w and u are the output of the NN controller and NN compensator, respectively. We hope that the compensator can cause the composite throughput from w to τ to be unity. This is the reason why the deadzone compensator using RBF networks is proposed in Section 4.

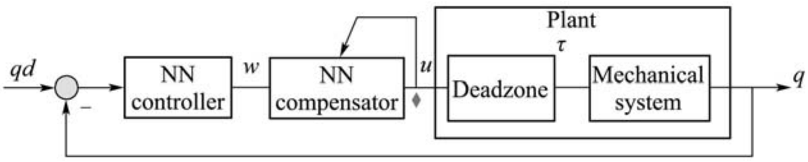

Fig. 1 Adaptive NN control robot system with actuator nonlinearities.

3 Adaptive NN controller design

3.1 GL matrix and operator

In this paper, we define GL matrix [8], denoted by "$\{\cdot\}$", and its product operator "$\circ$". Since the proofs in [6] can only prove the stability of one-link robot manipulator, GL matrix and its product operator are introduced to prove the stability of n-link robot manipulator.

Let I_0 be the set of integers, and $\theta_{i,j}, \xi_{ij} \in \mathbb{R}^{n_{ij}}$, $n_{ij} \in I_0,\ i = 1, \cdots, n,\ j = 1, \cdots, n$. For function approximation, θ_{ij} can be taken as the weight vector, and ξ_{ij} is the network basis function vector. The GL row vector $\{\theta_k\}$ and its transpose $\{\theta_k\}^{\mathrm{T}}$ are defined in the following way:

$$\{\theta_k\} = \{\theta_{k1}, \theta_{k2}, \cdots, \theta_{kn}\},$$
$$\{\theta_k\}^{\mathrm{T}} = \{\theta_{k1}^{\mathrm{T}}, \theta_{k2}^{\mathrm{T}}, \cdots, \theta_{kn}^{\mathrm{T}}\}.$$

The GL matrix $\{\theta\}$ and its transpose $\{\theta\}^{\mathrm{T}}$ are defined accordingly as

$$\{\theta\} = \left\{ \begin{matrix} \theta_{11} & \theta_{12} & \cdots & \theta_{1n} \\ \theta_{21} & \theta_{22} & \cdots & \theta_{2n} \\ \vdots & \vdots & & \vdots \\ \theta_{n1} & \theta_{n2} & \cdots & \theta_{nn} \end{matrix} \right\} = \left\{ \begin{matrix} \{\theta_1\} \\ \{\theta_2\} \\ \vdots \\ \{\theta_n\} \end{matrix} \right\},$$

$$\{\theta\}^{\mathrm{T}} = \left\{ \begin{matrix} \theta_{11}^{\mathrm{T}} & \theta_{12}^{\mathrm{T}} & \cdots & \theta_{1n}^{\mathrm{T}} \\ \theta_{21}^{\mathrm{T}} & \theta_{22}^{\mathrm{T}} & \cdots & \theta_{2n}^{\mathrm{T}} \\ \vdots & \vdots & & \vdots \\ \theta_{n1}^{\mathrm{T}} & \theta_{n2}^{\mathrm{T}} & \cdots & \theta_{nn}^{\mathrm{T}} \end{matrix} \right\}.$$

For a given GL matrix

$$\{\xi\} = \left\{ \begin{matrix} \xi_{11} & \xi_{12} & \cdots & \xi_{1n} \\ \xi_{21} & \xi_{22} & \cdots & \xi_{2n} \\ \vdots & \vdots & & \vdots \\ \xi_{n1} & \xi_{n2} & \cdots & \xi_{nn} \end{matrix} \right\} = \left\{ \begin{matrix} \{\xi_1\} \\ \{\xi_2\} \\ \vdots \\ \{\xi_n\} \end{matrix} \right\}.$$

The GL product of $\{\theta\}^{\mathrm{T}}$ and $\{\xi\}$ is an $n \times n$ matrix defined as:

$$[\{\theta\}^{\mathrm{T}} \circ \{\xi\}] = \begin{bmatrix} \theta_{11}^{\mathrm{T}}\xi_{11} & \theta_{12}^{\mathrm{T}}\xi_{12} & \cdots & \theta_{1n}^{\mathrm{T}}\xi_{1n} \\ \theta_{21}^{\mathrm{T}}\xi_{21} & \theta_{22}^{\mathrm{T}}\xi_{21} & \cdots & \theta_{2n}^{\mathrm{T}}\xi_{2n} \\ \vdots & \vdots & & \vdots \\ \theta_{n1}^{\mathrm{T}}\xi_{n1} & \theta_{n2}^{\mathrm{T}}\xi_{n2} & \cdots & \theta_{nn}^{\mathrm{T}}\xi_{nn} \end{bmatrix}.$$

The GL product of a square matrix and a GL row vector is defined as follows. Let $\varGamma_k = \varGamma_k^{\mathrm{T}} = [r_{k1}\ r_{k2} \cdots r_{kn}], r_{kj} \in \mathbb{R}^{m \times n_j}, m = \sum\limits_{j=1}^{n} n_j$, then we have

$$\begin{aligned} \varGamma_k \circ \{\xi_k\} &= \{\varGamma_k\} \circ \{\xi_k\} \\ &= \{r_{k1}\xi_{k1}\ r_{k2}\xi_{k2}\ \cdots\ r_{kn}\xi_{kn}\} \in \mathbb{R}^{m \times n}. \end{aligned}$$

Note that the GL product should be computed first in a mixed matrix product. For instance, in $\{A\} \circ \{B\}C$, the matrix $[\{A\} \circ \{B\}]$ should be computed first, and then followed by the multiplication of $[\{A\} \circ \{B\}]$ with matrix C.

3.2 Adaptive NN controller design

Given a desired robot arm trajectory $q_d(t) \in \mathbb{R}^n$, the tracking error is

$$e(t) = q_d(t) - q(t). \tag{5}$$

The filtered tracking error is

$$r = \dot{e} + \varLambda e, \tag{6}$$

where $\varLambda = \varLambda^{\mathrm{T}} > 0$ is a design parameter matrix, usually selected as diagonal matrix.

We define

$$\dot{q}_r = r + \dot{q}, \tag{7}$$
$$\ddot{q}_r = \dot{r} + \ddot{q}, \tag{8}$$
$$z = (q^{\mathrm{T}}\ \ \dot{q}^{\mathrm{T}})^{\mathrm{T}}. \tag{9}$$

So we have

$$\dot{q}_r = \dot{q}_d + \varLambda e, \tag{10}$$

$$\ddot{q}_r = \ddot{q}_d + \Lambda\dot{e}. \quad (11)$$

Based on the Gaussian RBF networks approximation property [8], one can emulate $M(q)$, $V_m(q,\dot{q})$ and $G(q)$, respectively:

$$M(q) = [\{W_m\}^{\mathrm{T}} \circ \{\sigma_M(q)\}] + \varepsilon_M(q), \quad (12)$$

$$V_m(q,\dot{q}) = [\{W_m\}^{\mathrm{T}} \circ \{\sigma_V(z)\}] + \varepsilon_V(q), \quad (13)$$

$$G(q) = [\{W_G\}^{\mathrm{T}} \circ \{\sigma_G(q)\}] + \varepsilon_G(q), \quad (14)$$

where $\{\circ\}$ is GL matrix and "$\circ$" is the GL product operator defined in Section 3.1. ε_v, ε_v and ε_G are the NN reconstruction errors. W_M, W_V and W_G are the ideal target weights. σ_M, σ_V and σ_G are all the radial basis function as $\sigma(x) = e^{\frac{-(x-c)^2}{p^2}}$. Then we can get

$$\begin{aligned} &M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + G(q) \\ &= [\{W_M\}^{\mathrm{T}} \circ \{\sigma_M(q)\}]\ddot{q} + [\{W_V\}^{\mathrm{T}} \circ \{\sigma_V(z)\}]\dot{q} \\ &\quad +[\{W_G\}^{\mathrm{T}} \circ \{\sigma_G(q)\}] + \varepsilon_M(q)\ddot{q} + \varepsilon_V(z)\dot{q} + \varepsilon_G(q) \\ &= [\{W_M\}^{\mathrm{T}} \circ \{\sigma_M(q)\}]\ddot{q}_r + [\{W_V\}^{\mathrm{T}} \circ \{\sigma_V(z)\}]\dot{q}_r \\ &\quad +[\{W_G\}^{\mathrm{T}} \circ \{\sigma_G(q)\}] - M\dot{r} - V_m(q,\dot{q})r + E, \quad (15)\end{aligned}$$

where $E = \varepsilon_M(q)\ddot{q}_r + \varepsilon_V(z)\dot{q}_r + \varepsilon_G(q)$.

By defining $\hat{W}_M$, $\hat{W}_V$ and $\hat{W}_G$ as estimates of the ideal NN weights, three RBF networks with parameter adaptation law can be used to emulate $M(q)$, $V_m(q,\dot{q})$ and $G(q)$, respectively:

$$\hat{M}(q) = \{\hat{W}_M\} \circ \{\sigma_M(q)\}, \quad (16)$$

$$\hat{V}_m(q,\dot{q}) = \{\hat{W}_V\} \circ \{\sigma_V(z)\}, \quad (17)$$

$$\hat{G}(q) = \{\hat{W}_G\} \circ \{\sigma_G(q)\}. \quad (18)$$

We choose the controller given by Ge et al. in [8] and [9]:

$$\begin{aligned} w &= \hat{M}(q)\ddot{q}_r + \hat{V}_m(q,\dot{q})\dot{q}_r + \hat{G}(q) + K_V r + K_r \mathrm{sgn}(r) \\ &= \{\hat{W}_M\}^{\mathrm{T}} \circ \{\sigma_M(q)\}\ddot{q}_r + \{\hat{W}_V\}^{\mathrm{T}} \\ &\quad \circ\{\sigma_V(z)\}\dot{q}_r + \{\hat{W}_G\}^{\mathrm{T}} \circ \{\sigma_G(q)\} \\ &\quad +K_V r + K_r \mathrm{sgn}(r). \quad (19)\end{aligned}$$

The adaptation laws are

$$\begin{cases} \dot{\hat{W}}_{Mk} = \Gamma_{Mk} \circ \sigma_{Mk}(q)\ddot{q}_r r_k, \\ \dot{\hat{W}}_{Vk} = \Gamma_{Vk} \circ \sigma_{Vk}(z)\dot{q}_r r_k, \\ \dot{\hat{W}}_{Gk} = \Gamma_{Gk} \circ \sigma_{Gk}(q) r_k. \end{cases} \quad (20)$$

4 NN deadzone compensator design

Based on the work of Lewis et al. [6], the deadzone inverse can be expressed in equivalent form as

$$D^{-1}(w) = w + w_{NN}, \quad (21)$$

where w_{NN} is the modified deadzone inverse and given by

$$w_{NN} = \begin{cases} g^{-1}(w) - w, & w < 0, \\ 0, & u = 0, \\ h^{-1}(w) - w, & w > 0. \end{cases} \quad (22)$$

It is defined that

$$u = w + \hat{w}_{NN}(w). \quad (23)$$

Based on the RBF network approximation property, one can approximate the deadzone function $D(u)$ and w_{NN}:

$$D(u) = \{W\}^{\mathrm{T}} \circ \{\sigma(u)\} + \varepsilon(u), \quad (24)$$

$$w_{NN} = \{W_i\}^{\mathrm{T}} \circ \{\sigma_i(w)\} + \varepsilon_i(w), \quad (25)$$

where $\varepsilon(u)$ and $\varepsilon_i(w)$ are the NN reconstruction error. W and W_i are the ideal target weights. σ and σ_i are both the radial basis function as $\sigma(x) = e^{\frac{-(x-c)^2}{p^2}}$.

By defining $\hat{W}$ and $\hat{W}_i$ as estimates of the ideal NN weights, two RBF networks are used to emulate $D(u)$ and w_{NN}:

$$\hat{D}(u) = \{\hat{W}\}^{\mathrm{T}} \circ \{\sigma(u)\}, \quad (26)$$

$$\hat{w}_{NN} = \{\hat{W}_i\}^{\mathrm{T}} \circ \{\sigma_i(w)\}. \quad (27)$$

We define $\tilde{W} = W - \hat{W}$ and $\tilde{W}_i = W_i - \hat{W}_i$ as the weight estimation error.

Theorem 1 Given the RBF network compensator (23) and (27), the throughput of the compensator plus the deadzone is given by

$$\begin{aligned}\tau = w &- [\{\hat{W}\}^{\mathrm{T}} \circ \{\sigma'(u)\}] \circ [\{\hat{W}_i\}^{\mathrm{T}} \circ \{\sigma_i(w)\}] \\ &+[\{\hat{W}\}^{\mathrm{T}} \circ \{\sigma'(u)\}] \circ [\{\tilde{W}_i\}^{\mathrm{T}} \circ \{\sigma_i(w)\}] \\ &+d(t), \quad (28)\end{aligned}$$

where the modeling mismatch term is

$$\begin{aligned} d(t) &= -[\{\tilde{W}\}^{\mathrm{T}} \circ \{\sigma'(u)\}] \circ [\{W_i\}^{\mathrm{T}} \circ \{\sigma_i(w)\}] \\ &\quad -b(t) + \varepsilon(u), \quad (29)\end{aligned}$$

$$\begin{aligned} b(t) &= \{W\}^{\mathrm{T}} \circ \{[\sigma'(w + \{\hat{W}_i\}^{\mathrm{T}}) \circ \{\sigma_i w\}] \\ &\quad \circ[\varepsilon_i(w)]\} + \{W\}^{\mathrm{T}} \circ \{R_1(\tilde{W}_i, w)\} \\ &\quad +\varepsilon(w + w_{NN}). \quad (30)\end{aligned}$$

Proof From (23) and (24), one has

$$\begin{aligned}\tau_k &= W_k^{\mathrm{T}}\sigma(u_k) + \varepsilon(u_k) \\ &= W_k^{\mathrm{T}}\sigma(w_k + \hat{w}_{NNk}) + \varepsilon(w_k + \hat{w}_{NNk}), \quad (31)\end{aligned}$$

where τ_k is the kth throughput of the compensator plus the deadzone.

From (4) and (25), it follows that

$$\begin{aligned} w_k &= W_k^{\mathrm{T}}\sigma(w_k + w_{NNk}) + \varepsilon(w_k + w_{NNk}) \\ &= W_k^{\mathrm{T}}\sigma(w_k + W_{ik}^{\mathrm{T}}\sigma_i(w_k) + \varepsilon_i(w_k)) \\ &\quad +\varepsilon(w_k + w_{NNk}) \\ &= W_k^{\mathrm{T}}\sigma(w_k + W_{ik}^{\mathrm{T}}\sigma_i(w_k) + \tilde{W}_{ik}^{\mathrm{T}}\sigma_i(w_k) + \varepsilon_i(w_k)) \\ &\quad +\sigma(w_k + w_{NNk}). \quad (32)\end{aligned}$$

Using the Taylor series expansion, one has

$$\begin{aligned} w_k &= W_k^{\mathrm{T}}\sigma(w_k + \hat{W}_{ik}^{\mathrm{T}}\sigma_i(w_k)) \\ &\quad +W_k^{\mathrm{T}}\sigma'(w_k + \hat{W}_{ik}^{\mathrm{T}}\sigma_i(w_k))(\tilde{W}_{ik}^{\mathrm{T}}\sigma_i(w_k) \\ &\quad +\varepsilon_i(w_k)) + W_k^{\mathrm{T}}R_1(\tilde{W}_{ik}, w_k) + \varepsilon(w_k + w_{NNk}), \quad (33)\end{aligned}$$

where $R_1(\tilde{W}_{ik}, w_k)$ is the remainder of the first Taylor polynomial. Regrouping the terms, one has

$$\begin{aligned} w_k &= W_k^{\mathrm{T}}\sigma(w_k + \hat{W}_{ik}^{\mathrm{T}}\sigma_i(w_k)) + W_k^{\mathrm{T}}\sigma' \\ &\quad (w_k + \hat{W}_{ik}^{\mathrm{T}}\sigma_i(w_k))\tilde{W}_{ik}^{\mathrm{T}}\sigma_i(w_k) + b_k(t), \quad (34)\end{aligned}$$

where

$$\begin{aligned} b_k(t) &= W_k^{\mathrm{T}}\sigma'(w_k + \hat{W}_{ik}^{\mathrm{T}}\sigma_i(w_k))\varepsilon_i(w_k) \\ &\quad +W_k^{\mathrm{T}}R_1(\tilde{W}_{ik}, w_k) + \varepsilon(w_k + w_{NNk}).\end{aligned}$$

Combining (23) and (24) gives

$$\begin{aligned} &w_k + \varepsilon(u_k) \\ &= W_k^{\mathrm{T}}\sigma(w_k + \hat{W}_{ik}^{\mathrm{T}}\sigma_i(w_k)) \\ &\quad +\varepsilon(u_k) + \hat{W}_k^{\mathrm{T}}\sigma'(w_k + \hat{W}_{ik}^{\mathrm{T}}\sigma_i(w_k))\tilde{W}_{ik}^{\mathrm{T}}\sigma_i(w_k) \\ &\quad +\tilde{W}_k^{\mathrm{T}}\sigma'(w_k + \hat{W}_{ik}^{\mathrm{T}}\sigma_i(w_k))\tilde{W}_{ik}^{\mathrm{T}}\sigma_i(w_k) + b_k(t) \\ &= W_k^{\mathrm{T}}\sigma(u_k) + \varepsilon(u_k) + \hat{W}_k^{\mathrm{T}}\sigma'(u_k)\tilde{W}_{ik}^{\mathrm{T}}\sigma_i(w_k)\end{aligned}$$

$$+\tilde{W}^{\mathrm{T}}\sigma\prime(u_k)\tilde{W}_{ik}^{\mathrm{T}}\sigma_i(w_k)+b_k(t)$$
$$= W_k^{\mathrm{T}}\sigma(u_k)+\varepsilon(u_k)+\hat{W}_k^{\mathrm{T}}\sigma'(u_k)\tilde{W}_{ik}^{\mathrm{T}}\sigma_i(w_k)$$
$$+\tilde{W}_k^{\mathrm{T}}\sigma(u_k)W_{ik}^{\mathrm{T}}\sigma_i(w_k)-\tilde{W}_k^{\mathrm{T}}\sigma'(u_k)\hat{W}_{ik}^{\mathrm{T}}\sigma_i(w_k)$$
$$+b_k(t), \tag{35}$$

which combined with (31) gives

$$\tau_k = w_k - \hat{W}_k^{\mathrm{T}}\sigma'(u_k)\tilde{W}_{ik}^{\mathrm{T}}\sigma_i(w_k)$$
$$+\tilde{W}_k^{\mathrm{T}}\sigma'(u_k)\hat{W}_{ik}\sigma_i(w_k)+d_k(t).$$

Using the GL matrix and product operator defined in section 3, one can get (28).

Theorem 2 The norm of the modeling mismatching term $d(t)$ in (29) is bounded on a compact set by

$$\|d(t)\| \leqslant a_1\|\tilde{W}\|_F + a_2\|\tilde{W}_i\|_F^2$$
$$+a_3\|\tilde{W}_i\|_F + a_5. \tag{36}$$

The proof is in [15] and [16].

Theorem 2 gives us the upper bound of the norm of $d(t)$, it is an important result in the stability proof of the next section. The NN tuning algorithm is chosen as

$$\begin{cases}\dot{\hat{W}}_k = -S\sigma'(u_k)\hat{W}_{ik}^{\mathrm{T}}\sigma_i(w_k)r_k - K_1 S\|r\|\hat{W}_k,\\ \dot{\hat{W}}_{ik} = T\sigma_i(w_k)r_k\hat{W}_k^{\mathrm{T}}\sigma'(u_k) - K_1 T\|r\|\hat{W}_{ik}\\ \quad -K_2 T\|r\|\|\hat{W}_i\|_F\hat{W}_{ik}\end{cases} \tag{37}$$

with any positive constant matrices $S = S^{\mathrm{T}}, T = T^{\mathrm{T}}$ and $K_1, K_2 > 0$. $\hat{W}_k$ and $\hat{W}_{ik}$ are the weights of the kth compensator and $1 \leqslant k \leqslant n$.

5 Adaptive NN controller with deadzone compensator

Based on the dynamics of robot manipulator equation (1) and the definitions in equation (5)~(8) in Section 3, there exists

$$M\dot{r} = -V_m r - \tau + f, \tag{38}$$

where $f = M\ddot{q}_r + V_m\dot{q}_r + G$.

From the adaptive NN controller (19), it can be written as

$$w = \hat{f} + K_v r, \tag{39}$$

where

$$\hat{f} = \{\hat{W}_M\}^{\mathrm{T}} \circ \{\sigma_M(q)\}\ddot{q}_r + \{\hat{W}_V\}^{\mathrm{T}} \circ \{\sigma_V(z)\}\dot{q}_r$$
$$+\{\hat{W}_G\}^{\mathrm{T}} \circ \{\sigma_G(q)\} + K_r\mathrm{sgn}(r).$$

Combining (28) and (38) gives

$$M\dot{r} = V_m r - K_V r + [\{\hat{W}\}^{\mathrm{T}} \circ \{\sigma'(u)\}] \circ [\{\tilde{W}_i\}^{\mathrm{T}}$$
$$\circ\{\sigma_i(w)\}] - [\{\tilde{W}\}^{\mathrm{T}} \circ \{\sigma'(u)\}] \circ [\{\hat{W}_i\}^{\mathrm{T}}$$
$$\circ\{\sigma_i(w)\}] - d(t) + f - \hat{f}. \tag{40}$$

Noting that

$$f - \hat{f} = \{\tilde{W}_M\}^{\mathrm{T}} \circ \{\sigma_M(q)\}\ddot{q}_r + \{\tilde{W}_V\}^{\mathrm{T}}$$
$$\circ \{\sigma_V(z)\}\dot{q}_r + \{\tilde{W}_G\}^{\mathrm{T}} \circ \{\sigma_G(q)\}$$
$$+E - K_r\mathrm{sgn}(r). \tag{41}$$

Theorem 3 Given the system in (1), assuming that the ideal weights W and W_i bounded as $\|W\|_F \leqslant W_M$ and $\|W_i\|_F \leqslant W_{iM}$, select the tracking control law (19) plus the deadzone compensator (23). Let NN weights update laws be provided by (20) and (37). Then the tracking error $r(t)$ is uniformly ultimately bounded, with bounds given by (56). Moreover, the tracking error $r(t)$ may be kept as small as desired by increasing the feedback gain $K_V > 0$.

Proof Select the Lyapunov function candidate

$$L = \frac{1}{2}r^{\mathrm{T}}Mr + \frac{1}{2}\sum_{k=1}^{n}\tilde{W}_k^{\mathrm{T}}S^{-1}\tilde{W}_k$$
$$+\frac{1}{2}\sum_{k=1}^{n}\tilde{W}_{ik}^{\mathrm{T}}T^{-1}\tilde{W}_{ik} + \frac{1}{2}\sum_{k=1}^{n}\tilde{W}_{Mk}^{\mathrm{T}}\varGamma_{Mk}^{-1}\tilde{W}_{Mk}$$
$$+\frac{1}{2}\sum_{k=1}^{n}\tilde{W}_{Vk}^{\mathrm{T}}\varGamma_{Vk}^{-1}\tilde{W}_{Vk} + \frac{1}{2}\sum_{k=1}^{n}\tilde{W}_{Gk}^{\mathrm{T}}\varGamma_{Gk}^{-1}\tilde{W}_{Gk}. \tag{42}$$

Differentiating L and using the robot property $\dot{M} - 2V_m$ being skew-symmetric matrix yields

$$\dot{L} = r^{\mathrm{T}}(M\dot{r} + V_m r) + \sum_{k=1}^{n}\tilde{W}_k^{\mathrm{T}}S^{-1}\dot{\tilde{W}}_k$$
$$+\sum_{k=1}^{n}\tilde{W}_{ik}^{\mathrm{T}}T^{-1}\dot{\tilde{W}}_{ik} + \sum_{k=1}^{n}\tilde{W}_{Mk}^{\mathrm{T}}\varGamma_{Mk}^{-1}\dot{\tilde{W}}_{Mk}$$
$$+\sum_{k=1}^{n}\tilde{W}_{Vk}^{\mathrm{T}}\varGamma_{Vk}^{-1}\dot{\tilde{W}}_{Vk} + \sum_{k=1}^{n}\tilde{W}_{Gk}^{\mathrm{T}}\varGamma_{Gk}^{-1}\dot{\tilde{W}}_{Gk}. \tag{43}$$

Using (40) and (41), one has

$$\dot{L} = r^{\mathrm{T}}[-K_V r + [\{\hat{W}\}^{\mathrm{T}} \circ \{\sigma'(u)\}]$$
$$\circ [\{\tilde{W}_i\}^{\mathrm{T}} \circ \{\sigma_i(w)\}] - [\{\tilde{W}\}^{\mathrm{T}} \circ \{\sigma'(u)\}]$$
$$\circ [\{\hat{W}_i\}^{\mathrm{T}} \circ \{\sigma_i(w)\}] - d(t)$$
$$+\{\tilde{W}_V\}^{\mathrm{T}} \circ \{\sigma_V(q)\}\ddot{q}_r + \{\tilde{W}_M\}^{\mathrm{T}} \circ \{\sigma_M(q)\}\dot{q}_r$$
$$+\{\tilde{W_G}\}^{\mathrm{T}} \circ \{\sigma_G(q)\} + E - K_r\mathrm{sgn}(r)]$$
$$+\sum_{k=1}^{n}\tilde{W}_k^{\mathrm{T}}S^{-1}\dot{\tilde{W}}_k + \sum_{k=1}^{n}\tilde{W}_{ik}^{\mathrm{T}}T^{-1}\dot{\tilde{W}}_{ik}$$
$$+\sum_{k=1}^{n}\tilde{W}_{Mk}^{\mathrm{T}}\varGamma_{Mk}^{-1}\dot{\tilde{W}}_{Mk} + \sum_{k=1}^{n}\tilde{W}_{Vk}^{\mathrm{T}}\varGamma_{Vk}^{-1}\dot{\tilde{W}}_{Vk}$$
$$+\sum_{k=1}^{n}\tilde{W}_{Gk}^{\mathrm{T}}\varGamma_{Gk}^{-1}\dot{\tilde{W}}_{Gk}. \tag{44}$$

By noting that

$$\begin{cases}r^{\mathrm{T}}[\{\tilde{W}_M\}^{\mathrm{T}} \circ \{\sigma_M(q)\}]\ddot{q}_r\\ = \sum_{k=1}^{n}\{\tilde{W}_{Mk}\}^{\mathrm{T}} \circ \{\sigma_{Mk}(q)\}\ddot{q}_r r_k,\\ r^{\mathrm{T}}\{\tilde{W}_V\}^{\mathrm{T}} \circ \{\sigma_V(z)\}\dot{q}_r\\ = \sum_{k=1}^{n}\{\tilde{W}_{Vk}\}^{\mathrm{T}} \circ \{\sigma_{Vk}(q)\}\dot{q}_r r_k,\\ r^{\mathrm{T}}\{\tilde{W}_G\}^{\mathrm{T}} \circ \{\sigma_G(q)\} = \sum_{k=1}^{n}\tilde{W}_{Gk}\sigma_{Gk}(q)r_k.\end{cases} \tag{45}$$

And using the adaptation laws of controller (20), one has

$$\dot{L} = r^{\mathrm{T}}[-K_V r + [\{\hat{W}\}^{\mathrm{T}} \circ \{\sigma'(u)\}]$$
$$\circ [\{\tilde{W}_i\}^{\mathrm{T}} \circ \{\sigma_i(w)\}] - [\{\hat{W}\}^{\mathrm{T}} \circ \{\sigma'(u)\}]$$
$$\circ [\{\tilde{W}_i\}^{\mathrm{T}} \circ \{\sigma_i(w)\}] - d(t) + E - K_r\mathrm{sgn}(r)$$
$$+\sum_{k=1}^{n}\tilde{W}_k^{\mathrm{T}}S^{-1}\dot{\tilde{W}}_k + \sum_{k=1}^{n}\tilde{W}_{ik}^{\mathrm{T}}T^{-1}\dot{\tilde{W}}_{ik}$$
$$= r^{\mathrm{T}}[-K_V r - K_r\mathrm{sgn}(r) + E] - r^{\mathrm{T}}[\{\tilde{W}\}^{\mathrm{T}}$$
$$\circ \{\sigma'(u)\}] \circ [\{\hat{W}_i\}^{\mathrm{T}} \circ \{\sigma_i(w)\}] + r^{\mathrm{T}}[\{\hat{W}\}^{\mathrm{T}} \circ \{\sigma'(u)\}]$$
$$\circ [\{\tilde{W}_i\}^{\mathrm{T}} \circ \{\sigma_i(w)\}] + \sum_{k=1}^{n}\tilde{W}_k^{\mathrm{T}}S^{-1}\dot{\tilde{W}}_k$$
$$+\sum_{k=1}^{n}\tilde{W_{ik}}^{\mathrm{T}}T^{-1}\dot{\tilde{W}}_{ik} - r^{\mathrm{T}}d(t). \tag{46}$$

Applying the compensator NN tuning rules (37), one may write

$$\dot{L}=r^{\mathrm{T}}[-K_V r-K_r\mathrm{sgn}(r)+E] -r^{\mathrm{T}}[\{\tilde{W}\}^{\mathrm{T}}\circ\{\sigma'(u)\}]\circ[\{\hat{W}_i\}^{\mathrm{T}}\circ\{\sigma_i(w)\}] +r^{\mathrm{T}}[\{\hat{W}\}^{\mathrm{T}}\circ\{\sigma'(u)\}]\circ[\{\tilde{W}_i\}^{\mathrm{T}}\circ\{\sigma_i(w)\}] +\sum_{k=1}^{n}\tilde{W}_k(\sigma_k'(u)\hat{W}_{ik}^{\mathrm{T}}\sigma_{ik}(w)r_k+K_1\|r\|\hat{W}_k) +\sum_{k=1}^{n}\tilde{W}_{ik}^{\mathrm{T}}(-\sigma_{ik}(w)r_k\hat{W}_k^{\mathrm{T}}\sigma_k'(u)+K_1\|r\|\hat{W}_{ik} +K_2\|r\|\|\hat{W}_i\|_F\hat{W}_{ik})-r^{\mathrm{T}}d(t). \tag{47}$$

By noting that

$$\begin{cases} r^{\mathrm{T}}[\{\tilde{W}\}^{\mathrm{T}}\circ\{\sigma'(u)\}]\circ[\{\hat{W}_i\}^{\mathrm{T}}\circ\{\sigma_i(w)\}] \\ =\sum_{k=1}^{n}\tilde{W}_k\sigma_k'(u)\hat{W}_{ik}^{\mathrm{T}}\sigma_{ik}(w)r_k, \\ r^{\mathrm{T}}[\{\hat{W}\}^{\mathrm{T}}\circ\{\sigma'(u)\}]\circ[\{\tilde{W}_i\}^{\mathrm{T}}\circ\{\sigma_i(w)\}] \\ =\sum_{k=1}^{n}\tilde{W}_{ik}^{\mathrm{T}}\sigma_{ik}(w)r_k\hat{W}_k^{\mathrm{T}}\sigma_k'(u). \end{cases} \tag{48}$$

Therefore,

$$\dot{L}=r^{\mathrm{T}}[-K_V r-K_r\mathrm{sgn}(r)+E]+\sum_{k=1}^{n}\tilde{W}_k^{\mathrm{T}}K_1\|r\|\hat{W}_k +\sum_{k=1}^{n}\tilde{W}_{ik}^{\mathrm{T}}(K_1\|r\|\hat{W}_{ik}+K_2\|r\|\|\hat{W}_i\|_F\hat{W}_{ik})-r^{\mathrm{T}}d(t) =r^{\mathrm{T}}[-K_V r-K_r\mathrm{sgn}(r)+E]+K_1\|r\|\mathrm{tr}[\tilde{W}^{\mathrm{T}}\hat{W}] +\|r\|\mathrm{tr}[\tilde{W}_i^{\mathrm{T}}K_1\hat{W}_i+\tilde{W}_i^{\mathrm{T}}K_2\|\hat{W}_i\|_F\hat{W}_i]-r^{\mathrm{T}}d(t) =r^{\mathrm{T}}[-K_V r-K_r\mathrm{sgn}(r)+E]+K_1\|r\|\mathrm{tr}[\tilde{W}^{\mathrm{T}} (W-\tilde{W}]+\|r\|\mathrm{tr}[\tilde{W}_i^{\mathrm{T}}K_1(W_i-\tilde{W}) +\tilde{W}_i^{\mathrm{T}}K_2\|\hat{W}_i\|_F(W_i-\tilde{W}_i)]-r^{\mathrm{T}}d(t). \tag{49}$$

By choosing $K_r\geqslant\|E\|$,

$$\dot{L}\leqslant -r^{\mathrm{T}}K_V r+K_1\|r\|\mathrm{tr}[\tilde{W}^{\mathrm{T}}(W-\hat{W})] +\|r\|\mathrm{tr}[\tilde{W}_i^{\mathrm{T}}K_1(W_i-\tilde{W}_i)+\tilde{W}_i^{\mathrm{T}}K_2\|\hat{W}_i\|_F (W_i-\tilde{W}_i)]-r^{\mathrm{T}}d(t). \tag{50}$$

Using the inequality $\mathrm{tr}[\tilde{x}^{\mathrm{T}}(x-\tilde{x})]\leqslant\|\tilde{x}\|_F\|x\|_F-\|\tilde{x}\|_F^2$ and (36), one may write

$$\dot{L}\leqslant -K_{V_{\min}}\|r\|^2+K_1\|r\|\|\tilde{W}\|_F(W_M-\|\tilde{W}\|_F) +K_1\|r\|\|\tilde{W}_i\|_F(W_{iM}-\|\tilde{W}_i\|_F) +K_2\|r\|\|\tilde{W}_i\|_F\|W_i-\tilde{W}_i\|_F(W_{iM}-\|\tilde{W}_i\|_F) +\|r\|(a_1\|\tilde{W}\|_F+a_2\|\tilde{W}_i\|_F^2+a_3\|\tilde{W}_i\|_F+a_5),$$

$$\dot{L}\leqslant -K_{V_{\min}}\|r\|^2+K_1\|r\|\|\tilde{W}\|_F(W_M-\|\tilde{W}\|_F) +K_1\|r\|\|\tilde{W}_i\|_F(W_{iM}-\|\tilde{W}_i\|_F)+K_2\|r\|\|\tilde{W}_i\|_F\|W_i -\tilde{W}_i\|_F W_{iM}-K_2\|r\|\|\tilde{W}_i\|_F^2\|W_i-\tilde{W}_i\|_F +\|r\|(a_1\|\tilde{W}\|_F+a_2\|\tilde{W}_i\|_F^2+a_3\|\tilde{W}_i\|_F+a_5),$$

$$\dot{L}\leqslant -K_{V_{\min}}\|r\|^2+K_1\|r\|\|\tilde{W}\|_F(W_M-\|\tilde{W}\|_F) +K_1\|r\|\|\tilde{W}_i\|_F(W_{iM}-\|\tilde{W}_i\|_F) +K_2\|r\|\|\tilde{W}_i\|_F\|W_{iM}^2+2K_2\|r\|\|\tilde{W}_i\|_F^2W_{iM} -K_2\|r\|\|\tilde{W}_i\|_F^3+\|r\|(a_1\|\tilde{W}\|_F+a_2\|\tilde{W}_i\|_F +a_3\|\tilde{W}_i\|_F+a_5),$$

$$\dot{L}\leqslant -\|r\|\{K_{V_{\min}}\|r\|-K_1\|\tilde{W}\|_F(W_M-\|\tilde{W}\|_F) -K_1\|\tilde{W}_i\|_F(W_{iM}-\|\tilde{W}_i\|_F)-K_2\|\tilde{W}_i\|_F\|W_{iM}^2 -2K_2\|\tilde{W}_i\|_F^2W_{iM}+K_2\|\tilde{W}_i\|_F^3-a_1\|\tilde{W}\|_F -a_2\|\tilde{W}_i\|_F-a_3\|\tilde{W}_i\|_F-a_5\}, \tag{51}$$

$$\dot{L}\leqslant -\|r\|\{K_{V_{\min}}\|r\|+K_1\|\tilde{W}\|_F^2-(K_1W+a_1)\|\tilde{W}\|_F +K_2\|\tilde{W}_i\|_F^3+(K_1-2K_2W_{iM}-a_2)\|\tilde{W}_i\|_F^2 -(K_1W_{iM}+K_2W_{iM}^2+a_3)\|\tilde{W}_i\|_F-a_5\},$$

$$\dot{L}\leqslant -\|r\|\{K_{V_{\min}}\|r\|+K_1[\|\tilde{W}\|_F -\frac{1}{2}(W_M+\frac{a_1}{K_1})]^2-\frac{1}{4}K_1(W_M+\frac{a_1}{K_1})^2 +g(\|\tilde{W}_i\|_F)-a_5\}, \tag{52}$$

where the function $g(x)$ is defined as

$$g(x)=K_2x^3+(K_1-2K_2W_{iM}-a_2)x^2 -(K_1W_{iM}+K_2W_{iM}^2+a_3)x. \tag{53}$$

Let the constant C be defined as

$$C=\inf\{g(x),x\geqslant 0\}. \tag{54}$$

Defining $h(x)=g(x)+C$, then

$$\dot{L}\leqslant -\|r\|\{K_{V_{\min}}\|r\|+K_1[\|\tilde{W}\|_F -\frac{1}{2}(W_M+\frac{a_1}{K_1})]^2-\frac{1}{4}K_1(W_M+\frac{a_1}{K_1})^2 +h(\|\tilde{W}_i\|_F)-C-a_5\}. \tag{55}$$

Therefore, $\dot{L}$ is guaranteed to be negative as long as

$$\|r\|\geqslant\frac{\frac{1}{4}K_1(W_M+\frac{a_1}{K_1})^2+C+a_5}{K_{V_{\min}}}. \tag{56}$$

Remark 1 The right-hand side of (56) can be taken as a practical bound on the tracking error in the sense that $r(t)$ will never stray far above it. Note that the tracking error $r(t)$ may be kept as small as desired by increasing the feedback gain $K_V>0$. Therefore, considering the controller and the compensator as an integral control part of the robot, the closed-loop system is proved to be uniformly ultimately bounded (UUB).

Remark 2 Three RBF networks are used to emulate the inertia matrix, centripetal matrix and gravitational vector of the robot directly. Nonlinear robot function $f(x)$ estimate in [6] is not necessary. Chattering caused by switch gain can be decreased effectively.

Remark 3 In this paper, the closed-loop system of n-link robot manipulator is proved to be uniformly ultimately bounded (UUB) by the help of the GL matrix and operator.

6 Simulation tests

For the simulation studies we consider a two-link robot manipulator as Fig.2, whose dynamics are described by

$$M(q)\ddot{q}+V_m(q,\dot{q})\dot{q}+G(q)=\tau, \tag{57}$$

where

$$M(q)=\begin{bmatrix} p_1+p_2+2p_3\cos q_2 & p_2+p_3\cos q_2 \\ p_2+p_3\cos q_2 & p_2 \end{bmatrix},$$

$$V_m(q,\dot{q})=\begin{bmatrix} -p_3\dot{q}_2\sin q_2 & -p_3(\dot{q}_1+\dot{q}_2)\sin q_2 \\ p_3\dot{q}_1\sin q_2 & 0 \end{bmatrix},$$

$$G(q)=\begin{bmatrix} p_4\cos q_1+p_5\cos(q_1+q_2) \\ p_5\cos(q_1+q_2) \end{bmatrix}.$$

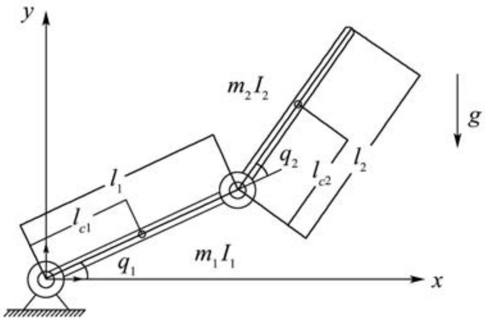

Fig. 2 Planar two-link manipulator.

We get equation (57) from [8], and the meaning of $p_1 \sim p_5$ are described as follows:

$$p_1 = m_1 l_{c1}^2 + m_2 l_1^2 + I_1,$$
$$p_2 = m_2 l_{c2}^2 + I_2,$$
$$p_3 = m_2 l_1 l_{c2},$$
$$p_4 = m_1 l_{c2} + m_2 l_1,$$
$$p_5 = m_2 l_{c2},$$

where m_i and l_i are the mass and length of link i, l_{ci} is the distance from joint $(i-1)$ to the centre of mass of link i, and I_i is the moment of inertia of link i.

The true parameters of the robot used for simulation are:

$$p = [p_1, p_2, p_3, p_4, p_5] = [2.9, 0.76, 0.87, 29.8, 8.5]\text{kg}\cdot\text{m}^2.$$

Suppose that the robot initially all rests at zero rad. The deadzone is assumed to have linear functions outside the deadband. We select

$$d_+ = 12,\ \ d_- = -10,$$
$$h(u) = u - d_+,\ \ g(u) = u + d_-.$$

The desired trajectory is chosen as follows:

$$q_d(t) = \begin{bmatrix} 2\sin(0.2\pi t) \\ \cos(0.2\pi t) \end{bmatrix}.$$

The parameters of adaptive NN controller are chosen as

$$\Lambda = \Lambda^{\mathrm{T}} = \mathrm{diag}[5.0],\ \ K_V = 20, K_r = 1.0 \quad (58)$$

$$\begin{cases} \Gamma_{Mk} = \mathrm{diag}[0.05], \\ \Gamma_{Vk} = \mathrm{diag}[0.01], \\ \Gamma_{Gk} = \mathrm{diag}[10]. \end{cases} \quad (59)$$

The centers of the RBF networks are uniformly distributed based on the range of the $q_d, \dot{q}_d$. The weights of the RBF controller are all initialized at zero.

6.1 The comparison of RBF network and BP-network

The neural network used for compensation in [6] is a modified multilayer BP-network. There were too many parameters to be initialized properly without difficulty and no principles for selecting parameters. However, using RBF networks only needs several parameters.

From (51) we know that small K_1 and K_2 can help guarantee the stability of the closed-loop system. Therefore, we choose $K_1 = K_2 = 0.0001$. We select $S = T = 200$. It is enough to choose 5 hidden-layer nodes for each robot link. The centers of the RBF networks are chosen as uniformly distributed based on the range of controller output. The weights $\hat{W}$ are all initialized at 1.0 and weights $\hat{W}_i$ are all initialized at zero.

Different from the modified multilayer BP-network proposed in [6], RBF networks have fewer parameters and become much easier to realize.

6.2 PD controller without compensation

Let us first investigate the control performance when we only use the PD control without deadzone compensation. In this case, the position errors of the robot two links and the corresponding control signals τ are shown in Figs.3 and 4.

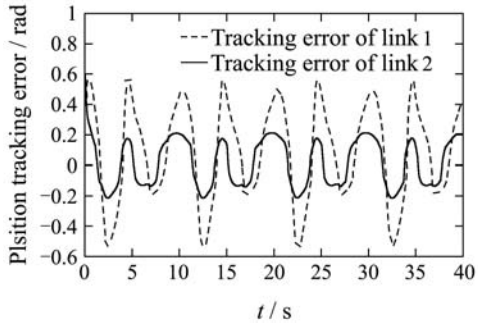

Fig. 3 Position tracking errors with PD control.

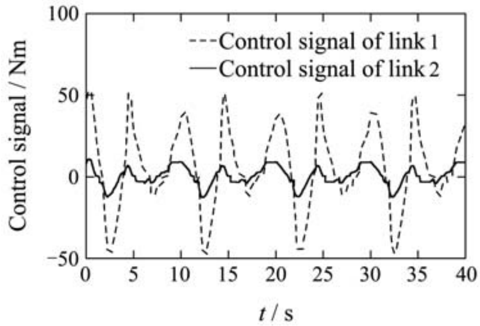

Fig. 4 Control signals τ.

It can be seen that the simple PD control has large tracking error and cannot handle the actuator nonlinearities such as deadzone.

6.3 Adaptive controller without compensator

In this case, the adaptive controller is adopted but no compensator is added. The position errors of the robotic two links and the corresponding control signals τ are shown in Figs.5 and 6.

It can be seen that the tracking errors are much smaller than that of the only PD case, but they cannot satisfy the high accuracy-tracking requirement, since only the adaptive controller cannot eliminate the effects of actuator nonlinearities.

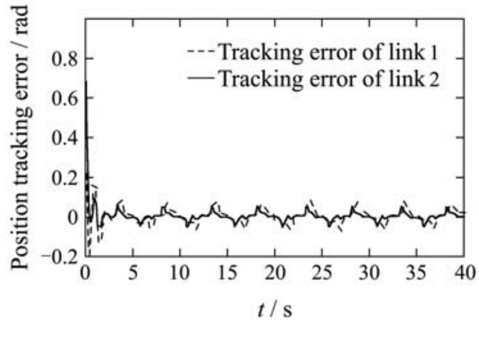

Fig. 5 Position tracking errors only using the adaptive controller.

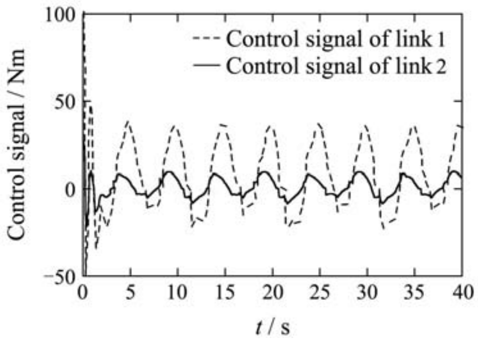

Fig. 6 Control signals τ.

6.4 Adaptive controller with compensation

In this case, the adaptive controller is adopted and the deadzone compensator is also added. The position errors of the robot is two links and the corresponding control signals τ are shown in Figs.7∼10.

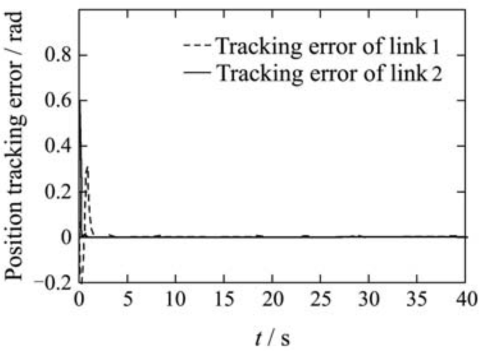

Fig. 7 Position tracking errors when using the adaptive controller and the compensator.

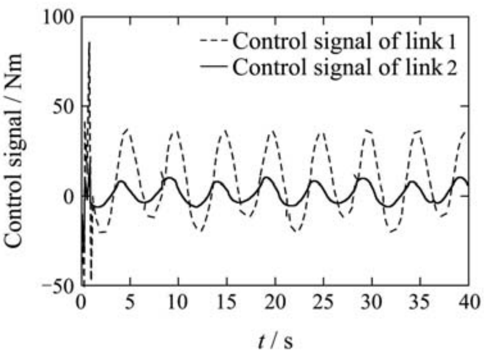

Fig. 8 Control signals τ.

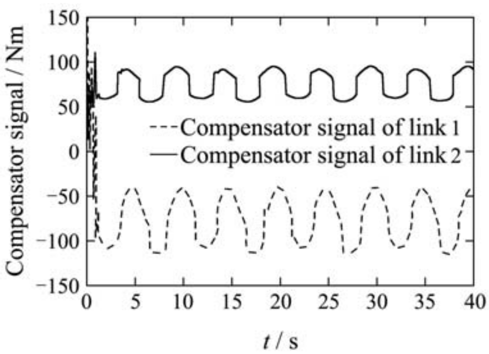

Fig. 9 Compensator signals $\hat{W}_{NN}$.

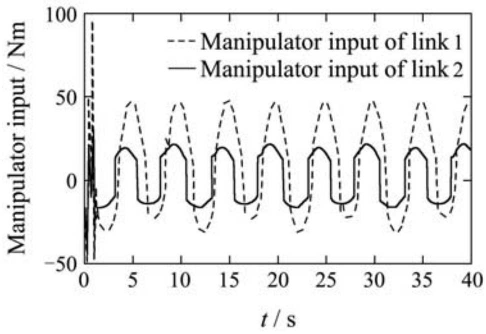

Fig. 10 Manipulator input u.

It can be seen that the tracking errors are smaller than those of the above two cases. Since the RBF networks compensator eliminates the effects of the actuator nonlinearity, the deadzone has almost no influence on the control signal τ as illustrated in Fig.7. Therefore, the robot manipulator with actuator nonlinearities can achieve the high accuracy-tracking purpose. In addition, an interesting feature can be seen from the Fig.8, the RBF compensator signals $\hat{W}_{NN}$ not only provide the compensation in the feedforward path, but also take part in the function of the adaptive controller. Therefore, combining the NN compensator with the adaptive NN controller as an integral control part of the robot makes the whole tracking system more robust and accurate.

7 Conclusions

In this paper, an adaptive neural network control scheme based on RBF neural network for a robot manipulator with deadzone is proposed. The control scheme consists of a new adaptive RBF neural network controller and an improved RBF compensator. Unlike the NN control proposed in [6], three RBF networks with parameter adaptive law are used to emulate the inertia matrix, centripetal matrix and gravitational vector of robot directly, nolinear robot function $f(x)$ estimate is not necessary, and chattering caused by switch gain can be decreased effectively. Compared to the multilayer BP-network in [6], RBF-network used in the improved control system has fewer parameters to be tuned and is much easier to realize in real time control engineering. For example, for BP neural network in [6], 20 hidden-layer nodes with sigmoid function and 4 jump function are chosen; in this paper, we only use 5 hidden layer nodes. GL matrix and operator are introduced to help prove the stability of the n-link robot manipulator system. The closed-loop system is proved to be uniformly ultimately bounded (UUB). The whole scheme can not only be used for deadzone compensation, but also for general actuator nonlinearities compensation, like backlash, saturation, etc. Simulation results verify the high accuracy-tracking performance of the designed scheme and the theoretical discussion.

In addition, the similar analysis and Matlab programs design were given in the book [16].

References

[1] B. Friedl. *Advanced Control System Design*[M]. Englewood Cliffs: Prentice-Hall, 1996.

[2] G. Tao, P. V. Kokotovic. Adaptive control of systems with unknown output backlash[J]. *IEEE Transactions on Automatic Control*, 1995, 40(2): 326 – 330.

[3] G. Tao, P. V. Kokotovic. Continuous-time adaptive control of systems with unknown backlash[J]. *IEEE Transactions on Automatic Control*, 1995, 40(6): 1083 – 1087.

[4] F. L. Lewis, W. K. Tim, L. Wang, et al. Deadzone compensation in motion control systems using adaptive fuzzy logic control[J]. *IEEE Transactions on Control Systems and Technology*, 1999, 7(6): 731 – 742.

[5] G. Tao, P. V. Kokotovic. *Adaptive Control of Systems with Actuator and Sensor Nonlinearities*[M]. New York: John Wiley & Sons, 1996.

[6] R. R. Selmic, F. L. Lewis. Deadzone compensation in motion control systems using neural networks[J]. *IEEE Transactions on Automatic Control*, 2000, 45(4): 602 – 612.

[7] M. Bianchini, P. Frasconi, M. Gori. Learning without local minima in radial basis function networks[J]. *IEEE Trans. Neural Networks*, 1995, 6(3): 749 – 755.

[8] S. S. Ge, T. H. Lee, C. J. Harris. *Adaptive Neural Network Control of Robotic Manipulators*[M]. Singapore: World Scientific Publishing Co., 1998.

[9] S. S. Ge, C. C. Hang, L. C. Woon. Adaptive neural network control of robot manipulators in task space[J]. *IEEE Trans. Ind. Electron*, 1997, 44(6): 746 – 752.

[10] S. S. Ge, C. C. Hang. Direct adaptive neural network control of robots[J]. *International Journal of System Science*, 1996, 27(6): 533 – 542.

[11] Q. Zhu, S. Fei, T. Zhang, et al. Adaptive RBF neural-networks control for a class of time-delay nonlinear systems [J]. *Neurocomputing*, 2008, 71(16/18): 3617 – 3624.

[12] H. Peng, J. Wu, G. Inoussa, et al. Nonlinear system modeling and predictive control using the RBF nets-based quasi-linear ARX model[J]. *Control Engineering Practice*, 2009, 17(1): 59 – 66.

[13] X. Wu, X. Zhu, G. Cao, et al. Predictive control of SOFC based on a GA-RBF neural network model[J].*Journal of Power Sources*, 2008, 179(1): 232 – 239.

[14] J. E. Slotine, W. Li. *Applied Nonlinear Control*[M]. Englewood Cliffs: Prentice-Hall, 1991.

[15] Y. Lu, J. Liu, F. Sun. Actuator nonlinearities compensation using rbf neural networks in robot control system[C]//*IMACS Multiconference on Computational Engineering in Systems Applications*. Beijing: Tsinghua University Press, 2006: 231 – 238.

[16] J. Liu. *Robot Control System Design and Matlab Simulation*[M]. Beijing: Tsinghua University Press, 2008.

Jinkun LIU was born in Heilongjiang Province, 1965. He received his Ph.D. degree in Department of Automatic Control from Northeastern University, China, in 1997. After a two-year post-doctor in Zhejiang University from 1997 to 1999, he joined School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, and he is currently a professor. His research concerns motion control, intelligent control, and robust control. E-mail: ljk@buaa.edu.cn.

Yu LU received the B.E. and M.E. degrees in Electrical Engineering from the Beijing University of Aeronautics and Astronautics, China. Since 2008, he has been pursuing the Ph.D. degree at the National University of Singapore. His current research interests are in the fields of computer networks, human-computer interaction and applications of control theory.