Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Adaptive Neural Network Tracking Control for Manipulators with Uncertainties ⋆

Long Cheng[†,‡], Zeng-Guang Hou[†], Min Tan[†], and Hong-Ming Wang[†]

[†] *Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China*
[‡] *Graduate University of Chinese Academy of Sciences, Beijing 100039, China*

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

1. INTRODUCTION

In the past decades, robust adaptive motion control has been thoroughly studied to counteract uncertainties in the robot dynamics (see surveys Ortega and Spong [1989], Abdallah et al. [1991]). However, in most applications, the manipulator end-effector is required to track a given trajectory in the task space. The kinematics of robot manipulators is commonly assumed known exactly to achieve this control objective. Unfortunately, due to the imprecise measurement of physical parameters and the interaction between robot and different environments, the kinematic parameters cannot be known a prior. Therefore, uncertainty in the robot kinematic model is a practical problem.

There are relatively few articles to address this topic. As reported by Arimoto [1999], the research on the control problem with uncertain kinematics is just a beginning. Cheah et al. [1999] first proposed an approximation Jacobian method to overcome kinematic uncertainties. Based on this method, Cheah et al. [2003] also suggested a hybrid position and force controller when the interaction between the manipulator end-effector and environments was considered. Dixon [2007] presented an amplitude limited controller and eliminated the bounded mismatch assumption for estimated Jacobian. However, above results are focusing on the regulation control of robot. As to tracking control problems, Cheah et al. [2004] studied an adaptive Jacobian approach for trajectory tracking of non-redundant robot with uncertain kinematics and dynamics. Extensions to the redundant robot and uncertain actuator parameters were done by Cheah et al. [2006]. It is noted that aforementioned controllers employ the standard adaptive control scheme to compensate 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 matrices". 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 the "universal approximation" property [Liu, 2001]. Several neural-network-based adaptive controllers are also presented to eliminate the "linearity-in-parameters" assumption in standard adaptive control (see Farrell and Polycarpou [2006] for the general framework of these methods). An attractive feature of these methods is that 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 by Lewis et al. [1998], Ge et al. [1998]. However, these controllers are designed to move robots along the desired joint angles, the manipulator kinematics is not taken into account.

This paper addresses the end-effector tracking problem of robot manipulators with uncertain kinematics and dynamics. By using adaptive Jacobian method, neural network approximation, and backstepping technique, an adaptive neural network controller is obtained. The adaptive updating laws for controller parameters 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 possible. Compared with controller used in Cheah et al. [2004], the "linearity-in-parameters" assumption for the manipulator dynamics is no longer necessary, external disturbances and surface frictions are taken into account, and the velocity of robot end-effector is not required.

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

2. MATHEMATICAL PRELIMINARIES

Let $\|\cdot\|$ denote any suitable vector norm. When it is required to be specific, $\|\cdot\|_p$ denotes the p-norm of given vector. The Frobenius norm of matrix $A = [a_{ij}] \in \mathbb{R}^{n\times m}$ is defined by

$$\|A\|_F^2 = \mathrm{Tr}\left(A^T A\right) = \mathrm{Tr}\left(AA^T\right) = \sum_{i,j} a_{ij}^2,$$

where $\mathrm{Tr}(\cdot)$ represents the *trace* operator. The Frobenius norm is compatible with the 2-norm such that $\|Ax\|_2 \le \|A\|_F\|x\|_2$ with $A \in \mathbb{R}^{n\times m}$ and $x \in \mathbb{R}^m$. The trace operator has the following useful property, that is

$$a^T b = \mathrm{Tr}\left(ab^T\right), \tag{1}$$

where $\forall a,\ b \in \mathbb{R}^n$.

2.1 Kinematics and Dynamics of Robot Manipulators

Consider a rigid n-link, serially connected robot manipulator. Let $x \in \mathbb{R}^m$ $(m \le n)$ represent a task-space vector which is related to the robot joint vector $q \in \mathbb{R}^n$ as

$$x = h(q),$$

where $h(q) \in \mathbb{R}^m$ is the differentiable forward kinematics of the manipulator. The task-space velocity $\dot{x}$ is related to the joint-space velocity $\dot{q}$ as

$$\dot{x} = J(q, \phi_J)\dot{q}, \tag{2}$$

where $\phi_J \in \mathbb{R}^p$ represents the kinematic parameters of robot manipulators, such as link lengths and joint offsets; $J(q,\phi_J) \overset{\mathrm{def}}{=} (\partial h/\partial q) \in \mathbb{R}^{m\times n}$ denotes the manipulator's *Jacobian* matrix which has the following property.

Property 1. The product of the manipulator Jacobian matrix with the joint velocity vector can be linearly parameterized as

$$J(q,\phi_J)\dot{q} = Y_J(q,\dot{q})\phi_J, \tag{3}$$

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

The dynamic model of robot manipulators can be expressed as [Lewis et al., 1998]

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_{ed} = \tau, \tag{4}$$

where $\ddot{q} \in \mathbb{R}^n$ denotes the joint acceleration vector; $M(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix; $V(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is the centripetal-Coriolis matrix; $F(\dot{q}) \in \mathbb{R}^n$ denotes the surface friction; $G(q) \in \mathbb{R}^n$ 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 this dynamics formulation are given as follows [Lewis et al., 1998].

Property 2. The inertia matrix $M(q)$ is symmetric and positive definite, and satisfies the following inequalities:

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

where m_1 and m_2 are known positive constants.

Property 3. The time derivative of the inertia matrix and the centripetal-Coriolis matrix satisfy the skew symmetric relation; that is,

$$x^T\left(\dot{M}(q) - 2V(q,\dot{q})\right)x = 0, \quad \forall x \in \mathbb{R}^n. \tag{5}$$

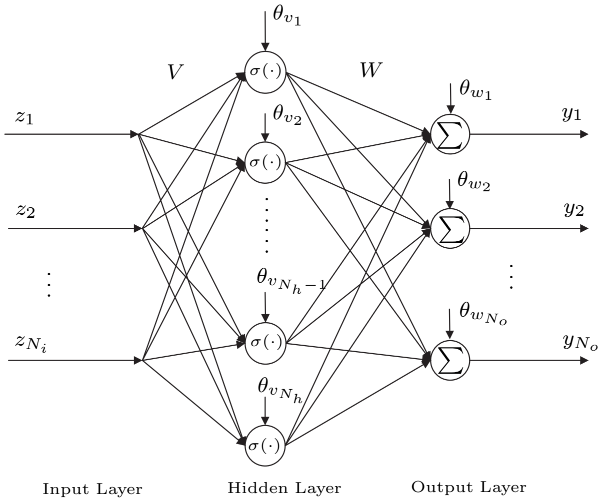

Fig. 1. The structure of the two-layer neural network.

2.2 Multilayer Neural Networks

An attractive ability of neural networks for control purpose is that they can approximate any smooth nonlinear function up to a small error. In this paper, a two-layer neural network shown in Fig. 1 is employed as the function approximator. The output of two-layer neural network can be determined by the following formula [Lewis et al., 1998]

$$y = W\sigma\left(V\bar{z}\right), \tag{6}$$

where N_i, N_h and N_o denote the numbers of input-layer neurons, hidden-layer neurons and output-layer neurons, respectively; $W \in \mathbb{R}^{N_o\times(N_h+1)}$, $V \in \mathbb{R}^{N_h\times(N_i+1)}$ are augmented weight matrices; $\bar{z} = [1, z^T]^T = [1, z_1, z_2, \cdots, z_{N_i}]^T$ is augmented input vector; $\sigma(V\bar{z}) = [1, \sigma(V_{r1}\bar{z}), \sigma(V_{r2}\bar{z}), \cdots, \sigma(V_{rN_h}\bar{z})]^T$ is the augmented activation function vector (V_{ri} represents the ith row of matrix V). By this augmented expression, any tuning of W and V will include tuning of the thresholds θ_{wi} and θ_{vj} as well. In this paper, following sigmoid function is adopted as the activation function,

$$\sigma(s) = \frac{1}{1+e^{-s}}. \tag{7}$$

Let S be a compact simply connected set of $\mathbb{R}^{N_i}$, and $g(z)$ be any smooth function from S to $\mathbb{R}^{N_o}$. Then, it can be shown that, for any given positive constant ε_N, there exist the ideal weight matrices W^* and V^* and the number of hidden layer neurons N_h such that

$$g(z) = W^*\sigma(V^*\bar{z}) + \varepsilon, \tag{8}$$

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

It should be noted that the ideal matrices W^* and V^* are only quantities required for analytical purpose. For real applications, their estimations $\hat{W}$ and $\hat{V}$ are used for the practical function approximation. The estimation of $g(z)$ can be given by

$$\hat{g}(z) = \hat{W}\sigma(\hat{V}\bar{z}). \tag{9}$$

Lemma 1. [Ge et al., 1999] For the neural network defined by (9), the function estimation error can be expressed as

$$\begin{aligned}\hat{g}(z) - g(z) &= \hat{W}\sigma(\hat{V}\bar{z}) - W^*\sigma(V^*\bar{z}) - \varepsilon \\ &= \tilde{W}\left(\hat{\sigma} - \hat{\sigma}'\hat{V}\bar{z}\right) + \hat{W}\hat{\sigma}'\tilde{V}\bar{z} + d_u, \end{aligned} \tag{10}$$

where $\hat{\sigma} = \sigma(\hat{V}\bar{z})$; $\hat{\sigma}' = \text{diag}\{0, \hat{\sigma}'_1, \hat{\sigma}'_2, \cdots, \hat{\sigma}'_{N_h}\}$ with $\hat{\sigma}'_i = d\sigma(s)/ds|_{s=\hat{V}_{ri}\bar{z}}$; the weight matrix estimation errors are $\tilde{W} = \hat{W} - W^*$ and $\tilde{V} = \hat{V} - V^*$; and the residual term is $d_u = \tilde{W}\hat{\sigma}'V^*\bar{z} + W^*O(\tilde{V}\bar{z})^2 - \varepsilon$, which is bounded by

$$\|d_u\| \le c_0 + c_1\|\bar{z}\| + c_2\|\tilde{W}\|_F\|\bar{z}\|, \tag{11}$$

where c_0, c_1 and c_2 are positive constants.

2.3 Stability of Systems

Definition 1. [Lewis et al., 1998] Given a nonlinear dynamical system

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

If there exists a compact set $U_x \in \mathbb{R}^n$ 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 $\|x(t)\| \le \delta$ for all $t \ge t_0 + T$, then the solution of the nonlinear dynamical system is called *uniformly ultimately bounded (UUB).*

3. ADAPTIVE NEURAL NETWORK CONTROLLER

The control objective is to develop a task-space tracking controller for the end-effector of robot manipulators 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 auxiliary controllers into the actual controller by "back stepping" through the system and reassembling it from its component subsystems [Kanellakopoulos et al., 1991].

First, a mild assumption, which always holds in practical applications, is stated as follows.

Assumption 1. Let $x_d(t) \in \mathbb{R}^m$ denote the desired task-space trajectory. It is assumed that $x_d(t)$ and its derivatives up to the second order are bounded in the sense, for instance, that

$$\left\| \begin{matrix} x_d(t) \\ \dot{x}_d(t) \\ \ddot{x}_d(t) \end{matrix} \right\| \le X_M, \tag{12}$$

where X_M is a known constant.

Define the tracking error $e_1(t) \in \mathbb{R}^m$ as

$$e_1 = x - x_d. \tag{13}$$

Differentiating $e_1(t)$ obtains

$$\dot{e}_1 = \dot{x} - \dot{x}_d = J(q, \phi_J)\dot{q} - \dot{x}_d. \tag{14}$$

In the presence of kinematic uncertainty, the parameter ϕ_J in the Jacobian matrix $J(q, \phi_J)$ is not known exactly. By replacing the uncertain parameter ϕ_J with its estimation $\hat{\phi}_J$, an approximate Jacobian $\hat{J}\left(q, \hat{\phi}_J\right) \in \mathbb{R}^{m\times n}$ can be obtained.

The first step of backstepping design is to treat $\dot{q}$ as an auxiliary control signal to the error dynamics defined by (14). This auxiliary signal is called as $\dot{q}_d$ and chosen as

$$\begin{aligned}\dot{q}_d =& \hat{J}^+\left(q, \hat{\phi}_J\right)(\dot{x}_d - K_1 e_1) + \\ & \left(I_n - \hat{J}^+\left(q, \hat{\phi}_J\right)\hat{J}\left(q, \hat{\phi}_J\right)\right)\lambda, \end{aligned} \tag{15}$$

where $\hat{J}^+\left(q, \hat{\phi}_J\right) = \hat{J}^T\left(q, \hat{\phi}_J\right)\left(\hat{J}\left(q, \hat{\phi}_J\right)\hat{J}^T\left(q, \hat{\phi}_J\right)\right)^{-1}$ is the generalized inverse matrix of the approximate Jacobian matrix; $K_1 \in \mathbb{R}^{m\times m}$ is a positive definite constant diagonal matrix; I_n is an n-dimensional unity matrix; $\lambda \in \mathbb{R}^n$ 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 is of full rank. This assumption is commonly adopted to deal with manipulator kinematic uncertainty in the existing literature [Cheah et al., 2004, 2006, Dixon, 2007]. Then the error dynamics defined by (14) can be rewritten as

$$\begin{aligned}\dot{e}_1 =& \left(J(q, \phi_J) - \hat{J}\left(q, \hat{\phi}_J\right)\right)\dot{q}_d + \hat{J}\left(q, \hat{\phi}_J\right)\dot{q}_d \\ & + J(q, \phi_J)e_2 - \dot{x}_d \\ =& -K_1 e_1 + J(q, \phi_J)e_2 + Y_J(q, \dot{q}_d)\left(\phi_J - \hat{\phi}_J\right) \\ =& -K_1 e_1 + J(q, \phi_J)e_2 - Y_J(q, \dot{q}_d)\tilde{\phi}_J, \end{aligned} \tag{16}$$

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

The second step is try to design the actual torque controller τ which makes e_2 as small as possible. To achieve this, the error dynamics for e_2 is derived by (4)

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

The torque controller τ is chosen as

$$\tau = \hat{F}_1 - K_2 e_2 - \gamma_1, \tag{18}$$

where K_2 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

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

It is noted that the term $-J^T(q, \phi_J)e_1$ in F_1 is used to compensate the coupling term $J(q, \phi_J)e_2$ 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 updating law. However, this assumption does not hold if the friction $F(\dot{q})$ has the particular nonlinear form [Ge et al., 2001]. Motivated by the universal approximation ability of neural networks, a multilayer neural network is employed to learn the uncertain function F_1. By the previous introduction for the multilayer neural network, it can be obtained that, over a compact set,

$$F_1 = W_1^* \sigma(V_1^* \bar{Z}_1) + \varepsilon_1, \tag{20}$$

with the approximation error ε_1 and neural network input $\bar{Z}_1 = [1, e_1^T, e_2^T, x_d^T, \dot{x}_d^T, \ddot{x}_d^T]^T$. The estimation of F_1 is given by

$$\hat{F}_1 = \hat{W}_1 \sigma(\hat{V}_1 \bar{Z}_1). \tag{21}$$

By Lemma 1, substituting (18), (19), (20) and (21) into (17) obtains that

$$\begin{aligned} & M(q)\dot{e}_2 + V(q, \dot{q})e_2 = \hat{F}_1 - K_2 e_2 - \gamma_1 - f_1 - \tau_{ed} \\ & = \hat{F}_1 - f_1 + J^T(q, \phi_J)e_1 - K_2 e_2 - J^T(q, \phi_J)e_1 - \gamma_1 - \tau_{ed} \\ & = -K_2 e_2 - J^T(q, \phi_J)e_1 - \gamma_1 + \delta_1 + \tilde{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}'_1\hat{V}_1\bar{Z}_1\right) \\ & \quad + \hat{W}_1\hat{\sigma}'_1\tilde{V}_1\bar{Z}_1, \end{aligned} \tag{22}$$

where $\hat{\sigma}_1 = \sigma(\hat{V}_1 \bar{Z}_1)$; $\hat{\sigma}'_1 = \sigma'(\hat{V}_1 \bar{Z}_1)$; $\delta_1 = -\tau_{ed} - \varepsilon_1 - \tilde{W}_1 \hat{\sigma}'_1 V_1^* \bar{Z}_1 - W_1^* O(\tilde{V}_1 \bar{Z}_1)^2$.
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 (By Lemma 1 and the stability result in the next section, it can be shown that δ_1 is indeed bounded). 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, the adaptive updating laws for the kinematic parameter $\hat{\phi}_J$ and neural network weight matrices $\hat{W}_1$ and $\hat{V}_1$ are derived as follows,

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

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

$$\dot{\hat{W}}_1 = \begin{cases} -e_2\left(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1\right)^T \Gamma_{w1}, \\ \text{if Tr}\left(\hat{W}_1 \Gamma_{w1}^{-1} \hat{W}_1^T\right) < W_{\mathrm{m}1}, \text{or if Tr}\left(\hat{W}_1 \Gamma_{w1}^{-1} \hat{W}_1^T\right) \\ = W_{\mathrm{m}1} \text{ and } e_2^T \hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1\right) > 0; \\ \dfrac{e_2^T \hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1\right)}{\text{Tr}\left(\hat{W}_1 \Gamma_{w1}^{-1} \hat{W}_1^T\right)} \hat{W}_1 - e_2\left(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1\right)^T \Gamma_{w1}, \\ \text{if Tr}\left(\hat{W}_1 \Gamma_{w1}^{-1} \hat{W}_1^T\right) = W_{\mathrm{m}1} \text{ and} \\ e_2^T \hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1\right) \leq 0; \end{cases} \tag{26}$$

$$\dot{\hat{V}}_1 = \begin{cases} -\left(e_2^T \hat{W}_1 \hat{\sigma}'_1\right)^T \bar{Z}_1^T \Gamma_{v1}, \text{if Tr}\left(\hat{V}_1 \Gamma_{v1}^{-1} \hat{V}_1^T\right) < V_{\mathrm{m}1}, \\ \text{or if Tr}\left(\hat{V}_1 \Gamma_{v1}^{-1} \hat{V}_1^T\right) = V_{\mathrm{m}1} \text{ and } e_2^T \hat{W}_1 \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1 > 0; \\ -\left(e_2^T \hat{W}_1 \hat{\sigma}'_1\right)^T \bar{Z}_1^T \Gamma_{v1} + \dfrac{e_2^T \hat{W}_1 \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1}{\text{Tr}\left(\hat{V}_1 \Gamma_{v1}^{-1} \hat{V}_1^T\right)} \hat{V}_1, \\ \text{if Tr}\left(\hat{V}_1 \Gamma_{v1}^{-1} \hat{V}_1^T\right) = V_{\mathrm{m}1} \text{ and } e_2^T \hat{W}_1 \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1 \leq 0; \end{cases} \tag{27}$$

where β_j is a given positive scalar; $W_{\mathrm{m}1}$ and $V_{\mathrm{m}1}$ are given positive constants for bounding neural network weight matrices; Γ_{w1} and Γ_{v1} are given positive definite diagonal matrices.
It is emphasized that the initial estimated kinematic parameter $\hat{\phi}_J(0)$ should be selected as

$$(\phi_J^-)_j \leq \left(\hat{\phi}_J(0)\right)_j \leq (\phi_J^+)_j, \tag{28}$$

and the initial neural network weight matrices $\hat{W}_1(0)$ and $\hat{V}_1(0)$ satisfy that

$$\text{Tr}\left(\hat{W}_1(0) \Gamma_{w1}^{-1} \hat{W}_1^T(0)\right) \leq W_{\mathrm{m}1}, \tag{29a}$$

$$\text{Tr}\left(\hat{V}_1(0) \Gamma_{v1}^{-1} \hat{V}_1^T(0)\right) \leq V_{\mathrm{m}1}. \tag{29b}$$

4. STABILITY ANALYSIS

Theorem 1. Given the robot manipulator defined by (2) and (4), if the controller is constructed as (18), the parameters updating laws are provided by (25), (26) and (27), and the initial values of estimated parameters satisfy the conditions (28) and (29), then e_1, e_2, $\hat{\phi}_J$, $\hat{W}_1$ and $\hat{V}_1$ are all uniformly ultimately bounded signals.

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 $\hat{W}_1$, let $L_{w1} = \text{Tr}\left(\hat{W}_1 \Gamma_{w1}^{-1} \hat{W}_1^T\right)$.
By (26), it follows that
1. when $L_{w1} = W_{\mathrm{m}1}$ and $e_2^T \hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1\right) > 0$,

$$\begin{aligned} \frac{dL_{w1}}{dt} &= 2\,\text{Tr}\left(\hat{W}_1 \Gamma_{w1}^{-1} \dot{\hat{W}}_1^T\right) \\ &= -2\,\text{Tr}\left(\hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1\right) e_2^T\right) \\ &= -2 e_2^T \hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1\right) < 0; \end{aligned}$$

2. when $L_{w1} = W_{\mathrm{m}1}$ and $e_2^T \hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1\right) \leq 0$,

$$\begin{aligned} \frac{dL_{w1}}{dt} &= 2\,\text{Tr}\left(\hat{W}_1 \Gamma_{w1}^{-1} \dot{\hat{W}}_1^T\right) \\ &= -2\,\text{Tr}\left(\hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1\right) e_2^T\right) + \\ &\quad 2\,\text{Tr}\left(\hat{W}_1 \Gamma_{w1}^{-1} \frac{e_2^T \hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1\right)}{\text{Tr}\left(\hat{W}_1 \Gamma_{w1}^{-1} \hat{W}_1^T\right)} \hat{W}_1^T\right) \\ &= -2 e_2^T \hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1\right) + 2 e_2^T \hat{W}_1(\hat{\sigma}_1 - \hat{\sigma}'_1 \hat{V}_1 \bar{Z}_1) \\ &= 0. \end{aligned}$$

Hence, if the initial neural network weight matrix $\hat{W}_1(0)$ satisfies (29a), then $\text{Tr}\left(\hat{W}_1 \Gamma_{w1}^{-1} \hat{W}_1^T\right) \leq W_{\mathrm{m}1}$ always holds, which means that $\hat{W}_1$ is bounded.
By the similar way, it can be proven that $\text{Tr}\left(\hat{V}_1 \Gamma_{v1}^{-1} \hat{V}_1^T\right) \leq V_{\mathrm{m}1}$ such that $\hat{V}_1$ is bounded.
To prove the uniform ultimate boundedness of error signals e_1 and e_2, the following Lyapunov function is considered,

$$V = V_1 + V_2, \tag{30}$$

where

$$V_1 = \frac{1}{2} e_1^T e_1 + \frac{1}{2} \tilde{\phi}_J^T \Gamma_\beta^{-1} \tilde{\phi}_J, \tag{31a}$$

$$V_2 = \frac{1}{2}\left(e_2^T M(q) e_2 + \text{Tr}\left(\tilde{W}_1 \Gamma_{w1}^{-1} \tilde{W}_1^T\right) + \text{Tr}\left(\tilde{V}_1 \Gamma_{v1}^{-1} \tilde{V}_1^T\right)\right) \tag{31b}$$

with $\Gamma_\beta = \text{diag}\,(\beta_1, \beta_2, \cdots, \beta_p) \in \mathbb{R}^{p \times p}$.
By (16) and (25), derivating V_1 with respect to time obtains that,

$$\begin{aligned} \dot{V}_1 &= e_1^T \dot{e}_1 + \tilde{\phi}_J^T \Gamma_\beta^{-1} \dot{\hat{\phi}}_J \\ &= -e_1^T K_1 e_1 + e_1^T J(q, \phi_J) e_2 - \tilde{\phi}_J^T (Y_J^T(q, \dot{q}_d) e_1 - \Gamma_\beta^{-1} \dot{\hat{\phi}}_J) \\ &\leq -e_1^T K_1 e_1 + e_1^T J(q, \phi_J) e_2. \end{aligned} \tag{32}$$

By (5), (22), (24) (26) and (27), the time derivative of V_2 is

$$\dot{V}_2 = e_2^T M(q)\dot{e}_2 + \frac{1}{2}e_2^T \dot{M}(q)e_2 + \mathrm{Tr}\left(\tilde{W}_1\Gamma_{w1}^{-1}\dot{\hat{W}}_1^T\right)$$
$$+ \mathrm{Tr}\left(\tilde{V}_1\Gamma_{v1}^{-1}\dot{\hat{V}}_1^T\right)$$
$$\leq \mathrm{Tr}\left(\tilde{W}_1\left(\hat{\sigma}_1 - e_2^T\hat{\sigma}_1'\hat{V}_1\bar{Z}_1\right)e_2^T\right) - e_2^T K_2 e_2 - e_2^T J^T(q,\phi_J)e_1$$
$$+ \mathrm{Tr}\left(\tilde{V}_1\bar{Z}_1 e_2^T\hat{W}_1\hat{\sigma}_1'\right) + \mathrm{Tr}\left(\tilde{W}_1\Gamma_{w1}^{-1}\dot{\hat{W}}_1^T\right)$$
$$+ \mathrm{Tr}\left(\tilde{V}_1\Gamma_{v1}^{-1}\dot{\hat{V}}_1^T\right) + \|e_2^T\|\|\delta_1\| - e_2^T\gamma_1$$
$$\leq \mathrm{Tr}\left(\tilde{V}_1\left(\bar{Z}_1 e_2^T\hat{W}_1\hat{\sigma}_1' + \Gamma_{v1}^{-1}\dot{\hat{V}}_1^T\right)\right) + \epsilon_1 - e_2^T J^T(q,\phi_J)e_1$$
$$+ \mathrm{Tr}\left(\tilde{W}_1\left(\left(\hat{\sigma}_1 - e_2^T\hat{\sigma}_1'\hat{V}_1\bar{Z}_1\right)e_2^T + \Gamma_{w1}^{-1}\dot{\hat{W}}_1^T\right)\right) - e_2^T K_2 e_2. \tag{33}$$

By (26), it follows that

1. If $\dot{\hat{W}}_1 = -\Gamma_{w1}e_2\left(\hat{\sigma}_1 - \hat{\sigma}_1'\hat{V}_1\bar{Z}_1\right)^T$, then
$$\mathrm{Tr}\left(\tilde{W}_1\left(\left(\hat{\sigma}_1 - e_2^T\hat{\sigma}_1'\hat{V}_1\bar{Z}_1\right)e_2^T + \Gamma_{w1}^{-1}\dot{\hat{W}}_1^T\right)\right) = 0.$$

2. If $\dot{\hat{W}}_1 = \dfrac{e_2^T\hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}_1'\hat{V}_1\bar{Z}_1\right)}{\mathrm{Tr}\left(\hat{W}_1\Gamma_{w1}^{-1}\hat{W}_1^T\right)}\hat{W}_1 - \Gamma_{w1}e_2\left(\hat{\sigma}_1 - \hat{\sigma}_1'\hat{V}_1\bar{Z}_1\right)^T$,

then $\mathrm{Tr}\left(\hat{W}_1\Gamma_{w1}^{-1}\hat{W}_1^T\right) = W_{\mathrm{m}1}$, $e_2^T\hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}_1'\hat{V}_1\bar{Z}_1\right) \leq 0$, and $\mathrm{Tr}\left(\tilde{W}_1\left(\left(\hat{\sigma}_1 - e_2^T\hat{\sigma}_1'\hat{V}_1\bar{Z}_1\right)e_2^T + \Gamma_{w1}^{-1}\dot{\hat{W}}_1^T\right)\right) =$
$$\frac{e_2^T\hat{W}_1\left(\hat{\sigma}_1 - \hat{\sigma}_1'\hat{V}_1\bar{Z}_1\right)}{\mathrm{Tr}\left(\hat{W}_1\Gamma_{w1}^{-1}\hat{W}_1^T\right)}\mathrm{Tr}\left(\tilde{W}_1\Gamma_{w1}^{-1}\hat{W}_1^T\right).$$

It is noted that
$$\mathrm{Tr}\left(\tilde{W}_1\Gamma_{w1}^{-1}\hat{W}_1^T\right) = \mathrm{Tr}\left(\tilde{W}_1\Gamma_{w1}^{-1}\tilde{W}_1^T\right) + \mathrm{Tr}\left(\tilde{W}_1\Gamma_{w1}^{-1}W_1^{*T}\right)$$
$$= \frac{1}{2}\mathrm{Tr}\left(\tilde{W}_1\Gamma_{w1}^{-1}\tilde{W}_1^T\right) + \frac{1}{2}\mathrm{Tr}\left(\hat{W}_1\Gamma_{w1}^{-1}\hat{W}_1^T\right)$$
$$- \frac{1}{2}\mathrm{Tr}\left(W_1^*\Gamma_{w1}^{-1}W_1^{*T}\right)$$
$$\geq 0,$$

where the facts that $\mathrm{Tr}\left(\hat{W}_1\Gamma_{w1}^{-1}\hat{W}_1^T\right) = W_{\mathrm{m}1}$, $W_{\mathrm{m}1} \geq \mathrm{Tr}\left(W_1^*\Gamma_{w1}^{-1}W_1^{*T}\right)$ and $\mathrm{Tr}\left(\tilde{W}_1\Gamma_{w1}^{-1}\tilde{W}_1^T\right) \geq 0$ have been used.

Therefore, it can be seen that in both cases the following fact holds
$$\mathrm{Tr}\left(\tilde{W}_1\left(\left(\hat{\sigma}_1 - e_2^T\hat{\sigma}_1'\hat{V}_1\bar{Z}_1\right)e_2^T + \Gamma_{w1}^{-1}\dot{\hat{W}}_1^T\right)\right) \leq 0.$$

By the similar way, it can be proven that
$$\mathrm{Tr}\left(\tilde{V}_1\left(\bar{Z}_1 e_2^T\hat{W}_1\hat{\sigma}_1' + \Gamma_{v1}^{-1}\dot{\hat{V}}_1^T\right)\right) \leq 0.$$

Hence, according to (33) and above two inequalities, it can be obtained that
$$\dot{V}_2 \leq -e_2^T K_2 e_2 - e_2^T J^T(q,\phi_J)e_1 + \epsilon_1. \tag{34}$$

Thus, by (32) and (34), the time derivative of Lyapunov function V is
$$\dot{V} = \dot{V}_1 + \dot{V}_2$$
$$\leq e_1^T J(q,\phi_J)e_2 - e_1^T K_1 e_1 - e_2^T K_2 e_2 - e_2^T J^T(q,\phi_J)e_1 + \epsilon_1$$
$$= -e^T K e + \epsilon_1$$
$$\leq -\lambda_{\min}(K)\|e\|^2 + \epsilon_1, \tag{35}$$

where $K = \mathrm{diag}(K_1, K_2)$, $e = \left(e_1^T, e_2^T\right)^T$, and $\lambda_{\min}(K)$ is the minimum eigenvalue of matrix K.

Table 1. The Denavit and Hartenberg parameters of PUMA 560 manipulator.

Link i	θ_i (rad)	a_i (rad)	α_i (m)	d_i (m)
1	q_1	$\pi/2$	0	0
2	q_2	0	α_2	0
3	q_3	$-\pi/2$	α_3	d_3
4	q_4	$\pi/2$	0	d_4
5	q_5	$-\pi/2$	0	0
6	q_6	0	0	0

Therefore, $\dot{V}$ is strictly negative outside the following compact set $\sum_e$
$$\sum\nolimits_e = \left\{e(t)\,\middle|\,0 \leq \|e(t)\| \leq \sqrt{\frac{\epsilon_1}{\lambda_{\min}(K)}}\right\}. \tag{36}$$

According to the Lyapunov theory extension [Lewis et al., 1998], this demonstrates that e_1, e_2 are both uniformly ultimately bounded signals. And it is easy to see that e_1, e_2 can be reduced as small as possible by choosing appropriate K and ϵ_1.

5. 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 by Corke and Armstrong-Helouvry [1994]. The initial joint configuration of PUMA 560 is $q(0) = [0,0,0,0,0,0]^T$rad and $\dot{q}(0) = [0,0,0,0,0,0]^T$rad/s. Table 1 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 kinematic parameters. In the proposed controller, they are estimated initially as $\hat{a}_2(0) = \hat{a}_3(0) = \hat{d}_3(0) = \hat{d}_4(0) = 0.1$m. The upper and lower limitations of the estimated parameters are $(0.5, 0.5, 0.5, 0.5)^T$ and $(0,0,0,0)^T$. The initial position of PUMA 560 end-effector is specified as $x(0) = (0.4521, -0.1500, 0.4318)^T$. The PUMA 560 end-effector is required to follow a predesigned straight line given by
$$x_d(1) = 0.01t + 0.4021,$$
$$x_d(2) = 0.03t - 0.2001,$$
$$x_d(3) = 0.02t + 0.3818.$$

Because the last three joints, q_4, q_5 and q_6, do no contribution to the position of manipulator end-effector, there is no need to consider them in the controller design procedure. In this simulation, the number of hidden neurons is 60; parameters of controller are set that $K_1 = \mathrm{diag}(5,5,5)$; $K_2 = \mathrm{diag}(50,50,50)$; $\lambda = 0$; $\tau_{ed} = (5\cos(\pi t/2), 4\sin(\pi t/2) + e^{-t}, 2\cos(t) + 3\sin(\pi t/3))^T$; $\Gamma_\beta = \mathrm{diag}(2,2,2,2)$; $N_h = 60$; $\hat{W}_i(0)$ and $\hat{V}_i(0)$ $(i = 1,2)$ are set to be zero matrices; $W_{\mathrm{m}1} = V_{\mathrm{m}1} = 1000$; $\delta_{M1} = 20$, $\epsilon_1 = 0.1$. The simulation results are shown in Figs. 2–4, which verifies the satisfactory tracking performance of the proposed controller.

6. 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.

REFERENCES

C. Abdallah, D. Dawson, P. Dorato, and M. Jamshidi. Survey of robust control for rigid robots. *IEEE Control Systems Magnize*, 11:24–30, 1991.

S. Arimoto. Robotics research toward explication of everyday physics. *International Journal of Robotics Research*, 18:1056–1063, 1999.

C.C. Cheah, S. Kawamura, and S. Arimoto. Feedback control for robotic manipulators with an uncertain jacobian matrix. *International Journal of Robotic Systems*, 12: 119–134, 1999.

C.C. Cheah, S. Kawamurab, and S. Arimotob. Stability of hybrid position and force control for robotic manipulator with kinematics and dynamics uncertainties. *Automatica*, 39:847–855, 2003.

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*, pages 3075–3080, 2004.

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*, 51:1024–1029, 2006.

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*, pages 1608–1613, 1994.

W.E. Dixon. Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics. *IEEE Transactions on Automatic Control*, 52: 488–493, 2007.

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

S.S. Ge, T.H. Lee, and C.J. Harris. *Adaptive Neural Network Control of Robotic Manipulators*. World Scientific, Singapore, 1998.

S.S. Ge, C.C. Huang, and T. Zhang. Nonlinear adaptive control using neural networks and its application to cstr systems. *Journal of Process Control*, 9:313–323, 1999.

S.S. Ge, T.H. Lee, and S.X. Ren. Adaptive friction compensation of servo mechanisms. *International Journal of Systems Science*, 32:523–532, 2001.

I. Kanellakopoulos, P.V. Kokotovic, and A.S. Morse. Systematic design of adaptive controllers for feedback linearizable systems. *IEEE Transactions on Automatic Control*, 36:1241–1253, 1991.

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

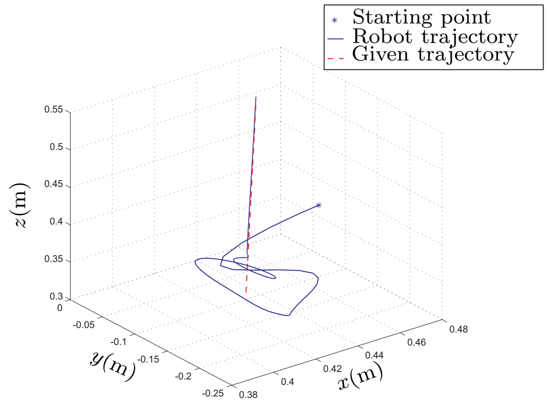

Fig. 2. The straight line trajectory tracking performance of PUMA 560 manipulator.

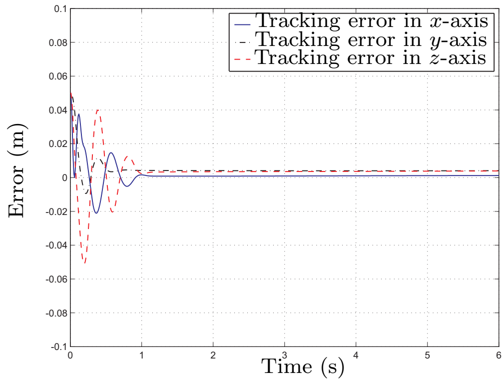

Fig. 3. The tracking errors in three coordinate axes with straight line trajectory.

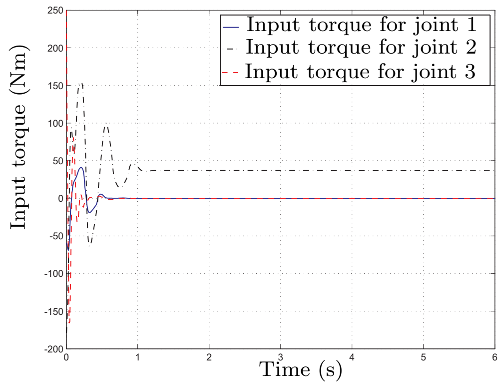

Fig. 4. Input torques for the first three joints.

G.P. Liu. *Nonlinear Identification and Control: a Neural Network Approach*. Springer, London, 2001.

R. Ortega and M.W. Spong. Adaptive motion control of rigid robots: a tutorial. *Automatica*, 25:877–888, 1989.