

DOI: 10.19322/j.cnki.issn.1006-4710.2016.04.011

# 基于模型分块逼近的三关节机器人鲁棒滑模控制

马莉丽, 钟 斌

(中国人民武装警察部队工程大学 装备工程学院, 陕西 西安 710086)

**摘要:** 三关节机器人结构参数、作业环境的外界干扰及结构振动等不确定因素均会造成其动力学模型不确定, 导致机器人关节位置镇定或轨迹跟踪控制器的设计具有一定的难度。为此, 设计三个RBF(Radical Basis Function)神经网络分别对机器人不确定模型中的三个不确定项进行分块逼近, 得到三个不确定项的估计信息, 从而得出机器人估计模型, 神经网络的权值采用适应算法。针对机器人估计模型设计鲁棒滑模控制律, 其中鲁棒项用于克服神经网络建模误差。通过定义Lyapunov函数, 证明了控制系统是稳定的。实验结果也表明了三关节均约在1s时达到期望位置或跟踪期望轨迹, 位置镇定误差或轨迹跟踪误差也快速、稳定地趋于零。

**关键词:** 三关节机器人; 模型分块逼近; 关节控制; RBF神经网络

中图分类号: TP242.2

文献标志码: A

文章编号: 1006-4710(2016)04-0437-06

## Research on three-joint robot's robust sliding mode control based on model's partitional approximating

MA Lili, ZHONG Bin

(College of Equipment Engineering, Engineering University of Chinese Armed Police Force, Xi'an 710086, China)

**Abstract:** Generally, the dynamic model of robot with three-joint is undetermined due to three-joint robot's uncertain structure parameters, working environment's external interfere and structural vibration. Accordingly, it is difficult to control the robot's joints' position stabilizing and trajectory tracking and controller's design due to the dynamic model's uncertainty. Therefore, three designed RBF(Radical Basis Function) neural networks are used to respectively model the three undetermined terms of the undetermined robot dynamic model, with partition approximating the three-joint robot. Three undetermined terms' estimation information is respectively obtained, with the robot's estimation model obtained. The neural networks' weights are obtained through the adaptive algorithm. The robust sliding mode control law is designed based on the robot's estimation model. The control law's robust term is used to overcome the neural networks' modeling error. The control system's stability is proved by defining Lyapunov function. The simulation experiments test verifies that three joints can trace ideal trajectory and reach an ideal position in 1s, and stabilization error and tracking error can fast and stably approximate to zero.

**Key words:** robot with three-joint; model's partitional approximating; joints' control; RBF neural network

三关节机器人(以下简称机器人)结构紧凑,所占空间小,灵活性强,工作空间较大,避障性好,广泛应用于工业机器人中。对机器人控制问题的研究一般集中于对其关节的位置控制,或者使每个关节能够按照期望的动态品质跟踪期望轨迹,或者使每个关节渐近稳定于指定的某个位置,即其控制问题归

纳为轨迹跟踪或位置镇定两类<sup>[1-5]</sup>。然而,机器人本身结构参数的变化或工作环境中的干扰往往导致机器人动力学模型的不确定,这就给控制器的设计带来了较大的难度<sup>[6-8]</sup>。通常针对这类问题采用的控制方法有滑模控制<sup>[9-11]</sup>、自适应控制、模糊控制<sup>[12]</sup>和鲁棒控制<sup>[13]</sup>。其中,滑模控制无需精确的动力学模

收稿日期: 2015-09-06

基金项目: 国家自然科学基金资助项目(51005246);中国人民武装警察部队工程大学基础研究基金资助项目(WJY201509)

作者简介: 马莉丽,女,博士,讲师,研究方向为机电系统智能控制及其自动化、机器人控制、军事装备理论及其应用等。  
E-mail: malilichina@163.com

型和专门的多变量解耦,只需根据轨迹跟踪或位置镇定误差合理设计滑模面,具有快速响应、无超调及鲁棒性强等优点,成为不确定机器人系统控制的首选方法。但该方法易导致控制器的输出幅值波动较大,所以滑模控制通常与其他控制方法相结合使用,才能得到较满意的控制效果,如文献[11]和文献[12]。与这些文献研究方法不同的是,本文采用了三个RBF神经网络分别对不确定机器人模型中三个不确定项进行建模,即采用分块逼近的方法得到其估计信息,神经网络的权值采用自适应算法。已有研究表明,RBF神经网络能以任意精度逼近任意非线性函数<sup>[14]</sup>。针对机器人估计模型,设计误差滑模函数和PI控制项及用于克服神经网络建模

误差的鲁棒项,即构成机器人的PI鲁棒滑模控制。利用Lyapunov稳定性理论证明了所设计的控制系统是全局渐近稳定的。实验结果也表明三关节均能达到稳定、快速的轨迹跟踪和位置镇定的控制要求。

## 1 三关节机器人动力学模型及问题的提出

三关节机器人结构示意图如图1所示。后臂质量为 $m_2$ ,前臂质量为 $m_3$ ,后臂长为 $l_2$ ,前臂长为 $l_3$ ,后臂质心到关节2的距离为 $r_2$ ,前臂质心到关节3的距离为 $r_3$ , $q_1$ 、 $q_2$ 和 $q_3$ 分别为关节1、关节2和关节3的转角,立柱转动惯量为 $I_1$ ,后臂转动惯量为 $I_2$ ,前臂转动惯量为 $I_3$ ,不计关节摩擦力矩。



图1 三关节机器人

Fig. 1 Robot with three joints

三关节机器人动力学模型为:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} \quad (1)$$

式中: $\mathbf{q}$ 、 $\dot{\mathbf{q}}$ 和 $\ddot{\mathbf{q}}$ 分别为关节转角、角速度和角加速度矢量, $\mathbf{q} = [q_1 \quad q_2 \quad q_3]^T$ ;  $\boldsymbol{\tau} = [\tau_1 \quad \tau_2 \quad \tau_3]^T$ 为关节

力矩矢量; $\mathbf{H}(\mathbf{q}) = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$ 为对称正定质量

惯性矩阵; $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$ 为哥氏力和离

心力矩阵; $\mathbf{G}(\mathbf{q}) = [g_1 \quad g_2 \quad g_3]^T$ 为重力矢量,其中:

$$h_{11} = I_1 + a_1 \cos^2(q_2) + a_2 \cos^2(q_2 + a_3) + 2a_3 \cos(q_2) \cos(q_2 + q_3)$$

$$h_{12} = h_{21} = h_{13} = h_{31} = 0$$

$$h_{22} = I_2 + a_1 + a_2 + 2a_3 \cos(q_3)$$

$$h_{23} = h_{32} = a_2 + a_3 \cos(q_3)$$

$$h_{33} = I_3 + a_2$$

$$c_{11} = -\frac{1}{2}a_1\dot{q}_2 \sin(2q_2) - \frac{1}{2}a_2(a_2 + \dot{q}_3) \sin(2q_2 + 2q_3) - a_3\dot{q}_2 \sin(2q_2 + q_3) - a_3\dot{q}_3 \cos(q_2) \sin(q_2 + q_3)$$

$$c_{12} = -\frac{1}{2}a_1\dot{q}_1 \sin(2q_2) - \frac{1}{2}a_2\dot{q}_1 \sin(2q_2 + 2q_3) - a_3\dot{q}_1 \sin(2q_2 + q_3)$$

$$c_{13} = -\frac{1}{2}a_1\dot{q}_1 \sin(2q_2 + 2q_3) - a_3\dot{q}_1 \cos(q_2) \sin(q_2 + q_3)$$

$$c_{21} = -c_{12}$$

$$c_{22} = -a_3\dot{q}_3 \sin(q_3)$$

$$c_{23} = -a_3(\dot{q}_2 + \dot{q}_3) \sin(q_3)$$

$$c_{31} = -c_{13}$$

$$c_{32} = -a_3\dot{q}_2 \sin(q_3)$$

$$c_{33} = 0$$

$$g_1 = 0$$

$$g_2 = b_1 \cos(q_2) + b_2 \cos(q_2 + q_3)$$

$$g_3 = b_2 \cos(q_2 + q_3)$$

其中, $a_1 = m_2 r_2^2 + m_3 l_2^2$ ,  $a_2 = m_3 r_3^2$ ,  $a_3 = m_3 r_3 l_2$ ,  $b_1 = (m_2 r_2 + m_3 l_2)g$ ,  $b_2 = m_3 r_3 g$ ,  $g$ 为重力加速度。

工程实际中,通常由于机器人本身结构参数、作业环境的外界干扰及结构振动模式等不确定性因素

的影响,导致式(1)所描述的机器人动力学模型是不精确的,即  $\mathbf{H}(\mathbf{q})$ 、 $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  和  $\mathbf{G}(\mathbf{q})$  三项是不精确的,甚至该三项是未知的,所以无法针对式(1)设计控制律。对此,可以设计三个 RBF 神经网络分别实现对  $\mathbf{H}(\mathbf{q})$ 、 $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  和  $\mathbf{G}(\mathbf{q})$  三项分块建模,并考虑到建模误差的存在,令:

$$\begin{cases} \mathbf{H}(\mathbf{q}) = \mathbf{H}_{nn}(\mathbf{q}) + \mathbf{E}_H \\ \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}_{nn}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{E}_C \\ \mathbf{G}(\mathbf{q}) = \mathbf{G}_{nn}(\mathbf{q}) + \mathbf{E}_G \end{cases} \quad (2)$$

式中:  $\mathbf{H}_{nn}(\mathbf{q})$ 、 $\mathbf{C}_{nn}(\mathbf{q}, \dot{\mathbf{q}})$  和  $\mathbf{G}_{nn}(\mathbf{q})$  分别为  $\mathbf{H}(\mathbf{q})$ 、 $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  和  $\mathbf{G}(\mathbf{q})$  的估计信息;  $\mathbf{E}_H$ 、 $\mathbf{E}_C$  和  $\mathbf{E}_G$  分别为  $\mathbf{H}(\mathbf{q})$ 、 $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  和  $\mathbf{G}(\mathbf{q})$  的建模误差。

现定义滑模函数:

$$\begin{cases} \mathbf{s} = \dot{\mathbf{e}} + \mathbf{\Lambda} \mathbf{e} \\ \dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d + \mathbf{\Lambda} \mathbf{e} \end{cases} \quad (3)$$

式中:  $\mathbf{\Lambda} = \lambda \times \mathbf{I}$ ,  $\lambda$  为正实数,  $\mathbf{I}$  为 3 阶单位阵; 位置误差  $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$ , 其中  $\mathbf{q}_d = [q_{d1} \quad q_{d2} \quad q_{d3}]^T$ , 为关节的期望位置或轨迹矢量, 其中  $q_{d1}$ 、 $q_{d2}$  和  $q_{d3}$  分别为关节 1、关节 2 和关节 3 的期望位置或轨迹。

于是,有:

$$\begin{cases} \dot{\mathbf{q}}_r = \mathbf{s} + \dot{\mathbf{q}} \\ \ddot{\mathbf{q}}_r = \dot{\mathbf{s}} + \ddot{\mathbf{q}} \end{cases} \quad (4)$$

由式(4),则式(1)左边变为  $\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{G}(\mathbf{q}) - \mathbf{H}(\mathbf{q})\dot{\mathbf{s}} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s}$ , 显然,由式(2),只要位置误差  $\mathbf{e} \rightarrow 0$ , 速度误差  $\dot{\mathbf{e}} \rightarrow 0$ , 则  $\mathbf{s} \rightarrow 0$ , 便有:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} \quad (5)$$

将式(2)代入式(5),得:

$$\mathbf{H}_{nn}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}_{nn}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{G}_{nn}(\mathbf{q}) + \mathbf{E} = \boldsymbol{\tau} \quad (6)$$

式中:  $\mathbf{E} = \mathbf{E}_H\ddot{\mathbf{q}}_r + \mathbf{E}_C\dot{\mathbf{q}}_r + \mathbf{E}_G$ 。

由式(6)可知,只要神经网络对  $\mathbf{H}(\mathbf{q})$ 、 $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  和  $\mathbf{G}(\mathbf{q})$  的建模误差  $\mathbf{E}_H \rightarrow 0$ 、 $\mathbf{E}_C \rightarrow 0$  和  $\mathbf{E}_G \rightarrow 0$ , 就可以针对式(6)所描述的机器人模型设计控制律。所以,问题的关键转化为分别对  $\mathbf{H}(\mathbf{q})$ 、 $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  和  $\mathbf{G}(\mathbf{q})$  设计三个 RBF 神经网络进行估计、逼近,并针对滑模函数(式(3))设计相应的控制律,在保证跟踪误差趋于零的同时,使神经网络的建模误差也趋于零。

## 2 RBF 神经网络算法及 PI 鲁棒滑模控制律设计

以神经网络对  $\mathbf{H}(\mathbf{q})$  的建模为例, RBF 神经网络算法为:

$$\begin{cases} \xi_{H_k}(\mathbf{e}) = \exp\left(-\frac{\|\mathbf{e} - \mathbf{x}_k\|^2}{2y_k^2}\right) & k = 1, 2, \dots, m \\ \mathbf{H}_{nn}(\mathbf{q}) = \{\mathbf{W}_H\}^T \boldsymbol{\Xi}_H(\mathbf{e}) \end{cases} \quad (7)$$

式中:  $m$  为神经网络神经元个数;  $\boldsymbol{\Xi}_H(\mathbf{e}) = [\xi_{H1}(\mathbf{e}), \xi_{H2}(\mathbf{e}), \dots, \xi_{Hm}(\mathbf{e})]$  为隐含层高斯基函数的输出;  $\mathbf{x}_k = [x_{k1}, x_{k2}, \dots, x_{k(2m)}]$  为第  $k$  个神经元的中心向量;  $\mathbf{y} = [y_1, y_2, \dots, y_m]^T$  为高斯基函数的基宽向量;  $\mathbf{W}_H = [\mathbf{W}_{H1}, \mathbf{W}_{H2}, \dots, \mathbf{W}_{Hm}]$  为输出层权值。

同理,有:

$$\begin{cases} \mathbf{C}_{nn}(\mathbf{q}, \dot{\mathbf{q}}) = \{\mathbf{W}_C\}^T \boldsymbol{\Xi}_C(\mathbf{e}, \dot{\mathbf{e}}) \\ \mathbf{G}_{nn}(\mathbf{q}) = \{\mathbf{W}_G\}^T \boldsymbol{\Xi}_G(\mathbf{e}) \end{cases} \quad (8)$$

式中:  $\boldsymbol{\Xi}_C(\mathbf{e}, \dot{\mathbf{e}}) = [\xi_{C1}(\mathbf{e}, \dot{\mathbf{e}}), \xi_{C2}(\mathbf{e}, \dot{\mathbf{e}}), \dots, \xi_{Cm}(\mathbf{e}, \dot{\mathbf{e}})]$ , 为位置误差与速度误差对应的隐含层高斯基函数的输出;  $\boldsymbol{\Xi}_G(\mathbf{e}) = [\xi_{G1}(\mathbf{e}), \xi_{G2}(\mathbf{e}), \dots, \xi_{Gm}(\mathbf{e})]$ , 为位置误差对应的隐含层高斯基函数的输出;  $\mathbf{W}_C = [\mathbf{W}_{C1}, \mathbf{W}_{C2}, \dots, \mathbf{W}_{Cm}]^T$ 、 $\mathbf{W}_G = [\mathbf{W}_{G1}, \mathbf{W}_{G2}, \dots, \mathbf{W}_{Gm}]^T$  分别为  $\boldsymbol{\Xi}_C(\mathbf{e}, \dot{\mathbf{e}})$  和  $\boldsymbol{\Xi}_G(\mathbf{e})$  对应的输出层权值。

三个神经网络均采用梯度下降训练方法,设神经网络对  $\mathbf{H}(\mathbf{q})$ 、 $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  和  $\mathbf{G}(\mathbf{q})$  的估计信息的自适应律为:

$$\begin{cases} \hat{\mathbf{H}}_{nn}(\mathbf{q}) = \{\hat{\mathbf{W}}_H\}^T \boldsymbol{\Xi}_H(\mathbf{e}) \\ \hat{\mathbf{C}}_{nn}(\mathbf{q}, \dot{\mathbf{q}}) = \{\hat{\mathbf{W}}_C\}^T \boldsymbol{\Xi}_C(\mathbf{e}, \dot{\mathbf{e}}) \\ \hat{\mathbf{G}}_{nn}(\mathbf{q}) = \{\hat{\mathbf{W}}_G\}^T \boldsymbol{\Xi}_G(\mathbf{e}) \end{cases} \quad (9)$$

式中:  $\hat{\mathbf{W}}_H$ 、 $\hat{\mathbf{W}}_C$  和  $\hat{\mathbf{W}}_G$  分别为  $\mathbf{W}_H$ 、 $\mathbf{W}_C$  和  $\mathbf{W}_G$  的估计权值。

设计  $\hat{\mathbf{W}}_H$ 、 $\hat{\mathbf{W}}_C$  和  $\hat{\mathbf{W}}_G$  的自适应律为:

$$\begin{cases} \dot{\hat{\mathbf{W}}}_H = \boldsymbol{\Gamma}_H \boldsymbol{\Xi}_H(\mathbf{e}) \ddot{\mathbf{q}}_r \mathbf{s} \\ \dot{\hat{\mathbf{W}}}_C = \boldsymbol{\Gamma}_C \boldsymbol{\Xi}_C(\mathbf{e}, \dot{\mathbf{e}}) \dot{\mathbf{q}}_r \mathbf{s} \\ \dot{\hat{\mathbf{W}}}_G = \boldsymbol{\Gamma}_G \boldsymbol{\Xi}_G(\mathbf{e}) \mathbf{s} \end{cases} \quad (10)$$

式中:  $\boldsymbol{\Gamma}_H$ 、 $\boldsymbol{\Gamma}_C$  和  $\boldsymbol{\Gamma}_G$  均为对称正定矩阵。

现设计如图 2 所示的控制系统,  $h_1, h_2, \dots, h_m$  表示  $m$  个神经元对应的隐含层,  $H_{nn1}(\cdot)$ 、 $H_{nn2}(\cdot)$ 、 $H_{nn3}(\cdot)$  为  $\mathbf{H}_{nn}(\mathbf{q})$  的 3 个标量函数,  $C_{nn1}(\cdot)$ 、 $C_{nn2}(\cdot)$ 、 $C_{nn3}(\cdot)$  为  $\mathbf{C}_{nn}(\mathbf{q}, \dot{\mathbf{q}})$  的 3 个标量函数,  $G_{nn1}(\cdot)$ 、 $G_{nn2}(\cdot)$ 、 $G_{nn3}(\cdot)$  为  $\mathbf{G}_{nn}(\mathbf{q})$  的 3 个标量函数。针对机器人估计模型的控制律为:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_m + \mathbf{K}_P \mathbf{s} + \mathbf{K}_I \int_0^t \mathbf{s} dt + \boldsymbol{\tau}_r \quad (11)$$

式中:  $\mathbf{K}_P$  和  $\mathbf{K}_I$  分别为比例系数和积分系数, 且  $\mathbf{K}_P > 0$ 、 $\mathbf{K}_I > 0$ ;  $\boldsymbol{\tau}_m$  为基于模型估计的控制律, 且

$$\boldsymbol{\tau}_m = \hat{\mathbf{H}}_{nn}(\mathbf{q})\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}_{nn}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \hat{\mathbf{G}}_{nn}(\mathbf{q}) \quad (12)$$

其中,  $\boldsymbol{\tau}_r$  为用于克服神经网络建模误差的鲁棒项, 且

$$\boldsymbol{\tau}_r = \mathbf{K}_r \text{sgn}(\mathbf{s}) \quad (13)$$

式中: 对角阵  $\mathbf{K}_r = \text{diag}[k_{r_i}]$ ,  $k_{r_i} \geq |E_i|$ ,  $E_i$  由式(6)中的  $\mathbf{E} = \mathbf{E}_H\ddot{\mathbf{q}}_r + \mathbf{E}_C\dot{\mathbf{q}}_r + \mathbf{E}_G$  确定,  $i$  为关节数,  $i = 1, 2, 3$ 。

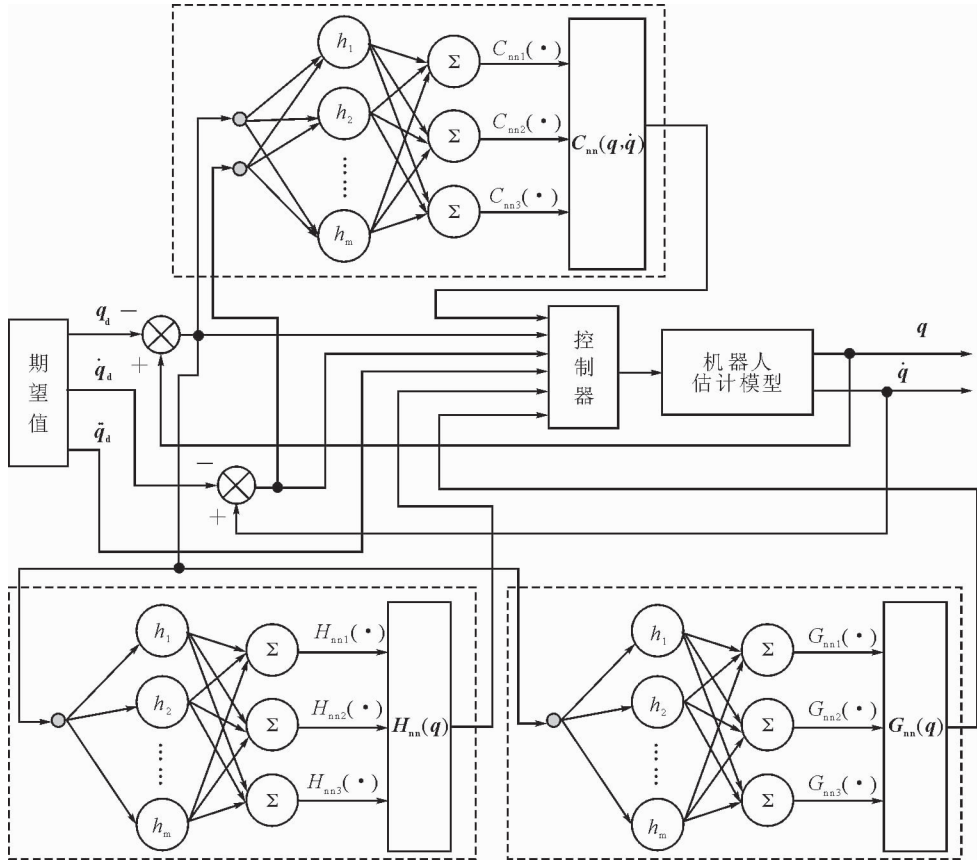


图 2 控制系统结构框图

Fig. 2 Control system's structure

### 3 控制系统稳定性分析

为分析控制系统的稳定性,现定义基于积分型的 Lyapunov 函数:

$$V = \frac{1}{2} s^T \mathbf{H}(\mathbf{q}) s + \frac{1}{2} \left( \int_0^t s dt \right)^T \mathbf{K}_I \left( \int_0^t s dt \right) + \frac{1}{2} \sum_{k=1}^m \{ \tilde{\mathbf{W}}_{Hk} \}^T \mathbf{\Gamma}_{Hk}^{-1} \tilde{\mathbf{W}}_{Hk} + \frac{1}{2} \sum_{k=1}^m \{ \tilde{\mathbf{W}}_{Ck} \}^T \mathbf{\Gamma}_{Ck}^{-1} \tilde{\mathbf{W}}_{Ck} + \frac{1}{2} \sum_{k=1}^m \{ \tilde{\mathbf{W}}_{Gk} \}^T \mathbf{\Gamma}_{Gk}^{-1} \tilde{\mathbf{W}}_{Gk} \quad (14)$$

式中:  $\mathbf{\Gamma}_{Hk}$ 、 $\mathbf{\Gamma}_{Ck}$  和  $\mathbf{\Gamma}_{Gk}$  均为对称正定矩阵; 误差项  $\tilde{\mathbf{W}}_{Hk} = \mathbf{W}_{Hk} - \hat{\mathbf{W}}_{Hk}$ 、 $\tilde{\mathbf{W}}_{Ck} = \mathbf{W}_{Ck} - \hat{\mathbf{W}}_{Ck}$ 、 $\tilde{\mathbf{W}}_{Gk} = \mathbf{W}_{Gk} - \hat{\mathbf{W}}_{Gk}$ , 其中  $\hat{\mathbf{W}}_{Hk}$ 、 $\hat{\mathbf{W}}_{Ck}$  和  $\hat{\mathbf{W}}_{Gk}$  分别为  $\mathbf{W}_{Hk}$ 、 $\mathbf{W}_{Ck}$  和  $\mathbf{W}_{Gk}$  的估计值,  $\mathbf{W}_{Hk}$ 、 $\mathbf{W}_{Ck}$  和  $\mathbf{W}_{Gk}$  分别为第  $k$  个神经元的权值。

显然,  $V$  是正定的, 现对  $V$  求一阶导数, 有:

$$\dot{V} = s^T [\mathbf{H}(\mathbf{q}) \dot{s} + \frac{1}{2} \dot{\mathbf{H}}(\mathbf{q}) s + \mathbf{K}_I \int_0^t s dt] + \sum_{k=1}^m \{ \tilde{\mathbf{W}}_{Hk} \}^T \mathbf{\Gamma}_{Hk}^{-1} \dot{\tilde{\mathbf{W}}}_{Hk} + \sum_{k=1}^m \{ \tilde{\mathbf{W}}_{Ck} \}^T \mathbf{\Gamma}_{Ck}^{-1} \dot{\tilde{\mathbf{W}}}_{Ck} + \sum_{k=1}^m \{ \tilde{\mathbf{W}}_{Gk} \}^T \mathbf{\Gamma}_{Gk}^{-1} \dot{\tilde{\mathbf{W}}}_{Gk} \quad (15)$$

根据机器人动力学方程(式(1))结构的斜对称性, 有  $s^T (\dot{\mathbf{H}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})) s = 0$ , 则:

$$\dot{V} = s^T [\mathbf{H}(\mathbf{q}) \dot{s} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) s + \mathbf{K}_I \int_0^t s dt] + \sum_{k=1}^m \{ \tilde{\mathbf{W}}_{Hk} \}^T \mathbf{\Gamma}_{Hk}^{-1} \dot{\tilde{\mathbf{W}}}_{Hk} + \sum_{k=1}^m \{ \tilde{\mathbf{W}}_{Ck} \}^T \mathbf{\Gamma}_{Ck}^{-1} \dot{\tilde{\mathbf{W}}}_{Ck} + \sum_{k=1}^m \{ \tilde{\mathbf{W}}_{Gk} \}^T \mathbf{\Gamma}_{Gk}^{-1} \dot{\tilde{\mathbf{W}}}_{Gk} \quad (16)$$

由式(1)、式(4)和式(8), 得:

$$\boldsymbol{\tau} = [\{ \mathbf{W}_H \}^T \mathbf{\Xi}_H(\mathbf{e})] \ddot{\mathbf{q}}_r + [\{ \mathbf{W}_C \}^T \mathbf{\Xi}_C(\mathbf{e}, \dot{\mathbf{e}})] \dot{\mathbf{q}}_r + [\{ \mathbf{W}_G \}^T \mathbf{\Xi}_G(\mathbf{e})] - \mathbf{H}(\mathbf{q}) \dot{s} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) s + \mathbf{E} \quad (17)$$

再由式(9)、式(11)~式(13), 得:

$$\boldsymbol{\tau} = [\{ \hat{\mathbf{W}}_H \}^T \mathbf{\Xi}_H(\mathbf{e})] \ddot{\mathbf{q}}_r + [\{ \hat{\mathbf{W}}_C \}^T \mathbf{\Xi}_C(\mathbf{e}, \dot{\mathbf{e}})] \dot{\mathbf{q}}_r + [\{ \hat{\mathbf{W}}_G \}^T \mathbf{\Xi}_G(\mathbf{e})] + \mathbf{K}_P s + \mathbf{K}_I \int_0^t s dt + \mathbf{K}_r \text{sgn}(s) \quad (18)$$

于是, 由式(17)和式(18), 得:

$$\mathbf{H}(\mathbf{q}) \dot{s} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) s + \mathbf{K}_I \int_0^t s dt = -\mathbf{K}_P s - \mathbf{K}_r \text{sgn}(s) + [\{ \tilde{\mathbf{W}}_H \}^T \mathbf{\Xi}_H(\mathbf{e})] \ddot{\mathbf{q}}_r + [\{ \tilde{\mathbf{W}}_C \}^T \mathbf{\Xi}_C(\mathbf{e}, \dot{\mathbf{e}})] \dot{\mathbf{q}}_r + [\{ \tilde{\mathbf{W}}_G \}^T \mathbf{\Xi}_G(\mathbf{e})] + \mathbf{E} \quad (19)$$

式中: 误差项  $\tilde{\mathbf{W}}_H = \mathbf{W}_H - \hat{\mathbf{W}}_H$ 、 $\tilde{\mathbf{W}}_C = \mathbf{W}_C - \hat{\mathbf{W}}_C$ 、 $\tilde{\mathbf{W}}_G = \mathbf{W}_G - \hat{\mathbf{W}}_G$ , 其中  $\hat{\mathbf{W}}_H$ 、 $\hat{\mathbf{W}}_C$  和  $\hat{\mathbf{W}}_G$  分别为  $\mathbf{W}_H$ 、 $\mathbf{W}_C$  和  $\mathbf{W}_G$  的估计值。

将式(19)代入式(16), 得:

$$\begin{aligned} \dot{V} = & \mathbf{s}^T \mathbf{K}_p \mathbf{s} + \mathbf{s}^T \mathbf{E} + \mathbf{s}^T \mathbf{K}_r \operatorname{sgn}(\mathbf{s}) + \\ & \mathbf{s}^T [\{\tilde{\mathbf{W}}_H\}^T \{\boldsymbol{\xi}_H(\mathbf{e})\} \ddot{\mathbf{q}}_r + \mathbf{s}^T [\{\tilde{\mathbf{W}}_C\}^T \{\boldsymbol{\xi}_C(\mathbf{e}, \dot{\mathbf{e}})\} \dot{\mathbf{q}}_r + \\ & \mathbf{s}^T [\{\tilde{\mathbf{W}}_G\}^T \{\boldsymbol{\xi}_G(\mathbf{e})\}] + \sum_{k=1}^m \{\tilde{\mathbf{W}}_{Hk}\}^T \boldsymbol{\Gamma}_{Hk}^{-1} \tilde{\mathbf{W}}_{Hk} + \\ & \sum_{k=1}^m \{\tilde{\mathbf{W}}_{Ck}\}^T \boldsymbol{\Gamma}_{Ck}^{-1} \tilde{\mathbf{W}}_{Ck} + \sum_{k=1}^m \{\tilde{\mathbf{W}}_{Gk}\}^T \boldsymbol{\Gamma}_{Gk}^{-1} \tilde{\mathbf{W}}_{Gk} \end{aligned} \quad (20)$$

由于:

$$\begin{aligned} \mathbf{s}^T [\{\tilde{\mathbf{W}}_H\}^T \{\boldsymbol{\xi}_H(\mathbf{e})\} \ddot{\mathbf{q}}_r = \\ \begin{bmatrix} s_1 & s_2 & \cdots & s_m \end{bmatrix} \begin{bmatrix} \{\tilde{\mathbf{W}}_{H1}\}^T \boldsymbol{\xi}_{H1}(\mathbf{e}) \ddot{\mathbf{q}}_r \\ \{\tilde{\mathbf{W}}_{H2}\}^T \boldsymbol{\xi}_{H2}(\mathbf{e}) \ddot{\mathbf{q}}_r \\ \vdots \\ \{\tilde{\mathbf{W}}_{Hm}\}^T \boldsymbol{\xi}_{Hm}(\mathbf{e}) \ddot{\mathbf{q}}_r \end{bmatrix} = \\ \sum_{k=1}^m \{\tilde{\mathbf{W}}_{Hk}\}^T \boldsymbol{\xi}_{Hk}(\mathbf{e}) \ddot{\mathbf{q}}_r s_k \end{aligned} \quad (21)$$

同理:

$$\begin{cases} \mathbf{s}^T [\{\tilde{\mathbf{W}}_C\}^T \{\boldsymbol{\xi}_C(\mathbf{e}, \dot{\mathbf{e}})\} \dot{\mathbf{q}}_r = \sum_{k=1}^m \{\tilde{\mathbf{W}}_{Ck}\}^T \boldsymbol{\xi}_{Ck}(\mathbf{e}, \dot{\mathbf{e}}) \dot{\mathbf{q}}_r s_k \\ \mathbf{s}^T [\{\tilde{\mathbf{W}}_G\}^T \{\boldsymbol{\xi}_G(\mathbf{e})\}] = \sum_{k=1}^m \{\tilde{\mathbf{W}}_{Gk}\}^T \boldsymbol{\xi}_{Gk}(\mathbf{e}) s_k \end{cases} \quad (22)$$

式中:  $s_1, s_2, \dots, s_m$  表示滑模函数  $\mathbf{s}$  的各项;  $\boldsymbol{\xi}_{Hk}(\mathbf{e})$ 、 $\boldsymbol{\xi}_{Ck}(\mathbf{e}, \dot{\mathbf{e}})$ 、 $\boldsymbol{\xi}_{Gk}(\mathbf{e})$  和  $s_k$  分别表示第  $k$  个神经元输出和滑模函数。

将式(21)和式(22)代入式(20),得:

$$\begin{aligned} \dot{V} = & -\mathbf{s}^T \mathbf{K}_p \mathbf{s} + \mathbf{s}^T \mathbf{E} - \mathbf{s}^T \mathbf{K}_r \operatorname{sgn}(\mathbf{s}) + \\ & \sum_{k=1}^m \{\tilde{\mathbf{W}}_{Hk}\}^T \boldsymbol{\xi}_{Hk}(\mathbf{e}) \ddot{\mathbf{q}}_r s_k + \sum_{k=1}^m \{\tilde{\mathbf{W}}_{Ck}\}^T \boldsymbol{\xi}_{Ck}(\mathbf{e}, \dot{\mathbf{e}}) \dot{\mathbf{q}}_r s_k + \\ & \sum_{k=1}^m \{\tilde{\mathbf{W}}_{Gk}\}^T \boldsymbol{\xi}_{Gk}(\mathbf{e}) s_k + \sum_{k=1}^m \{\tilde{\mathbf{W}}_{Hk}\}^T \boldsymbol{\Gamma}_{Hk}^{-1} \tilde{\mathbf{W}}_{Hk} + \\ & \sum_{k=1}^m \{\tilde{\mathbf{W}}_{Ck}\}^T \boldsymbol{\Gamma}_{Ck}^{-1} \tilde{\mathbf{W}}_{Ck} + \sum_{k=1}^m \{\tilde{\mathbf{W}}_{Gk}\}^T \boldsymbol{\Gamma}_{Gk}^{-1} \tilde{\mathbf{W}}_{Gk} \end{aligned} \quad (23)$$

由于  $\dot{\tilde{\mathbf{W}}}_{Hk} = -\dot{\mathbf{W}}_{Hk}$ ,  $\dot{\tilde{\mathbf{W}}}_{Ck} = -\dot{\mathbf{W}}_{Ck}$ ,  $\dot{\tilde{\mathbf{W}}}_{Gk} = -\dot{\mathbf{W}}_{Gk}$ , 并将式(10)代入式(23),则

$$\dot{V} = -\mathbf{s}^T \mathbf{K}_p \mathbf{s} + \mathbf{s}^T \mathbf{E} - \mathbf{s}^T \mathbf{K}_r \operatorname{sgn}(\mathbf{s}) \quad (24)$$

由于  $k_{rvi} \geq |E_i|$ , 即  $\mathbf{K}_r \geq \mathbf{E}$ , 则:

$$\dot{V} \leq -\mathbf{s}^T \mathbf{K}_p \mathbf{s} \leq 0 \quad (25)$$

显然,由式(25), $\dot{V}$  是负定的,控制系统是全局渐近稳定的,即机器人关节在所设计的控制律式(11)的作用下,能够渐近稳定地跟踪期望轨迹或镇定期望位置。

## 4 实验验证

机器人参数:  $m_2 = 30 \text{ kg}$ ,  $m_3 = 26 \text{ kg}$ ;  $r_2 = 0.6 \text{ m}$ ,  $r_3 = 0.5 \text{ m}$ ;  $I_1 = 3.61 \text{ (kg} \cdot \text{m}^2)$ ,  $I_2 = 2.35 \text{ (kg} \cdot \text{m}^2)$ ,  $I_3 = 1.95 \text{ (kg} \cdot \text{m}^2)$ 。

关节初始值:

$$\begin{aligned} \mathbf{q} &= [-0.4 \quad 1.5 \quad -1.5]^T \\ \dot{\mathbf{q}} &= [0.0 \quad 0.0 \quad 0.0]^T \end{aligned}$$

三关节位置镇定期望指令:

$$\mathbf{q}_d = \begin{cases} [1 \quad 2 \quad 3]^T [-2t^3 + 3t^2] & (0 \leq t \leq 1) \\ [1 \quad 2 \quad 3]^T & (t > 1) \end{cases}$$

轨迹跟踪期望指令:

$$\mathbf{q}_d = [0.5 \quad 1 \quad 1.5]^T [\sin(\pi t)]$$

RBF 神经网络的神经元个数  $m=5$ , 高斯基函数基宽为 3。控制器参数:  $\mathbf{K}_p = \mathbf{K}_i = 100 \times \mathbf{I}$ ,  $\mathbf{A} = 5 \times \mathbf{I}$ 。

三关节的位置镇定结果如图 3 所示。从图中可以看出,三关节约在 1 s 时达到并稳定在期望位置,也能在 1 s 时跟踪期望轨迹,实验还表明三关节也能在 1 s 时跟踪期望速度。轨迹跟踪结果如图 4 所示。为了验证本文控制方法的有效性和优势,与文献[14]所采用的 RBF 神经网络滑模控制方法进行了对比实验,限于篇幅,仅给出了如图 5 所示的关节位置镇定控制结果。通过对比图 3 和图 5 可以看出,采用文献[14]的控制方法时,三个关节约在 4 s 时才能达到并稳定在期望位置。

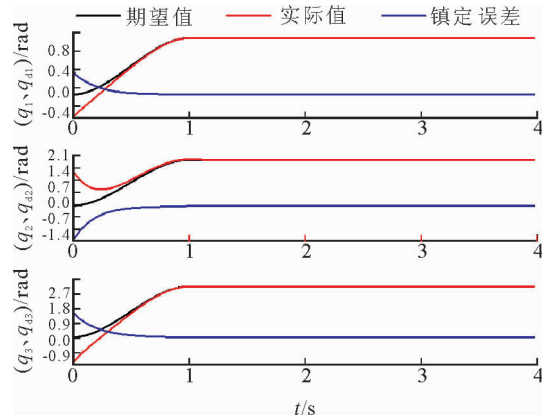


图 3 采用本文控制方法的关节位置镇定结果  
Fig. 3 Control of position stabilizing with method of this paper

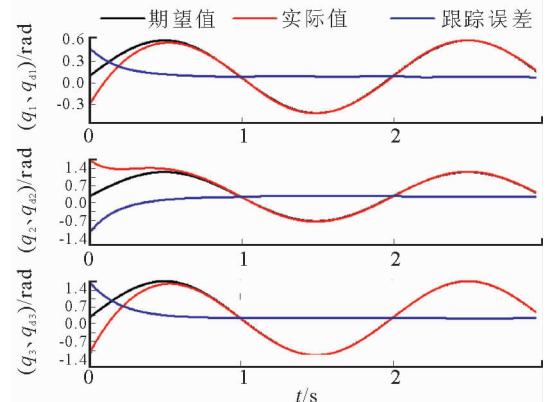


图 4 采用本文控制方法的关节轨迹跟踪结果  
Fig. 4 Control of trajectory tracking with method of this paper

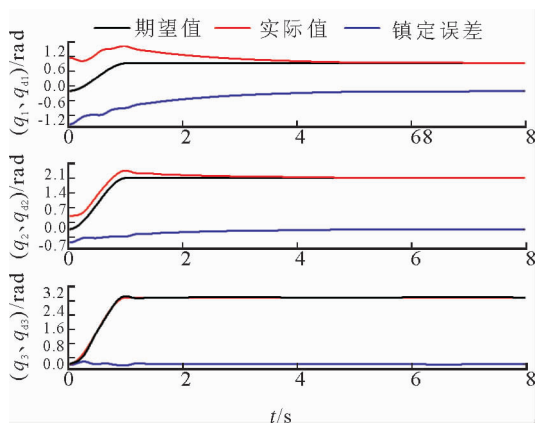


图5 采用文献[14]的控制方法时关节位置镇定结果  
Fig.5 Control of position stabilizing with control method of reference [14]

## 5 结论

为了解决三关节机器人不确定动力学模型的关节位置镇定或轨迹跟踪控制器设计较困难的问题,通过设计三个RBF神经网络,分别对机器人不确定模型中的三个不确定项进行分块逼近,得到机器人估计模型。针对机器人的估计模型设计了鲁棒滑模控制律,通过本文的研究得出了以下主要结论:

- 1) 通过定义Lyapunov函数,证明了控制系统是全局渐近稳定的;
- 2) 仿真实验表明,三关节机器人的各关节均约在1s时达到期望位置或跟踪期望轨迹,其位置镇定误差或轨迹跟踪误差也约在1s时渐近地趋于零。

### 参考文献:

[1] 季晔,刘宏昭,原大宁,等. 并联机构位置正解方法研究[J]. 西安理工大学学报, 2010, 26(3): 277-282.  
JI Ye, LIU Hongzhao, YUAN Daning. Research on approach of forward positional analysis of parallel manipulator[J]. Journal of Xi'an University of Technology, 2010, 26(3): 277-282.

[2] HARMON F G, FRANK A A, JOSHI S S. The control of a parallel hybrid-electric propulsion system for a small unmanned aerial vehicle using a CMAC neural network[J]. Neural Networks, 2012, 18(5): 772-780.

[3] ODA M. Space robot experiments on NASDA'S ETS-VII satellite; preliminary overview of the experiment results[C]// IEEE International Conference on Robotics and Automation. Piscataway, NJ, USA: IEEE, 2011: 1390-1395.

[4] 赵涛,刘明雍,周良荣. 基于Lyapunov方法的轮式移动机器人全局轨迹跟踪控制[J]. 火力与指挥控制, 2010, 35(7): 87-89.  
ZHAO Tao, LIU Mingyong, ZHOU Liangrong. Lyapunov based global trajectory tracking control for wheeled mobile robot [J]. Fire Control & Command Control, 2010, 35(7): 87-89.

[5] 赵涛,刘明雍,周良荣. 移动机器人的轨迹跟踪控制新方法[J]. 火力与指挥控制, 2010, 35(11): 33-34, 46.  
ZHAO Tao, LIU Mingyong, ZHOU Liangrong. Novel method for trajectory tracking control of mobile robot [J]. Fire Control & Command Control, 2010, 35(11): 33-34, 46.

[6] LI Z J, ZHANG Y N, YANG Y P. Support vector machine optimal control for mobile wheeled inverted pendulums with unmodeled dynamics [J]. Neurocomputing, 2011, 13(73): 2773-2782.

[7] BANERJEE A, BHATTACHARYA B, MALLIK A K. Large deformation of cantilever beams with geometric non-linearity: analytical and numerical approaches [J]. International Journal of Non-linear Mechanics, 2010, 43(5): 366-376.

[8] 田慧慧,苏玉鑫. 机器人系统输出反馈重复学习轨迹跟踪控制[J]. 控制与决策, 2012, 27(11): 1756-1760.  
TIAN Huihui, SU Yuxin. Global asymptotic stable repetitive output feedback tracking control of robot manipulators [J]. Control and Decision, 2012, 27(11): 1756-1760.

[9] UTKIN V I. Sliding mode control design principles and applications to electric drives [J]. IEEE Trans and Electron, 2010, 40(1): 23-26.

[10] LI T, HSENG S, HUANG Y C. MIMO adaptive fuzzy terminal sliding-mode controller for robotic manipulators [J]. Information Sciences, 2010, 180(6): 4641-4660.

[11] DELAVARI H, GHADERI R, RANJBAR A, et al. Fuzzy fractional order sliding mode controller for non-linear systems [J]. Commun in Nonlinear Sci Numer and Simulat, 2010, 15(4): 963-978.

[12] 胡盛斌,陆敏恂. 多个关节机器人反馈线性化双模糊滑模控制[J]. 机械科学与技术, 2013, 32(1): 105-110.  
HU Shengbin, LU Minxun. Feedback linearization double fuzzy sliding mode control for multi-links robot [J]. Mechanical Science and Technology for Aerospace Engineering, 2013, 32(1): 105-110, 115.

[13] ISLAM S, LIU P X. Robust adaptive fuzzy output feedback control system for robot manipulators [J]. IEEE/ASME Trans on Mechatronics, 2011, 16(2): 288-296.

[14] 邵克勇,马千惠,邹运,等. 基于径向基函数神经网络的多关节机器人滑模控制器[J]. 计算机测量与控制, 2014, 22(5): 1385-1387.  
SHAO Keyong, MA Qianhui, ZOU Yun, et al. Sliding mode control for multi-joint robot based on RBF neural network [J]. Computer Measurement & Control, 2014, 22(5): 1385-1387.