Automatica 45 (2009) 2312–2318

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model[☆]

Long Cheng, Zeng-Guang Hou*, Min Tan

Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 14 May 2008
Received in revised form
30 March 2009
Accepted 9 June 2009
Available online 24 July 2009

Keywords:
Manipulators
Tracking
Uncertainty
Neural networks

ABSTRACT

A neural-network-based adaptive controller is proposed for the tracking problem of manipulators with uncertain kinematics, dynamics and actuator model. The adaptive Jacobian scheme is used to estimate the unknown kinematics parameters. Uncertainties in the manipulator dynamics and actuator model are compensated by three-layer neural networks. External disturbances and approximation errors are counteracted by robust signals. The actuator controller is designed based on the backstepping scheme. Compared with the existing work, the proposed method considers the manipulator kinematics uncertainty, does not need the "linearity-in-parameters" assumption for the uncertain terms in the dynamics of manipulator and actuator, and guarantees the tracking error to be as small as desired. Finally, the performance of the proposed approach is illustrated by the simulation example.

1. Introduction

Recently, several adaptive controllers have been proposed to deal with the manipulator trajectory tracking problem in the presence of dynamics uncertainties (see the survey (Ortega & Spong, 1989)). However, a critical assumption in these controllers is that the uncertain term should satisfy the "linearity-in-parameters" condition. Moreover, tedious analysis and computations have to be done to determine the regressor matrix. To overcome these drawbacks, a class of neural-network-based adaptive approaches has been proposed for the manipulator tracking problem (Kwan, Lewis, & Dawson, 1998; Lewis, Jagannathan, & Yesildirek, 1998). For the general framework of this neural-network-based method, the readers are referred to Farrell and Polycarpou (2006).

It is noted that most existing controllers are designed for the joint trajectory tracking (Kwan et al., 1998; Lewis et al., 1998; Ortega & Spong, 1989). However, on many occasions, it is more convenient to drive the end-effector to follow a given trajectory in the Cartesian space. In this case, the manipulator kinematics should be considered. Due to the imprecise measurement of physical parameters and the interaction between manipulator and different environments, the kinematics parameters may not be known *a priori*. As reported in Arimoto (1999), the research on the control problem with uncertain kinematics is just a beginning. To deal with the kinematics uncertainty, some results have been published which are based on the approximate Jacobian technique (Cheah, Kawamura, & Arimoto, 2003; Dixon, 2007). However, these methods focus on the setpoint control of a robot. As to the tracking control, Cheah, Liu, and Slotine (2004) suggested an adaptive Jacobian approach for the non-redundant robot with uncertain kinematics and dynamics. Extensions to the redundant robots and unknown actuator parameters were made in Cheah, Liu, and Slotine (2006). Braganza, Dixon, Dawson, and Xian (2008) also presented a tracking controller for manipulators with uncertain kinematics and dynamics; the unit quaternion was used to represent the orientation of manipulator end-effector. It is noted that controllers proposed in Braganza et al. (2008), Cheah et al. (2003), Cheah et al. (2004), Cheah et al. (2006), and Dixon (2007) employed the traditional adaptive control scheme to deal with the uncertain dynamics of manipulator and actuator. Therefore, they suffer from the "linearity-in-parameters" assumption and other aforementioned drawbacks. In addition, external disturbances in manipulator dynamics have been neglected in the controller design.

This paper addresses the manipulator tracking problem in the presence of uncertain kinematics, dynamics, and actuator model. Adaptive Jacobin method, neural network approximation, and the backstepping method are employed to design the tracking controller. The contributions of this paper are: (1) the manipulator kinematics uncertainty is considered in the controller design; (2) compared with the previous work (Braganza et al., 2008; Cheah et al., 2003, 2004, 2006; Dixon, 2007), the "linearity-in-parameters" assumption for the uncertain dynamics of manip-

☆ The material in this paper was partially presented at the 17th IFAC World Congress, 2008, Seoul, Korea. This work was supported in part by the National Natural Science Foundation of China (Grants 60775043, 60725309, and 60621001) and the National Hi-Tech R&D Program (863) of China (Grant 2009AA04Z201). This paper was recommended for publication in revised form by Associate Editor Raul Ordóñez under the direction of Editor Miroslav Krstic.

* Corresponding address: Institute of Automation, Chinese Academy of Science, No. 95, Zhongguancun East Road, P.O. Box 2728, Beijing, 100190, China. Tel.: +86 10 62565502; fax: +86 10 62565502.

E-mail addresses: hou@compsys.ia.ac.cn, zengguang.hou@ia.ac.cn (Z.-G. Hou).

0005-1098/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2009.06.007

ulator and actuator is not necessary, and external disturbances in the dynamics of manipulator and actuator are taken into account; (3) the tracking error can be reduced as small as desired by choosing appropriate controller parameters. Therefore, the proposed method contributes to the current literature. This work is an extension to the conference papers (Cheng, Hou, & Tan, 2008; Cheng, Hou, Tan, & Wang, 2008), which considers the uncertain actuator model and further analyzes the tracking performance.

Notations. For a given vector, $\|\cdot\|$ denotes the vector Euclidean norm; for a given matrix, $\|\cdot\|_F$ denotes the matrix *Frobenius* norm; I_n denotes the n-dimensional unity matrix; $(\cdot)_i$ denotes the ith element of a given vector; $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ are the minimum and maximum eigenvalues of a given matrix, respectively; $\mathrm{Tr}(\cdot)$ denotes the trace operator.

2. Problem formulation and preliminaries

2.1. Manipulator-plus-actuator system description

The dynamics model for a rigid n-link, serially connected manipulator can be expressed as (Lewis et al., 1998)

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

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ denote the joint position, velocity, and acceleration vectors, respectively; $M(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix; $V(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is the centripetal-Coriolis matrix; $G(q) \in \mathbb{R}^n$ is the gravitational vector; $\tau_{ed} \in \mathbb{R}^n$ denotes the bounded unknown disturbance vector including unstructured unmodeled dynamics, and it is assumed that $\|\tau_{ed}\| \le \Delta_{\tau ed}$; $\tau \in \mathbb{R}^n$ represents the torque input vector. Two important properties of the dynamics equation described by (1) are given as follows (Lewis et al., 1998).

Property 1. *The inertia matrix $M(q)$ is symmetric and positive definite, and satisfies: $m_1\|y\|^2 \le y^{\mathrm{T}}M(q)y \le m_2\|y\|^2$, $\forall y \in \mathbb{R}^n$, where m_1 and m_2 are known positive constants.*

Property 2. *The time derivative of the inertia matrix and the centripetal-Coriolis matrix satisfy the skew symmetric relation; that is, $y^{\mathrm{T}}\left(\dot{M}(q) - 2V(q,\dot{q})\right)y = 0$, $\forall y \in \mathbb{R}^n$.*

For simplicity, it is assumed that the manipulator is driven by armature-controlled DC motors with voltages being input to amplifiers. The dynamics of this type of motor can be described as follows (Cheah et al., 2006)

$$\tau = K_T I, \tag{2a}$$

$$L\dot{I} + RI + K_e\dot{q} + u_{ed} = u, \tag{2b}$$

where $I \in \mathbb{R}^n$ is the armature current vector; $u \in \mathbb{R}^n$ is the armature voltage vector; $u_{ed} \in \mathbb{R}^n$ is the additive bounded voltage disturbance vector, and it is assumed that $\|u_{ed}\| \le \Delta_{ued}$; $K_T \in \mathbb{R}^{n\times n}$ is the positive definite constant diagonal matrix which characterizes the electro-mechanical conversion between current and torque; $R, L, K_e \in \mathbb{R}^{n\times n}$ are the positive definite constant diagonal matrices denoting the circuit resistance, circuit inductance, and voltage constant of the motor, respectively. And it is assumed that the following bounded condition holds

$$k_1\|x\|^2 \le x^{\mathrm{T}}K_T x \le k_2\|x\|^2, \quad \forall x \in \mathbb{R}^n, \tag{3}$$

where k_1 and k_2 are known positive constants.

By (2a) and (2b), it follows that

$$L\dot{\tau} + R\tau + K_T K_e\dot{q} + K_T u_{ed} = K_T u. \tag{4}$$

Let $x \in \mathbb{R}^m$ $(m \le n)$ represent the Cartesian space position vector which is related to the manipulator joint vector as $x = h(q)$, where $h(q) \in \mathbb{R}^m$ is the differentiable forward kinematics of the manipulator. The Cartesian space velocity $\dot{x}$ is related to joint velocity $\dot{q}$ as

$$\dot{x} = J(q,\phi_J)\dot{q}, \tag{5}$$

where $\phi_J \in \mathbb{R}^p$ represents the kinematics parameters, such as link lengths and joint offsets; $J(q,\phi_J) \stackrel{\text{def}}{=} (\partial h/\partial q) \in \mathbb{R}^{m\times n}$ 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*

$$J(q,\phi_J)\dot{q} = Y_J(q,\dot{q})\phi_J, \tag{6}$$

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

2.2. Multi-layer neural networks

The three-layer neural network, shown in Fig. 1, is usually used for the function approximation. The output of neural network can be determined as follows

$$y_i = \sum_{j=1}^{N_h}\left[w_{ij}\bar{\sigma}\left(\sum_{k=1}^{N_i} v_{jk}z_k + \theta_{vj}\right) + \theta_{wi}\right], \quad i = 1,\ldots,N_o, \tag{7}$$

where N_i, N_h and N_o denote the numbers of input-layer neurons, hidden-layer neurons and output-layer neurons, respectively; w_{ij} and v_{jk} are the adjustable synaptic weights, respectively. The threshold offsets are denoted by θ_{wi} and θ_{vj}; $\bar{\sigma}(\cdot)$ is the sigmoid activation function

$$\bar{\sigma}(s) = \frac{1}{1+\mathrm{e}^{-s}}. \tag{8}$$

For convenience, Eq. (7) can be rewritten in the following compact form

$$y = W\sigma(V\bar{z}), \tag{9}$$

where $W \in \mathbb{R}^{N_o\times(N_h+1)}$, $V \in \mathbb{R}^{N_h\times(N_i+1)}$ are augmented weight matrices; $\bar{z} = [1, z_1, z_2, \ldots, z_{N_i}]^{\mathrm{T}} \in \mathbb{R}^{N_i+1}$; $y = [y_1, y_2, \ldots, y_{N_o}]^{\mathrm{T}} \in \mathbb{R}^{N_o}$; $\sigma(V\bar{z}) = \left[1, \bar{\sigma}\left(V_{r_1}\bar{z}\right), \bar{\sigma}\left(V_{r_2}\bar{z}\right), \ldots, \bar{\sigma}\left(V_{r_{N_h}}\bar{z}\right)\right]^{\mathrm{T}} \in \mathbb{R}^{N_h+1}$ (V_{r_i} represents the ith row of matrix V). It is emphasized that $\sigma(\cdot)$ is a map from $\mathbb{R}^{N_h}$ to $\mathbb{R}^{N_h+1}$. By this augmented expression, θ_{wi} and θ_{vj} are included as the first columns of W and V, respectively. Therefore, any tuning of W and V will include tuning of the thresholds as well.

Let S be a compact simply connected set of $\mathbb{R}^{N_i}$, and $g(z)$ be a continuous function from S to $\mathbb{R}^{N_o}$. Then, for any given positive constant ε_N, there exist ideal parameters W^*, V^*, N_h such that

$$g(z) = W^*\sigma(V^*\bar{z}) + \varepsilon, \tag{10}$$

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

Assumption 1. The ideal neural network parameters are bounded by some positive values. That is $\|V^*\|_F \le V_M$ and $\|W^*\|_F \le W_M$.

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

$$\hat{g}(z) = \hat{W}\sigma(\hat{V}\bar{z}). \tag{11}$$

Lemma 1. *For the neural network defined by (11), the function approximation error is, $\hat{g}(z) - g(z) = \tilde{W}\left(\sigma(\hat{V}\bar{z}) - \hat{\sigma}'(\hat{V}\bar{z})\hat{V}\bar{z}\right) + \hat{W}\hat{\sigma}'(\hat{V}\bar{z})\tilde{V}\bar{z} + d_u$, where $\hat{\sigma}'(\hat{V}\bar{z}) = [\mathbf{0}, \mathrm{diag}\{\hat{\sigma}_1', \hat{\sigma}_2', \ldots, \hat{\sigma}_{N_h}'\}]^{\mathrm{T}} \in \mathbb{R}^{(N_h+1)\times N_h}$ with $\hat{\sigma}_i' = \mathrm{d}\bar{\sigma}(s)/\mathrm{d}s|_{s=\hat{V}_{ri}\bar{z}}$ and $\mathbf{0} = (0,0,\ldots,0)^{\mathrm{T}} \in \mathbb{R}^{N_h}$; It is emphasized that $\hat{\sigma}'(\cdot)$ is a map from $\mathbb{R}^{N_h}$ to $\mathbb{R}^{(N_h+1)\times N_h}$; the weight estimation errors are $\tilde{W} = \hat{W} - W^*$ and $\tilde{V} = \hat{V} - V^*$; and the residual term is $d_u = \tilde{W}\hat{\sigma}'(\hat{V}\bar{z})V^*\bar{z} + W^*O(\tilde{V}\bar{z})^2 - \varepsilon$, which is*