2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Adaptive Neural Network Tracking Control of Manipulators Using Quaternion Feedback

Long Cheng, Zeng-Guang Hou, and Min Tan

Abstract— **An adaptive neural network controller is proposed to deal with the task-space tracking problem of manipulators with kinematic and dynamic uncertainties. The orientation of manipulator is represented by the unit quaternion, which avoids singularities associated with three-parameter representation. By employing the adaptive Jacobian scheme, neural networks, and backstepping technique, the torque controller is obtained which is demonstrated to be stable by the Lyapunov approach. The adaptive updating laws for controller parameters are derived by the projection method, and the tracking error can be reduced as small as desired. The favorable features of the proposed controller lie in that: (1) the uncertainty in manipulator kinematics is taken into account; (2) the unit quaternion is used to represent the end-effector orientation; (3) the "linearity-in-parameters" assumption for the uncertain terms in dynamics of manipulators is no longer necessary; (4) effects of external disturbances are also considered in the controller design. Finally, the satisfactory performance of the proposed approach is illustrated by simulation results on a PUMA 560 robot.**

I. INTRODUCTION

Most research so far in robot control has assumed either kinematics or Jacobian matrix of the robot manipulator is known exactly [1], [2]. Unfortunately, due to the imprecision measurement of manipulator parameters and interactions between robot and different environments, it is consequently difficult to obtain the exact kinematic model. Therefore, robot kinematic uncertainty is a practical problem when the control objective is formulated directly in task space.

In [3], Arimoto described the importance of the problem with uncertain kinematic parameters and stated the research which targeted this problem was just in a beginning stage. In [4], Cheah *et al.* developed an approximate Jacobian feedback controller which exploited a static, best-guess estimate of the manipulator Jacobian to achieve control objectives. Some drawbacks of this controller, such as the requirement of task-space velocity of the robot end-effector, were resolved in [5]. In contrast to the use of a static estimated Jacobian, in [6], an adaptive controller was proposed to compensate the parametric uncertainty in the manipulator Jacobian, which eliminated the bounded mismatch assumption required in [4], [5]. However, above methods focus on the setpoint control of robot. As to tracking control problems, Cheah *et al.* suggested an adaptive Jacobian controller for trajectory tracking of non-redundant robot with uncertain kinematics and dynamics [7]. Extensions to the redundant robot and uncertain actuator parameters were done in [8]. In [9], the orientation tracking problem of non-redundant manipulators was solved well by employing the unit quaternion representation. It is noted that aforementioned adaptive controllers employ the standard adaptive control scheme to compensate the effects of gravity and other terms in the manipulator dynamics, which means that they will suffer from the "linearity-in-parameters" assumption and the tedious analysis of determining "regression matrix". In addition, the surface friction and external disturbance in robot dynamics have been neglected in the controller design.

Recently, neural networks have been successfully used for the nonlinear system identification and control due to their "universal-approximation" property [10]. Several neural-network-based adaptive controllers are also presented to eliminate the "linearity-in-parameters" assumption in standard adaptive control (see [11] for the general framework of these methods). The stability of this neural-network-based adaptive control system is guaranteed by the Lyapunov synthesis method, and synaptic weights of neural networks are tuned on-line without any off-line learning phases. In literature, some adaptive neural network controllers have been proposed for the tracking control of robot manipulators [12], [13]. However, these controllers are designed to move the robot along the desired joint angles, the manipulator kinematics is not taken into account.

This paper addresses the task-space tracking problem of robot manipulators with uncertain kinematics and dynamics. The unit quaternion is utilized to represent the orientation of manipulators, then the singularity problem occurred in the three-parameter representations (e.g. Euler angles, Rodrigues parameters) will be avoided. By employing adaptive Jacobian method, neural network approximation, and backstepping technique, an adaptive neural network controller is obtained. The adaptive updating laws for uncertain kinematic parameters and neural network synaptic weights are derived by the projection method. Stability of the proposed controller is guaranteed by the Lyapunov theory. And the tracking error can be reduced as small as desired. Compared with aforementioned controllers for uncertain kinematics, the proposed controller has several features: (1) the "linearity-in-parameters" assumption for the manipulator dynamics is no longer necessary; (2) external disturbances and surface frictions are taken into account; (3) the task-space velocity of robot end-effector is not required.

The remainder of this paper is organized as follows.

Long Cheng, Zeng-Guang Hou, Min Tan are with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, The Chinese Academic of Sciences, Beijing, China; Long Cheng is also with the Graduate School of Chinese Academy of Sciences; (emails: {long.cheng, zengguang.hou, min.tan}@ia.ac.cn).

This work was supported in part by the National Natural Science Foundation of China (Grants 60635010 and 60775043).

Section II introduces the problem formulation and some preliminary results. Section III discusses the controller design procedure. Section IV demonstrates the stability of proposed controller. Illustrative simulation is given in Section V. Section VI concludes this paper with final remarks.

II. Mathematical Preliminaries

A. Kinematics and Dynamics of Robot Manipulators

The dynamic model for a rigid n-link, serially connected robot manipulator can be expressed as [13]

$$M(\theta)\ddot{\theta} + V(\theta,\dot{\theta})\dot{\theta} + F(\dot{\theta}) + G(\theta) + \tau_{ed} = \tau, \qquad (1)$$

where $\theta, \dot{\theta}, \ddot{\theta} \in \mathbb{R}^n$ denote the joint position, velocity, and acceleration vectors, respectively; $M(\theta) \in \mathbb{R}^{n\times n}$ is the inertia matrix; $V(\theta,\dot{\theta}) \in \mathbb{R}^{n\times n}$ is the centripetal-Coriolis matrix; $F(\dot{\theta}) \in \mathbb{R}^n$ denotes the surface friction; $G(\theta) \in \mathbb{R}^n$ is the gravitational vector; $\tau_{ed} \in \mathbb{R}^n$ denotes the bounded external disturbance vector including unstructured model dynamics; $\tau \in \mathbb{R}^n$ represents the torque input vector.

Two important properties of the dynamics equation described by (1) are given as follows [6].

Property 1: The inertia matrix $M(\theta)$ is symmetric and positive definite, and satisfies the following inequalities:

$$m_1\|y\|_2^2 \le y^T M(\theta) y \le m_2 \|y\|_2^2, \quad \forall y \in \mathbb{R}^n,$$

where m_1 and m_2 are known positive constants, and $\|\cdot\|_2$ denotes the standard Euclidean norm.

Property 2: The time derivative of the inertia matrix and the centripetal-Coriolis matrix satisfy the skew symmetric relation; that is,

$$y^T\left(\dot{M}(\theta) - 2V(\theta,\dot{\theta})\right) y = 0, \quad \forall y \in \mathbb{R}^n.$$

Let Ψ_m and Ψ_b be orthogonal coordinate frames attached to the manipulator end-effector and fixed base, respectively. Let $p(t) \in \mathbb{R}^3$ represent the position of the original of Ψ_m relative to the origin of Ψ_b. Traditionally, the orientation of Ψ_m relative to Ψ_b can be described by a rotation matrix $R(t)$. However, this representation is clearly impractical because there are too many elements in the matrix, and not all of elements are independent. As an alternative, the three-parameter representation (e.g. Euler angles, Rodrigues parameters) is used widely to specify the orientation. Although it is the simplest representation method, the singularity problem is inevitable, which results in the degraded performance or unpredictable responses by the manipulator. To resolve this problem, an efficient way to specify the orientation is the quaternion description which is given by $q(t) \overset{\text{def}}{=} \left[q_o(t), q_v^T(t)\right]^T \in \mathbb{R}^4$. It is shown in [14] that the rotation matrix $R(t)$ can be calculated by

$$R(t) = (q_o^2 - q_v^T q_v)I_3 + 2q_v q_v^T + 2q_o A(q_v), \qquad (2)$$

where I_3 is the 3×3 identity matrix, and the notation $A(a)$, $\forall a = [a_1, a_2, a_3]^T$ denotes the following skew-symmetric matrix:

$$A(a) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Quaternions have many interesting properties, such as normality, uniqueness, and it is related to the end-effector angular velocity $\omega(t)$ via the following differential equation [14], [15]

$$\dot{q}(t) = \frac{1}{2}B(q)\omega, \qquad (3)$$

where

$$B(q) = \begin{bmatrix} -q_v^T \\ q_o I_3 - A(q_v) \end{bmatrix}.$$

It is well-known that the manipulator forward kinematics can be expressed by

$$\begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(\theta, \phi_J)\dot{\theta} \qquad (4)$$

where $\phi_J \in \mathbb{R}^r$ represents the kinematic parameters, such as link lengths and joint offsets; $J(\theta,\phi_J) \in \mathbb{R}^{6\times n}$ denotes the manipulator Jacobian matrix which has the following property.

Property 3: The product of the manipulator Jacobian matrix with the joint velocity vector can be linearly parameterized as

$$J(\theta,\phi_J)\dot{\theta} = Y_J(\theta,\dot{\theta})\phi_J, \qquad (5)$$

where $Y_J(\theta,\dot{\theta}) \in \mathbb{R}^{6\times p}$ can be computed directly by the measurable joint position θ and velocity vectors $\dot{\theta}$.

B. Radial Basis Function Neural Networks

In control engineering, neural networks are usually employed as the function approximator to emulate the unknown ideal control signal. Due to the "linear-in-the-weight" property, the radial basis function neural network (RBFNN) is a good candidate for this purpose. In this paper, the following RBFNN [13] is used to approximate the continuous function $h(Z): \mathbb{R}^m \to \mathbb{R}^n$,

$$h_{nn}(Z) = W^T S(Z), \qquad (6)$$

where the input vector $Z \in \Omega \subset \mathbb{R}^m$, weight matrix $W \in \mathbb{R}^{l\times n}$, l denotes the number of neural network node, and $S(Z) = [s_1(Z), \cdots, s_l(Z)]^T$ with

$$s_i(Z) = \exp\left[\frac{-(Z-\mu_i)^T(Z-\mu_i)}{\sigma_i^2}\right], \quad i = 1, 2, \cdots, l$$

where $\mu_i = [\mu_{i1}, \mu_{i2}, \cdots, \mu_{im}]^T$ is the center of the receptive field and σ_i is the width of the Gaussian function.

It has been proven that above RBFNN can approximate any smooth function over a compact set $\Omega_Z \subset \mathbb{R}^m$ to arbitrarily accuracy. That is, for any given positive constant ε_N, there exist the ideal weight matrix W^* and the number of neural network node l such that

$$h(Z) = W^{*T} S(Z) + \varepsilon, \qquad (7)$$

where ε is the bounded function approximation error satisfying $|\varepsilon| < \varepsilon_N$ in Ω_Z.

It is noted that the ideal matrix W^* is only quantity required for analytical purpose. For real applications, its estimation $\hat{W}$ is used for the practical function approximation. The estimation of $h(z)$ can be given by

$$\hat{h}(Z) = \hat{W}^T S(Z). \qquad (8)$$

C. Stability of Systems

Definition 1 ([13]): Given a nonlinear dynamical system

$$\dot{x}(t) = f(x,t), \quad x(t) \in \mathbb{R}^n, \quad t \geq t_0.$$

If there exists a compact set $U_x \subset \mathbb{R}^n$ such that for all $x(t_0) = x_0 \in U_x$, there exist a $\delta > 0$ and a number $T(\delta, x_0)$ such that $\|x(t)\| \leq \delta$ for all $t \geq t_0 + T$, then the solution of the nonlinear dynamical system is called *uniformly ultimately bounded (UUB)*.

III. Adaptive Neural Network Controller

The control objective is to develop a task-space tracking controller for the robot manipulator with uncertainties and external disturbances. Here, the backstepping approach is employed to achieve this control goal. The backstepping method designs partial Lyapunov functions and auxiliary controllers for each subsystem of the whole nonlinear system, and integrates these individual controllers into the actual controller by "back stepping" through the system and reassembling it from its component subsystems [16].

First, the desired position and orientation of the robot end-effector is defined by a desired orthogonal coordinate frame Ψ_d. Let $p_d(t) \in \mathbb{R}^3$ denote the desired position of the origin of Ψ_d relative to the origin of Ψ_b. It is commonly assumed that $p_d(t)$ and its derivatives up to the second order are bounded. Define the end-effector position tracking error $e_p(t)$ as

$$e_p = p - p_d. \tag{9}$$

The orientation of Ψ_d relative to Ψ_b is specified by a desired unit quaternion $q_d(t) = [q_{od}(t), q_{ov}^T(t)]^T \in \mathbb{R}^4$. Then by (2), the rotation matrix $R_d(t)$ of Ψ_d relative to Ψ_b can be obtained. According to (3), the time derivative of $q_d(t)$ is related to the desired angular velocity of the end-effector $\omega_d(t) \in \mathbb{R}^3$ as follows

$$\dot{q}_d = B(q_d)\omega_d. \tag{10}$$

According to the analysis of [15], the rotation matrix $\tilde{R}(t) = R_d^T R$ from Ψ_m to Ψ_d is defined to quantify the difference between the actual and desired end-effector orientations. The quaternion representation of $\tilde{R}$ is given by $e_q(t) = [e_o(t), e_v^T(t)]^T$ whose derivation has the following form

$$\dot{e}_o = -\frac{1}{2} e_v^T R_d^T (\omega - \omega_d), \tag{11a}$$

$$\dot{e}_v = \frac{1}{2}\left(e_o I_3 - S(e_v)\right) R_d^T (\omega - \omega_d). \tag{11b}$$

It can be seen that, if $\lim_{t\to\infty} e_v(t) = 0$, then $\lim_{t\to\infty} e_o(t) = 1$ and $\lim_{t\to\infty} \tilde{R}(t) = I_3$, which means that the orientation tracking error is zero [15]. Therefore, the tracking control objective can be stated as

$$\lim_{t\to\infty} e_p(t) = 0, \quad \lim_{t\to\infty} e_v(t) = 0. \tag{12}$$

To achieve this control goal, by the methodology of backstepping approach, $\dot{\theta}$ is first designed as an auxiliary controller which makes $e_p(t)$ and $e_v(t)$ approach zero. Construct the following Lyapunov function

$$L_1 = \frac{1}{2} e_p^T e_p + (1 - e_o)^2 + e_v^T e_v. \tag{13}$$

By (4) and (11), the derivative of L_1 can be obtained that

$$\begin{aligned}\dot{L}_1 =& - e_p^T K_p e_p - e_v^T K_v e_v + e_1^T \begin{bmatrix} \dot{p}_d + K_p e_p \\ -R_d^T \omega_d + K_v e_v \end{bmatrix} \\ &+ e_1^T \Lambda J(\theta, \phi_J)\dot{\theta},\end{aligned} \tag{14}$$

where $e_1 = \left[e_p^T, e_v^T\right]^T$, and K_p, $K_v \in \mathbb{R}^{3\times 3}$ are diagonal positive definite matrices, respectively; and

$\Lambda = \begin{bmatrix} -I_{3\times 3} & \Theta_{3\times 3} \\ \Theta_{3\times 3} & R_d^T \end{bmatrix}$, $\Theta_{3\times 3}$ denotes the zero matrix.

In the presence of kinematic uncertainty, the parameter ϕ_J in the Jacobian matrix $J(\theta, \phi_J)$ is not known exactly. By replacing the unknown parameter ϕ_J with its estimation $\hat{\phi}_J$, an approximate Jacobian $\hat{J}\left(\theta, \hat{\phi}_J\right)$ can be obtained. Then the auxiliary controller for $\dot{\theta}$, which is called by $\dot{\theta}_d$, is chosen as

$$\begin{aligned}\dot{\theta}_d =& -\left(\Lambda \hat{J}(\theta, \hat{\phi}_J)\right)^+ \begin{bmatrix} \dot{p}_d + K_p e_p \\ -R_d^T \omega_d + K_v e_v \end{bmatrix} + (I_{n\times n} - \\ &\left(\Lambda \hat{J}(\theta, \hat{\phi}_J)\right)^+ \left(\Lambda \hat{J}(\theta, \hat{\phi}_J)\right))\lambda,\end{aligned} \tag{15}$$

where $(\cdot)^+$ denotes the pseudoinverse of given matrix, i.e. $(G)^+ = G^T \left(GG^T\right)^{-1}$; $\lambda \in \mathbb{R}^n$ is an auxiliary term which can be used for optimization purposes. It is assumed that the manipulator is operating in a finite task space such that the approximate Jacobian matrix $\hat{J}\left(\theta, \hat{\phi}_J\right)$ is of full rank. This assumption is commonly adopted to deal with manipulator kinematic uncertainty in the existing literature [6]–[9]. Then (14) can be rewritten as

$$\begin{aligned}\dot{L}_1 =& -e_p^T K_p e_p - e_v^T K_v e_v + e_1^T \Lambda J(\theta, \phi_J) e_2 \\ &+ e_1^T \Lambda \left(J(\theta, \phi_J) - \hat{J}(\theta, \hat{\phi}_J)\right)\dot{\theta}_d \\ =& -e_p^T K_p e_p - e_v^T K_v e_v + e_1^T \Lambda Y_J(\theta, \dot{\theta}_d)\left(\phi_J - \hat{\phi}_J\right) \\ &+ e_1^T \Lambda J(\theta, \phi_J) e_2 \\ =& e_1^T \Lambda \left(J(\theta, \phi_J) e_2 - Y_J(\theta, \dot{\theta}_d)\tilde{\phi}_J\right) - e_p^T K_p e_p \\ &- e_v^T K_v e_v,\end{aligned} \tag{16}$$

where $\tilde{\phi}_J = \hat{\phi}_J - \phi_J$ and $e_2 = \dot{\theta} - \dot{\theta}_d$. It is assumed that the uncertain parameter ϕ_J in manipulator kinematics is bounded by its upper limit ϕ_J^+ and lower limit ϕ_J^-, i.e. $(\phi_J^-)_i \leq (\phi_J)_i \leq (\phi_J^+)_i$, $i = 1, 2, \cdots, p$, where $(\cdot)_i$ denotes the ith element of given vector.

The second step is try to design real torque controller τ which makes e_2 as small as desired. To achieve this, the error dynamics for e_2 is derived by (1) that

$$\begin{aligned}M(\theta)\dot{e}_2 + V(\theta, \dot{\theta})e_2 =& \tau - M(\theta)\ddot{\theta}_d - V(\theta, \dot{\theta})\dot{\theta}_d - F(\dot{\theta}) \\ &- G(\theta) - \tau_{ed} \\ =& \tau - f_1 - \tau_{ed}.\end{aligned} \tag{17}$$

The torque controller τ is chosen as

$$\tau = \hat{F}_1 - K_1 e_2 - \gamma_1, \tag{18}$$

where K_1 is a diagonal positive definite gain matrix; γ_1 is a robustness signal which counteracts the approximation error and external disturbances in the second step. $\hat{F}_1$ is the estimation of F_1 which is defined by

$$F_1 = f_1 - J^T(\theta, \phi_J)\Lambda^T e_1. \tag{19}$$

It is emphasized that the term $-J^T(\theta, \phi_J)\Lambda^T e_1$ in F_1 is used to compensate the coupling term $e_1^T \Lambda J(\theta, \phi_J) e_2$ in (16). In the standard adaptive scheme, it has to assume that the uncertain term F_1 has the "linearity-in-parameters" property in order to obtain the adaptive parameters updating law. However, this assumption does not hold if the friction $F(\dot{\theta})$ has the particular nonlinear form (see examples in [17]). Motivated by the universal approximation ability of neural networks, the RBFNN is employed to learn the unknown function F_1. By the previous introduction for RBFNN, it can be obtained that, over a compact set,

$$F_1 = {W^*}^T S(Z_1) + \varepsilon_1, \tag{20}$$

with the approximation error ε_1 and neural network input $Z_1 = [e_1^T, e_2^T, p_d^T, \dot{p}_d^T, \ddot{p}_d^T, q^T, \omega_d^T, \dot{\omega}_d^T]^T$. The estimation of F_1 is given by

$$\hat{F}_1 = \hat{W}^T S(Z_1). \tag{21}$$

Substituting (18), (19), (20) and (21) into (17) obtains that

$$\begin{aligned} M(\theta)\dot{e}_2 + V(\theta, \dot{\theta})e_2 &= \hat{F}_1 - K_2 e_2 - \gamma_1 - f_1 - \tau_{ed} \\ &= -K_1 e_2 - J^T(\theta, \phi_J)\Lambda^T e_1 + \delta_1 - \gamma_1 + \tilde{W}^T S(Z_1), \end{aligned} \tag{22}$$

where $\tilde{W} = \hat{W} - W^*$; $\delta_1 = -\tau_{ed} + \varepsilon_1$.
The robustness signal γ_1 takes the following hyperbolic tangent form

$$\gamma_1 = \delta_{M1} \tanh\left(\frac{2k_u \delta_{M1} e_2}{\epsilon_1}\right), \tag{23}$$

where $k_u = 0.2785$, ϵ_1 is a positive design scalar, δ_{M1} is the upper bound of δ_1. It is easy to check that γ_1 has the following properties

$$e_2^T \gamma_1 \geq 0, \tag{24a}$$

$$\delta_{M1}\|e_2\| - e_2^T \gamma_1 \leq \epsilon_1. \tag{24b}$$

By the projection algorithm [6], the updating laws for estimated kinematic parameters $\hat{\phi}_J$ and neural network weight matrix $\hat{W}$ are derived as follows

$$\left(\dot{\hat{\phi}}_J\right)_j = \begin{cases} \beta_j \left(Y_J^T(q, \dot{q}_d)\Lambda^T e_1\right)_j, \text{ if } (\phi_J^-)_j < (\hat{\phi}_J)_j < (\phi_J^+)_j \\ \text{or if } (\hat{\phi}_J)_j = (\phi_J^-)_j \text{ and } \left(Y_J^T(q, \dot{q}_d)\Lambda^T e_1\right)_j > 0, \\ \text{or if } (\hat{\phi}_J)_j = (\phi_J^+)_j \text{ and } \left(Y_J^T(q, \dot{q}_d)\Lambda^T e_1\right)_j \leq 0; \\ 0, \text{ if } (\hat{\phi}_J)_j = (\phi_J^-)_j \text{ and } \left(Y_J^T(q, \dot{q}_d)\Lambda^T e_1\right)_j \leq 0, \\ \text{or if } (\hat{\phi}_J)_j = (\phi_J^+)_j \text{ and } \left(Y_J^T(q, \dot{q}_d)\Lambda^T e_1\right)_j > 0; \end{cases}$$

$$\text{for} \quad j = 1, 2, \cdots, p. \tag{25}$$

$$\dot{\hat{W}} = \begin{cases} -\Gamma_w S(Z_1)e_2^T, \text{ if } \operatorname{Tr}\left(\hat{W}^T \Gamma_w^{-1} \hat{W}\right) < W_m, \text{ or if} \\ \operatorname{Tr}\left(\hat{W}^T \Gamma_w^{-1} \hat{W}\right) = W_m \text{ and } e_1^T \hat{W}^T S(Z_1) > 0; \\ \Gamma_w \dfrac{e_2^T \hat{W}^T S(Z_1)}{\operatorname{Tr}\left(\hat{W}^T \Gamma_w^{-1} \hat{W}\right)} W - \Gamma_w S(Z_1)e_2^T, \text{if} \\ \operatorname{Tr}\left(\hat{W}^T \Gamma_w^{-1} \hat{W}\right) = W_m \text{ and } e_1^T \hat{W}^T S(Z_1) \leq 0, \end{cases} \tag{26}$$

where W_m is the positive constant for limiting the estimated neural network weight matrix, which satisfies the condition $\operatorname{Tr}({W^*}^T \Gamma_w^{-1} W^*) \leq W_m$; Γ_w is a diagonal matrix with positive diagonal elements which controls the parameter adaption rate.

It is emphasized that the initial kinematics parameter $\hat{\phi}_J(0)$ should be selected as

$$(\phi_J^-)_j \leq \left(\hat{\phi}_J(0)\right)_j \leq (\phi_J^+)_j, \tag{27}$$

and the initial neural network weight matrix $\hat{W}(0)$ satisfies that

$$\operatorname{Tr}\left(\hat{W}(0)\Gamma_{w1}^{-1}\hat{W}^T(0)\right) \leq W_m. \tag{28}$$

IV. Stability Analysis

Theorem 1: Given the robot manipulator defined by (1) and (4), if the controller is constructed as (18), the parameters updating laws are provided by (25) and (26), and the initial values of estimated parameters satisfy the conditions (27) and (28), then e_p, e_v, e_2, $\hat{\phi}_J$ and $\hat{W}$ are all uniformly ultimately bounded signals. Moreover, e_p, e_v and e_2 can be reduced as small as desired by choosing appropriate controller parameters.

Proof: According to the principle of projection algorithm, it is easy to check that $\hat{\phi}_J$ is bounded by its upper and lower limitations.
To prove the boundness of $\hat{W}$, let $L_w = \operatorname{Tr}\left(\hat{W}^T \Gamma_w^{-1} \hat{W}\right)$. By (26), it follows that
(1). When $L_w = W_m$ and $e_2^T \hat{W}^T S(Z_1) > 0$,

$$\begin{aligned} \frac{dL_w}{dt} &= 2\operatorname{Tr}\left(\hat{W}^T \Gamma_w^{-1} \dot{\hat{W}}\right) = -2\operatorname{Tr}\left(\hat{W}^T S(Z_1)e_2^T\right) \\ &= -2e_2^T \hat{W}^T S(Z_1) < 0; \end{aligned}$$

(2). When $L_w = W_m$ and $e_2^T \hat{W}^T S(Z_1) \leq 0$,

$$\begin{aligned} \frac{dL_w}{dt} &= 2\operatorname{Tr}\left(\hat{W}^T \Gamma_w^{-1} \dot{\hat{W}}\right) = -2\operatorname{Tr}(\hat{W}^T S(Z_1)e_2^T) \\ &\quad + 2\operatorname{Tr}\left(\hat{W}^T \Gamma_w^{-1} \frac{e_2^T \hat{W}^T S(Z_1)}{\operatorname{Tr}\left(\hat{W}^T \Gamma_w^{-1} \hat{W}\right)} \hat{W}\right) \\ &= -2e_2^T \hat{W}^T S(Z_1) + 2e_2^T \hat{W}^T S(Z_1) = 0. \end{aligned}$$

Hence, if the initial neural network weight matrix $\hat{W}(0)$ satisfies (28), then $\operatorname{Tr}\left(\hat{W}^T \Gamma_w^{-1} \hat{W}\right) \leq W_m$ always holds, which means that $\hat{W}$ is bounded.
To prove the uniform ultimate boundedness of error signals e_1 and e_2, the following Lyapunov function is considered,

$$V = V_1 + V_2, \tag{29}$$

where

$$V_1 = L_1 + \frac{1}{2}\tilde{\phi}_J^T \Gamma_\beta^{-1} \tilde{\phi}_J, \tag{30a}$$

$$V_2 = \frac{1}{2}\left(e_2^T M(\theta) e_2 + \mathrm{Tr}\left(\tilde{W}^T \Gamma_w^{-1} \tilde{W}\right)\right). \tag{30b}$$

with $\Gamma_\beta = \mathrm{diag}\,(\beta_1, \beta_2, \cdots, \beta_p) \in \mathbb{R}^{p\times p}$.

By (25) and (16), the time derivative of V_1 can be obtained that

$$\begin{aligned}\dot{V}_1 &= \dot{L}_1 + \tilde{\phi}_J^T \Gamma_\beta^{-1} \dot{\tilde{\phi}}_J \\ &= e_1^T \Lambda J(\theta, \phi_J) e_2 - e_p^T K_p e_p - e_v^T K_v e_v \\ &\quad - \tilde{\phi}_J^T \left(Y_J^T(q, \dot{q}_d)\Lambda^T e_1 - \Gamma_\beta^{-1}\dot{\hat{\phi}}_J\right) \\ &\leq -e_p^T K_p e_p - e_v^T K_v e_v + e_1^T \Lambda J(\theta, \phi_J) e_2. \end{aligned} \tag{31}$$

By (1), (22), (24) and (26), the time derivative of V_2 is

$$\begin{aligned}\dot{V}_2 &= e_2^T M(\theta)\dot{e}_2 + \frac{1}{2} e_2^T \dot{M}(\theta) e_2 + \mathrm{Tr}\left(\tilde{W}^T \Gamma_w^{-1} \dot{\hat{W}}\right) \\ &\leq -e_2^T K_1 e_2 - e_2^T J^T(\theta, \phi_J)\Lambda^T e_1 + \|e_2^T\|\|\delta_1\| - e_2^T \gamma_1 \\ &\quad + e_2^T \tilde{W}^T S(Z_1) + \mathrm{Tr}\left(\tilde{W}^T \Gamma_w^{-1} \dot{\hat{W}}\right) \\ &\leq -e_2^T K_1 e_2 - e_2^T J^T(\theta, \phi_J)\Lambda^T e_1 + \epsilon_1 \\ &\quad + \mathrm{Tr}\left(\tilde{W}^T \left(S(Z_1) e_2^T + \Gamma_w^{-1}\dot{\hat{W}}\right)\right). \end{aligned} \tag{32}$$

By (26), it follows that

(1). If $\dot{\hat{W}} = -\Gamma_w S(Z_1) e_2^T$, then

$$\mathrm{Tr}\left(\tilde{W}^T \left(S(Z_1) e_2^T + \Gamma_w^{-1}\dot{\hat{W}}\right)\right) = 0.$$

(2). If $\dot{\hat{W}} = \Gamma_w \dfrac{e_2^T \hat{W}^T S(Z_1)}{\mathrm{Tr}\left(\hat{W}^T \Gamma_w^{-1}\hat{W}\right)} W - \Gamma_w S(Z_1) e_2^T$, then $\mathrm{Tr}\left(\hat{W}^T \Gamma_w^{-1}\hat{W}\right) = W_m,\ e_2^T \hat{W}^T S(Z_1) \leq 0$, and $\mathrm{Tr}\left(\tilde{W}^T (S(Z_1) e_2^T + \Gamma_w^{-1}\dot{\hat{W}})\right) = \dfrac{e_2^T \hat{W}^T S(Z_1)}{\mathrm{Tr}\left(\hat{W}^T \Gamma_w^{-1}\hat{W}\right)}\mathrm{Tr}\left(\tilde{W}^T \Gamma_w^{-1}\hat{W}\right)$.

It is noted that

$$\begin{aligned} &2\,\mathrm{Tr}\left(\tilde{W}^T \Gamma_w^{-1}\hat{W}\right) = 2\,\mathrm{Tr}\left(\tilde{W}^T \Gamma_w^{-1}\tilde{W}\right) + 2\,\mathrm{Tr}\left(\tilde{W}^T \Gamma_w^{-1} W^*\right) \\ &= \mathrm{Tr}\left(\tilde{W}^T \Gamma_w^{-1}\tilde{W}\right) + \mathrm{Tr}\left(\hat{W}^T \Gamma_w^{-1}\hat{W}\right) - \mathrm{Tr}\left(W^{*T} \Gamma_w^{-1} W^*\right) \\ &\geq 0, \end{aligned}$$

where the facts that $\mathrm{Tr}\left(W^{*T} \Gamma_w^{-1} W^*\right) \leq W_m$ and $\mathrm{Tr}\left(\tilde{W}^T \Gamma_w^{-1}\tilde{W}\right) \geq 0$ have been used.

Therefore, it can be seen that in both cases the following fact holds

$$\mathrm{Tr}\left(\tilde{W}^T \left(S(Z_1) e_2^T + \Gamma_w^{-1}\dot{\hat{W}}\right)\right) \leq 0.$$

Hence, according to (32), it can be obtained that

$$\dot{V}_2 \leq -e_2^T K_1 e_2 - e_2^T J^T(\theta, \phi_J)\Lambda^T e_1 + \epsilon_1. \tag{33}$$

Thus, by (31) and (33), the time derivative of Lyapunov function V is

$$\begin{aligned}\dot{V} &= \dot{V}_1 + \dot{V}_2 \leq -e_2^T K_1 e_2 - e_p^T K_p e_p - e_v^T K_v e_v + \epsilon_1 \\ &= -e^T K e + \epsilon_1 \leq -\lambda_{\min}(K)\|e\|^2 + \epsilon_1, \end{aligned} \tag{34}$$

where $K = \mathrm{diag}\,(K_1, K_p, K_v)$, $e = \left(e_2^T, e_p^T, e_v^T\right)^T$, and $\lambda_{\min}(K)$ is the minimum eigenvalue of matrix K.

Therefore, $\dot{V}$ is strictly negative outside the following compact set $\sum_e$

$$\sum\nolimits_e = \left\{ e(t) \,\middle|\, 0 \leq \|e(t)\| \leq \sqrt{\frac{\epsilon_1}{\lambda_{\min}(K)}} \right\}. \tag{35}$$

According to the Lyapunov theory extension [13], this demonstrates that e_2, e_p and e_v are all uniformly ultimately bounded signals. And $\exists T_o$, for $\forall t > T_o$, $e(t) \in \left\{\|e\| \,\middle|\, 0 \leq \|e\| \leq \sqrt{\frac{\epsilon}{k}} + \varepsilon_s\right\}$, where $\varepsilon_s > 0$ is an arbitrarily small positive constant. Therefore, by choosing K sufficiently large and ϵ_1 sufficiently small, e_2, e_p and e_v can be reduced as small as desired. ■

V. Simulation Example

Computer simulations based on the Unimation PUMA 560 robot arm is conducted to demonstrate the effectiveness of the proposed controller. The mechanical configuration and coordinate system are given in [18]. The initial joint configuration of PUMA 560 is $\theta(0) = [\pi/4, \pi/4, \pi/4, \pi/4, \pi/4, \pi/4]^T$rad and $\dot{\theta}(0) = [0, 0, 0, 0, 0, 0]^T$rad/s. Table I gives the Denavit and Hartenberg parameters of the PUMA 560 manipulator, where link length $\alpha_2 = 0.4318$m, $\alpha_3 = 0.0203$m and joint offset $d_3 = 0.15005$m, $d_4 = 0.4318$m. α_2, α_3, d_3 and d_4 are the uncertain kinematics parameters. In the proposed controller, they are estimated as $\hat{a}_2(0) = \hat{a}_3(0) = \hat{d}_3(0) = \hat{d}_4(0) = 0.1$m. The upper and lower limits of the estimated parameters are $(0.5, 0.5, 0.5, 0.5)^T$ and $(0, 0, 0, 0)^T$. The PUMA 560 end-effector is required to follow a given trajectory [15]

$$p_d(t) = \begin{bmatrix} 0.1\sin(0.1t)\left(1 - \exp(-0.01t^3)\right) + 0.02 \\ 0.1\sin(0.1t)\left(1 - \exp(-0.01t^3)\right) - 0.25 \\ 0.1\sin(0.1t)\left(1 - \exp(-0.01t^3)\right) + 0.40 \end{bmatrix} \text{m}.$$

And the end-effector orientation is commanded to be stable at $q_d(t) = [-0.5, 0.5, 0.5, 0.5]^T$. In this simulation, parameters of controller are set that $K_p = 2I_{3\times3}$; $K_v = 5I_{3\times3}$; $K_1 = 20I_{6\times6}$; $\lambda = 0$; $\tau_{ed} = (5\cos(\pi t/2), 4\sin(\pi t/2) + e^{-t}, 2\cos(t) + 3\sin(\pi t/3))^T$; $\Gamma_\beta = 2I_{4\times4}$; $\hat{W}(0)$ is set to be zero matrices; $W_m = 500$; $\delta_{M1} = 20$, $\epsilon_1 = 0.5$. The simulation results are shown in Figs. 1–4, which verifies the good tracking performance of the proposed controller.

VI. Conclusion

A neural-network-based adaptive controller is proposed to deal with the manipulator task-space tracking problem. The proposed controller eliminates the "linearity-in-parameters" assumption for the uncertain terms in manipulator dynamics, avoids the tedious computation of regression matrix, and considers the external disturbance. The good control performance can be demonstrated by the Lyapunov approach and illustrated by the simulation example. Finally, by the cascade backstepping design procedure, the proposed controller can also be extended to the cases where the uncertain actuator model or the flexible joint manipulator are considered.

TABLE I

THE DENAVIT AND HARTENBERG PARAMETERS OF PUMA 560.

Link i	θ_i (rad)	a_i (rad)	α_i (m)	d_i (m)
1	θ_1	$\pi/2$	0	0
2	θ_2	0	α_2	0
3	θ_3	$-\pi/2$	α_3	d_3
4	θ_4	$\pi/2$	0	d_4
5	θ_5	$-\pi/2$	0	0
6	θ_6	0	0	0

REFERENCES

[1] C. Abdallah, D. Dawson, P. Dorato and M. Jamshidi, "Survey of robust control for rigid robots," *IEEE Control Systems Magnize*, vol. 11, no. 2, pp. 24–30, 1991.

[2] H. Berghuis, R. Ortega and H. Nijmeijer, "A robust adaptive robot controller," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 6, pp. 825–830, 1993.

[3] S. Arimoto, "Robotics research toward explication of everyday physics," *International Journal of Robotics Research*, vol. 18, no. 11, pp. 1056–1063, 1999.

[4] C.C. Cheah, S. Kawamura and S. Arimoto, "Feedback control for robotic manipulators with an uncertain Jacobian matrix," *Journal of Robotic Systems*, vol. 12, no. 2, pp. 119–134, 1999.

[5] C.C. Cheah, S. Kawamurab and S. Arimoto, "Approximate Jacobian control for robots with uncertain kinematics and dyanmics", *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 692–702, 2003.

[6] W.E. Dixon, "Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics," *IEEE Transactions on Automatic Control*, vol. 52, no. 3, pp. 488–493, 2007.

[7] C.C. Cheah, C. Liu and J.J.E. Slotine, "Approximate Jacobian adaptive control for robot manipulators," *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, LA, pp. 3075–3080, 2004.

[8] C.C. Cheah, C. Liu and J.J.E. Slotine, "Adaptive Jacobian tracking control of robots with uncertainties in kinematic, dynamic and actuator models," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, pp. 1024–1029, 2006.

[9] D. Braganza, W.E. Dixon, D.M. Dawson and B. Xian, "Tracking Control for Robot Manipulators with Kinematic and Dynamic Uncertainty," *IEEE Conference on Decision and Control*, Seville, Spain, pp. 5293–5297, 2005.

[10] K.S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 252–262, 1991.

[11] J.A. Farrell and M.M. Polycarpou, *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches*, Hoboken, NJ: Wiley-Interscience, 2006.

[12] F.L. Lewis, A. Yesildirek and K. Liu, "Multilayer nerual-net robot controller with guaranteed tracking performance," *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 388–398, 1996.

[13] F.L. Lewis, S. Jagannathan and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. New York: Taylor & Francis, 1998.

[14] J.S. Yuan, "Closed-loop manipulator control using quaternion feedback," *IEEE Transactions on Robotics and Automation*, vol. 4, no. 4, pp. 434–440, 1988.

[15] B. Xian, M.S. de Queiroz, D. Dawson and I. Walker, "Task-space tracking control of robot manipulators via quaternion feedback," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 160–167, 2004.

[16] I. Kanellakopoulos, P.V. Kokotovic and A.S. Morse, "Systematic design of adaptive controllers for feedback linearizable systems," *IEEE Transactions on Automatic Control*, vol. 36, no. 11, pp. 1241–1253, 1991.

[17] S.S. Ge, T.H. Lee and S.X. Ren, "Adaptive friction compensation of servo mechanisms," *International Journal of Systems Science*, vol. 32, no. 4, pp. 523–532, 2001.

[18] P.I. Corke and B. Armstrong-Helouvry, "A search for consensus among model parameters reported for the PUMA 560 robot," *Proceedings of IEEE International Conference on Robotics and Automation*, San Diego, CA, pp. 1608–1613, 1994.

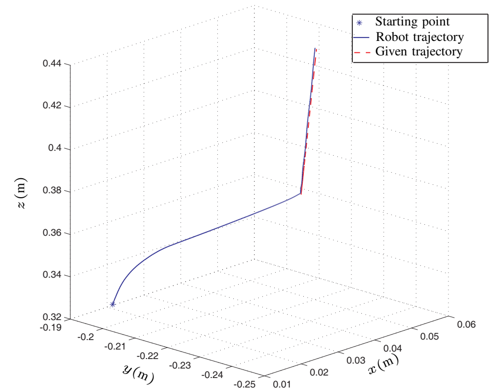

Fig. 1. The tracking performance of PUMA 560 manipulator.

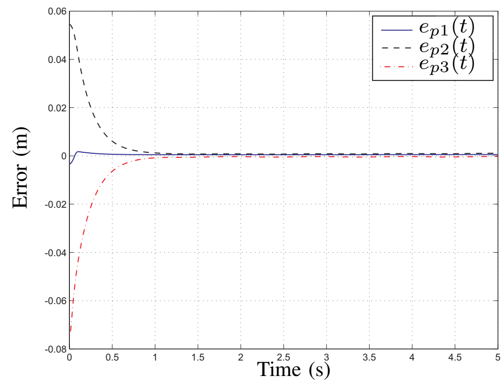

Fig. 2. The position tracking errors.

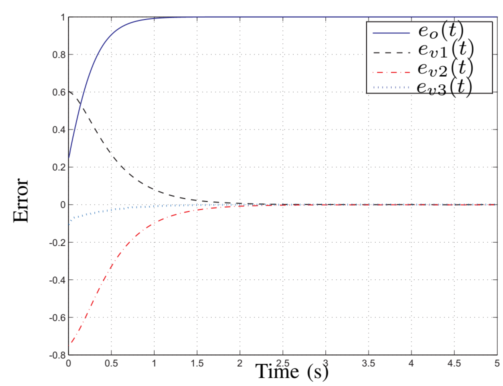

Fig. 3. The orientation tracking errors.

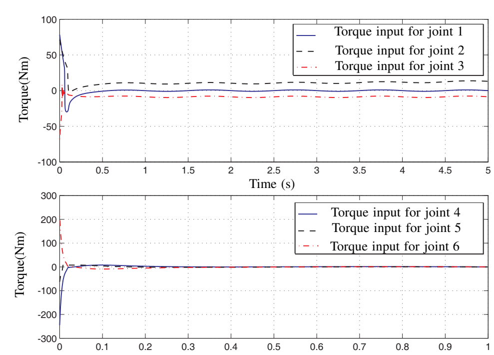

Fig. 4. The profile of control torque input.