

基于参考轨迹实时修正的机器人轨迹跟踪控制方法

肖帆¹ 李公法¹ 章晓峰¹ 陶波² 蒋国璋² 李光³

(1. 武汉科技大学冶金装备及其控制教育部重点实验室, 武汉 430081; 2. 武汉科技大学精密制造研究院, 武汉 430081; 3. 湖南工业大学机械工程学院, 株洲 412007)

摘要: 针对动力学不确定性的机器人轨迹跟踪问题, 本文提出一种基于参考轨迹实时修正的机器人轨迹跟踪控制方法。将轨迹跟踪中已产生的误差进行累加, 实时前馈补偿到参考轨迹上即将被跟踪的点上。给出了该方法的控制框图, 由控制框图导出跟踪误差与命令误差之间的关系式。关系式表明只需控制器中的控制算法保证速度误差稳定, 即可保证跟踪误差收敛。此外, 提高补偿增益的值可以提高误差的收敛速度。分析 PD 控制律能满足所提方法的收敛条件。给出了所提方法中参数的调节方案。通过仿真和实验验证了该方法的有效性。实验结果表明, 各个关节跟踪轨迹 1 得到的误差绝对值均不大于 0.008 7 rad; 跟踪轨迹 2 得到的误差绝对值均不大于 0.005 9 rad。

关键词: 机器人; 轨迹跟踪; 参考轨迹; PD 控制

中图分类号: TP242 文献标识码: A 文章编号: 1000-1298(2023)02-0441-09 OSID: 

Robot Trajectory Tracking Control Based on Reference Trajectory Real-time Modification

XIAO Fan¹ LI Gongfa¹ ZHANG Xiaofeng¹ TAO Bo² JIANG Guozhang² LI Guang³

(1. Key Laboratory of Metallurgical Equipment and Control Technology, Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China
2. Institute of Precision Manufacturing, Wuhan University of Science and Technology, Wuhan 430081, China
3. College of Mechanical Engineering, Hunan University of Technology, Zhuzhou 412007, China)

Abstract: A simple and effective control method for robot trajectory tracking with uncertain dynamics was presented. The core idea of the proposed method was to modify the reference trajectory in real-time. The main operation of this method was to accumulate the generated tracking errors and compensate them feedforward in real-time to the points to be tracked on the reference trajectory. Firstly, the control block diagram for the proposed method was showed. Then, the equation for the relationship between the tracking error and the command error was derived from the control block diagram. The equation showed that the control algorithm in the controller only needed to ensure that the velocity error was stable and the tracking error would converge. The increase in compensation gain can also accelerate the convergence of the error. Subsequently, the convergence condition that PD control law can satisfy the proposed method was analyzed. At the same time, the adjustment scheme of parameters in the proposed method was given. Finally, the effectiveness of the proposed method was verified by simulation and physical experiment. In the physical experiment, the absolute value of error obtained from tracking trajectory 1 of each joint was no more than 0.008 7 rad; the absolute value of error obtained from tracking trajectory 2 of each joint was no more than 0.005 9 rad.

Key words: robot; trajectory tracking; reference trajectory; PD control

0 引言

机器人可以代替人类从事危险和重复性的工

作, 其已被广泛应用于工业、农业、服务等行业中。机器人的任务也从搬运、码垛等拓展到了 3D 打印、弧焊、喷涂等对轨迹跟踪精度要求更高的任务中。

收稿日期: 2022-03-24 修回日期: 2022-04-27

基金项目: 国家自然科学基金项目(52075530, 51575407, 51505349, 61733011, 41906177)

作者简介: 肖帆(1988—), 男, 博士生, 主要从事机器人运动控制研究, E-mail: xiaofan@wust.edu.cn

通信作者: 李公法(1979—), 男, 教授, 博士生导师, 主要从事机器人运动控制、机器视觉和图像处理研究, E-mail: ligongfa@wust.edu.cn

在完成这些任务时,通常采用逆运动学^[1]将机器人的笛卡尔轨迹跟踪问题转换为关节轨迹跟踪控制问题。然而,由于机器人的动力学高度非线性的本质及关节摩擦力等因素,使得机器人的轨迹跟踪控制问题变得困难。

机器人关节轨迹跟踪控制可分为基于模型控制和无模型控制两种。计算力矩控制^[2]是一种基于模型控制的策略,其前馈路径能很好地抑制由于系统动态特性而引起的跟踪滞后,其反馈路径可以用来抵消不确定性引起的扰动。然而,由于结构和非结构的不确定性,使得精确的机器人动力学模型难以被确定,从而限制了计算力矩控制的跟踪性能。因此,机器人动力学模型通常被分为模型部分和不确定部分。后者由其他控制方法处理,如自适应控制^[3]、鲁棒控制^[4-5]、滑模控制^[6-8]、智能控制^[9-11]等。

比例-微分(PD)或比例-积分-微分(PID)^[12-13]是一种典型的无模型控制算法,由于其控制规律简单,被广泛应用于机器的控制器中。具有重力补偿的PD控制算法被用于定点跟踪任务中时,能够使机器人的关节位置误差全局渐近稳定^[14-15]。然而,重力项在实际应用中会因载荷未知或时变而变化,不能被完全补偿。因此,PD控制方案存在稳态误差。在PID控制中引入积分项是消除PD控制引起的静态误差的一种方法。在执行定点跟踪任务时,因不需要考虑动态因素的影响,PID可以达到很好的效果。但是在轨迹跟踪中,PID控制方案下的跟踪误差会受到机器人非线性动态行为的影响。文献[16]认为PID不适合处理高度非线性的系统,因为控制律的设计仅仅是基于局部参数。因此,模糊逻辑^[17-19]、自适应^[20]、神经网络^[21-22]等算法被用于在线调节PID的参数。虽然这些方法通过自动调整参数可以提高PID算法的跟踪精度,但额外的计算量阻碍了它们在实时处理控制中的应用。一些学者^[23-26]基于文献[27],采用饱和PID控制算法使机器人系统稳定。大部分机器人轨迹跟踪控制的方法在整个控制过程中参考轨迹基本与期望轨迹保持一致。实际上,可以通过修正参考轨迹来提高机器人的轨迹跟踪精度。

修正参考轨迹这种方法或思路,已经被一些学者研究并成功应用^[28-34],但是建立在传统迭代学习控制的基础上。该方法需要对期望轨迹重复跟踪,然后再修正参考轨迹,耗费大量时间成本。

本文提出一种基于参考轨迹实时修正的机器人轨迹控制方法。该方法将轨迹跟踪过程中已被跟踪点的误差进行累加,然后实时前馈补偿至参考轨迹

上即将被跟踪的点。通过实时修正参考轨迹来实现机器人各个关节的实际运动符合期望轨迹。只需控制器中的算法保证速度误差稳定,即可保证轨迹跟踪误差收敛。通过仿真和实验验证所提方法的有效性。

1 参考轨迹实时修正方案

1.1 控制框图

在机器人关节控制结构中加入了前馈补偿,如图1所示。在图中, q_d 表示期望的关节位置向量。它既可以表示为一个定点,也可以表示为一段连续轨迹。 q 表示机器人各关节当前的位置向量。 q_d 和 q 之间的误差 e 称为跟踪误差。 K_c 是一个对角矩阵,其维度与机器人自由度一致,称为补偿增益。符号 $\sum$ 表示累加器,它将轨迹上已被跟踪点的 $K_c e$ 累加起来。累加器的输出加到 q_d 上,得到新的关节命令或修正后的参考轨迹 q_c 。然后控制器根据指令误差 δq 对机器人关节运动进行调整, δq 是 q_c 与 q 之间的误差。当 $K_c=0$ 时, $q_c=q_d$, $\delta q=e$ 。

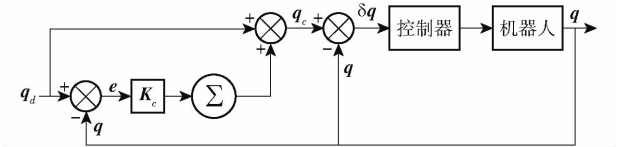


图1 基于参考轨迹实时修正的机器人控制结构
Fig. 1 Robot control structure based on reference trajectory real-time modification

1.2 跟踪误差收敛性分析

假设每个关节的轨迹被离散为无限个点,相邻点之间的采样时间间隔为 T 。根据图1可以得到第1个期望点与其当前关节位置之间的误差为

$$e_1 = q_d^{(1)} - q^{(1)} \tag{1}$$

以及第1次补偿后的新关节指令和指令误差

$$q_c^{(1)} = q_d^{(1)} + K_c^{(1)} e_1 \tag{2}$$

$$\delta q^{(1)} = q_c^{(1)} - q^{(1)} \tag{3}$$

将式(1)、(2)代入式(3)可以得到第1个点的跟踪误差与指令误差间的关系

$$e_1 = \delta q^{(1)} - K_c^{(1)} e_1 \tag{4}$$

第1个点补偿后,在控制器作用后,机器人各个关节到达 $q^{(2)}$ 位置。类似之前,可以得到

$$e_2 = q_d^{(2)} - q^{(2)} \tag{5}$$

$$q_c^{(2)} = q_d^{(2)} + K_c^{(1)} e_1 + K_c^{(2)} e_2 \tag{6}$$

$$\delta q^{(2)} = q_c^{(2)} - q^{(2)} \tag{7}$$

将式(5)、(6)代入式(7),得

$$e_2 = \delta q^{(2)} - K_c^{(1)} e_1 - K_c^{(2)} e_2 \tag{8}$$

以此类推,分别得到

$$\mathbf{e}_n = \delta \mathbf{q}^{(n)} - \mathbf{K}_c^{(1)} \mathbf{e}_1 - \mathbf{K}_c^{(2)} \mathbf{e}_2 \cdots - \mathbf{K}_c^{(n-1)} \mathbf{e}_{n-1} - \mathbf{K}_c^{(n)} \mathbf{e}_n \quad (9)$$

$$\mathbf{e}_{n+1} = \delta \mathbf{q}^{(n+1)} - \mathbf{K}_c^{(1)} \mathbf{e}_1 - \mathbf{K}_c^{(2)} \mathbf{e}_2 \cdots - \mathbf{K}_c^{(n)} \mathbf{e}_n - \mathbf{K}_c^{(n+1)} \mathbf{e}_{n+1} \quad (10)$$

接下来,令式(10)减去式(9),整理后可得

$$\mathbf{e}_{n+1} = (\mathbf{I} + \mathbf{K}_c^{(n+1)})^{-1} (\mathbf{e}_n + \mathbf{E}_{n+1}) \quad (n \in \mathbf{Z}^+) \quad (11)$$

其中 $\mathbf{E}_{n+1} = \delta \mathbf{q}^{(n+1)} - \delta \mathbf{q}^{(n)}$

式中 $\mathbf{I}$ ——单位矩阵

$\mathbf{E}_{n+1}$ ——指令误差的差分

为方便描述,利用无下标的 $\mathbf{E}$ 表示各次补偿中得到的指令误差的差分。

将式(11)迭代展开得到

$$\mathbf{e}_{n+1} = (\mathbf{I} + \mathbf{K}_c^{(n+1)})^{-1} \cdots (\mathbf{I} + \mathbf{K}_c^{(2)})^{-1} (\mathbf{e}_1 + \mathbf{E}_1) \cdots + (\mathbf{I} + \mathbf{K}_c^{(n+1)})^{-1} \cdots (\mathbf{I} + \mathbf{K}_c^{(j+1)})^{-1} \mathbf{E}_j \cdots + (\mathbf{I} + \mathbf{K}_c^{(n+1)})^{-1} \mathbf{E}_{n+1} \quad (12)$$

式中 $\mathbf{e}_1$ ——初始状态下关节实际位置与期望轨迹之间的误差

从式(12)可以看出,只要 $\mathbf{K}_c$ 是正定矩阵且指令误差的差分 $\mathbf{E}$ 等于零,跟踪误差 $\mathbf{e}$ 就会收敛,不受 $\mathbf{e}_1$ 的影响,并且 $\mathbf{K}_c$ 中的元素值越大,收敛速度越快。

为了令式(12)具有物理意义,式(12)两边同时除以采样时间 T ,可得

$$\frac{\mathbf{e}_{n+1}}{T} = (\mathbf{I} + \mathbf{K}_c^{(n+1)})^{-1} \cdots (\mathbf{I} + \mathbf{K}_c^{(2)})^{-1} \left(\frac{\mathbf{e}_1}{T} + \frac{\mathbf{E}_1}{T} \right) \cdots + (\mathbf{I} + \mathbf{K}_c^{(n+1)})^{-1} \cdots (\mathbf{I} + \mathbf{K}_c^{(j+1)})^{-1} \frac{\mathbf{E}_j}{T} \cdots + (\mathbf{I} + \mathbf{K}_c^{(n+1)})^{-1} \frac{\mathbf{E}_{n+1}}{T} \quad (13)$$

式(13)中 $\mathbf{E}/T$ 表示由指令误差差分计算得到的指令速度误差。由于指令速度误差受控制器中算法的影响,因此,式(12)、(13)的一个收敛条件为控制器算法能保证指令速度误差稳定。

综上所述,本文提出的方法能够使跟踪误差收敛的前提是控制器中算法能够保证速度误差稳定且 $\mathbf{K}_c$ 正定。其中, $\mathbf{K}_c$ 中元素的值越大,收敛速度越快。

1.3 PD 控制律证明

对于机器人,其动力学描述为

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}}) + \boldsymbol{\tau}_d = \mathbf{u} \quad (14)$$

式中 $\ddot{\mathbf{q}}$ ——关节加速度向量

$\mathbf{M}(\mathbf{q})$ ——惯性矩阵, $\in \mathbf{R}^m$

$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ ——科氏力和向心力矩阵, $\in \mathbf{R}^{m \times m}$

$\mathbf{g}(\mathbf{q})$ ——重力矩项, $\in \mathbf{R}^m$

$\mathbf{F}(\dot{\mathbf{q}})$ ——关节的黏度/静摩擦力矩, $\in \mathbf{R}^m$

$\boldsymbol{\tau}_d$ ——外部干扰, $\in \mathbf{R}^m$

$\mathbf{u}$ ——作用于各个关节的总控制输入, $\in \mathbf{R}^m$

为了便于描述,式(14)的左边用 $\mathbf{f}$ 替代。

图1中控制器可采用PD控制律,其表达式为

$$\mathbf{u} = \mathbf{K}_p \delta \mathbf{q} + \mathbf{K}_d \delta \dot{\mathbf{q}} \quad (15)$$

式中 $\mathbf{K}_p$ 、 $\mathbf{K}_d$ ——比例和微分增益,都为正定矩阵

$\delta \dot{\mathbf{q}}$ ——指令速度误差

PD控制律稳定性的证明可参照文献[36-37]。文献[36]从理论上证明了在增益 $\mathbf{K}_d$ 足够大的情况下,PD控制律可使机器人任意精度地跟踪给定期望轨迹。然而,实际中机器人每个驱动器的转矩有限,因此 $\mathbf{K}_d$ 不可能被设置得非常大。文献[37]证明了PD控制一致有界,这表明可以通过调整参数使轨迹跟踪达到期望的精度。

将式(15)代入式(14),并考虑Lyapunov函数 $V = \delta \mathbf{q}^T \mathbf{K}_p \delta \mathbf{q} / 2$,得

$$\dot{V} = \frac{1}{2} \delta \dot{\mathbf{q}}^T \mathbf{K}_p \delta \mathbf{q} + \frac{1}{2} \delta \dot{\mathbf{q}}^T \mathbf{K}_p \delta \mathbf{q} = \frac{1}{2} \delta \dot{\mathbf{q}}^T (\mathbf{K}_p \delta \mathbf{q} + \mathbf{f}) - \frac{1}{2} \delta \dot{\mathbf{q}}^T \mathbf{K}_d \delta \dot{\mathbf{q}} \quad (16)$$

在式(16)中,只需 $\mathbf{K}_p \delta \mathbf{q} = -\mathbf{f}$, $\dot{V}$ 即为半正定,系统也将可以稳定并使式(12)中的跟踪误差收敛。

机器人在实际运动中, $\mathbf{f}$ 会随运动状态的变化而变化,而本文 $\mathbf{K}_p$ 是固定值,因此要满足稳定条件, $\delta \mathbf{q}$ 须发生变化并且其值等于 $\mathbf{K}_p^{-1}(-\mathbf{f})$ 。本文所提的方法,实际上起着自动调节 $\delta \mathbf{q}$ 的作用。当 $\mathbf{K}_p \delta \mathbf{q} \neq -\mathbf{f}$ 时,机器人必将产生一个跟踪误差,将该误差补偿至参考轨迹上得到指令误差,会自动调节 $\mathbf{K}_p \delta \mathbf{q}$ 向 $-\mathbf{f}$ 变化。在实际运动中, $\delta \mathbf{q}$ 的自动调节难以精准地抵消掉 $\mathbf{f}$, 因此, $\mathbf{K}_p \delta \mathbf{q} + \mathbf{f}$ 会存在一个残差。由于存在残差,系统会在 $V=0$ 的邻域内振荡。

控制器的采样时间越小或者期望轨迹的速度越小,振荡邻域范围会越小。这种情况下,相邻采样点之间的误差变化十分接近,相当于提高了 $\delta \mathbf{q}$ 调节的分辨率。因此,在控制过程中, $\mathbf{K}_p \delta \mathbf{q} + \mathbf{f}$ 的残差也会随之减小,系统也会更接近稳定点。因为摩擦力等不是连续变化的,所以实际跟踪中误差可能会出现突变。

2 参数调节方案

在实际控制中,PD控制律通常采用离散形式,并且它的速度误差由位置误差差分计算得到。因此,式(15)可改写为

$$\mathbf{u}_j = \mathbf{K}_p \delta \mathbf{q}_j + \mathbf{K}_d \frac{\mathbf{E}_j}{T} \quad (17)$$

由1.2节和1.3节可知,增大 $\mathbf{K}_p$ 和 $\mathbf{K}_c$ 都能有效地减小最终的跟踪误差。从已有的PD控制律的参

数调节经验可知,过大的 K_d 会放大噪声,从而引起系统振荡并导致机器人的关节抖动。因此,所提算法主要调节 K_p 和 K_e 。这两个参数的调节方案如下:

方案 1:如果系统受摩擦力等不连续力的影响大,则主要增加 K_p ,可有效地降低在它们不连续区域的跟踪误差。

方案 2:如果系统受摩擦力等不连续力的影响小,则主要增加 K_e ,可有效地缩小二者的调节范围。

在调节 K_p 和 K_e 的过程中,如果跟踪误差出现了发散性的振荡,则适当降低 K_d ,再增加 K_p 或 K_e 。其中 K_p 和 K_d 的初始值可根据已有经验设置, K_e 的初始值可以设置为一个接近于零的值。

3 仿真和实验验证

3.1 期望轨迹

轨迹 1 用于参数调节和测试,轨迹 2 用于验证所提方法对不同轨迹的适应性以及与 PID 的对比,可得

$$\boldsymbol{q}_{di}(t) = \begin{cases} \boldsymbol{q}_{di}^0 + \boldsymbol{A}_i \cos(\boldsymbol{\omega}_i t) & \text{(轨迹 1)} \\ \boldsymbol{q}_{di}^0 + \boldsymbol{A}_i \cos(2\pi \boldsymbol{f}_{i1}(1 + \cos(2\pi \boldsymbol{f}_{i2}t))t) & \text{(轨迹 2)} \end{cases}$$

轨迹 2 能激发机器人在慢速和快速状态下的动力学特性^[35]。

3.2 仿真

以一款只含旋转关节的 2 自由度平面机器人为例。惯性矩阵表达式为

$$\boldsymbol{M} = [m_{ij}]_{2 \times 2}$$

其中 $m_{11} = m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos q_2) + I_{c1} + I_{c2}$

$$m_{12} = m_{21} = m_2 (l_{c2}^2 + l_1 l_{c2} \cos q_2) + I_{c2}$$
$$m_{22} = m_2 l_{c2}^2 + I_{c2}$$

科氏力和向心力矩矩阵表达式为

$$\boldsymbol{C} = [c_{ij}]_{2 \times 2}$$

其中 $c_{11} = \dot{h}q_2 \quad c_{12} = \dot{h}q_1 + \dot{h}q_2$

$$c_{21} = -\dot{h}q_1 \quad c_{22} = 0 \quad h = -m_2 l_1 l_{c2} \sin q_2$$

重力矩表达式为

$$\boldsymbol{G} = [G_1 \quad G_2]^T$$

其中 $G_1 = (m_1 l_{c1} + m_2 l_1) g \cos q_1 + m_2 l_{c2} g \cos(q_1 + q_2)$

$$G_2 = m_2 l_{c2} g \cos(q_1 + q_2)$$

关节黏度/静摩擦力矩表达式为

$$\boldsymbol{F}_r = [F_{r1} \quad F_{r2}]^T$$

其中 $F_{r1} = 3 \operatorname{sign}(\dot{q}_1) + 5 \dot{q}_1$

$$F_{r2} = 3 \operatorname{sign}(\dot{q}_2) + 5 \dot{q}_2$$

函数 $\operatorname{sign}()$ 表示只取输入的正负符号。

干扰向量表达式为

$$\boldsymbol{\tau}_d = [\boldsymbol{\tau}_{d1} \quad \boldsymbol{\tau}_{d2}]^T$$

其中 $\boldsymbol{\tau}_{d1} = 0.3 \sin t \quad \boldsymbol{\tau}_{d2} = 0.1(1 - e^{-t})$

机器人参数分别为 $m_1 = 10 \text{ kg}, m_2 = 5 \text{ kg}, l_1 = 1 \text{ m}, l_2 = 0.5 \text{ m}, l_{c1} = 0.5 \text{ m}, l_{c2} = 0.25 \text{ m}, I_{c1} = 0.83 \text{ kg} \cdot \text{m}^2, I_{c2} = 0.3 \text{ kg} \cdot \text{m}^2, g = 9.81 \text{ m/s}^2$ 。所有算法采样时间 T 设为 0.005 s 。

3.2.1 轨迹 1 跟踪结果

轨迹 1 参数分别设为 $\boldsymbol{A} = (1, 1) \text{ rad}, \boldsymbol{q}^0 = (-1, -1) \text{ rad}, \boldsymbol{\omega} = (0.4\pi, 0.4\pi)$ 。轨迹执行总时间设为 10 s 。

所提算法的 3 个参数按照第 2 节中的方案人工试错调节得到,分别为 $\boldsymbol{K}_p = \operatorname{diag}\{80, 10\,690\}, \boldsymbol{K}_d = \operatorname{diag}\{30, 30\}, \boldsymbol{K}_e = \operatorname{diag}\{21, 1\}$ 。第 1 个关节参数根据方案 1 调节得到,第 2 个关节根据方案 2 调节得到。

图 2 为期望轨迹 $\boldsymbol{q}_d$ 、关节实际轨迹 $\boldsymbol{q}$ 和实时修正后的参考轨迹 $\boldsymbol{q}_e$ 。由图 2 可以看出,修正后参考轨迹与其他两条有很大的区别,而另外 2 条重合。表明本文所提出的方法在机器人轨迹跟踪控制问题上可行。图 3 为关节实际轨迹与期望轨迹之间的误差,两个关节误差分别在 $[-0.000\,3, 0.003\,4] \text{ rad}$ 和 $[-0.000\,2, 0.000\,45] \text{ rad}$ 之间,表明所提方法有效,跟踪精度优。但是跟踪误差存在周期性的尖峰,且在初始阶段尖峰值最大。周期性的尖峰是速度过零时引起摩擦力突变造成。初始阶段高尖峰值则还

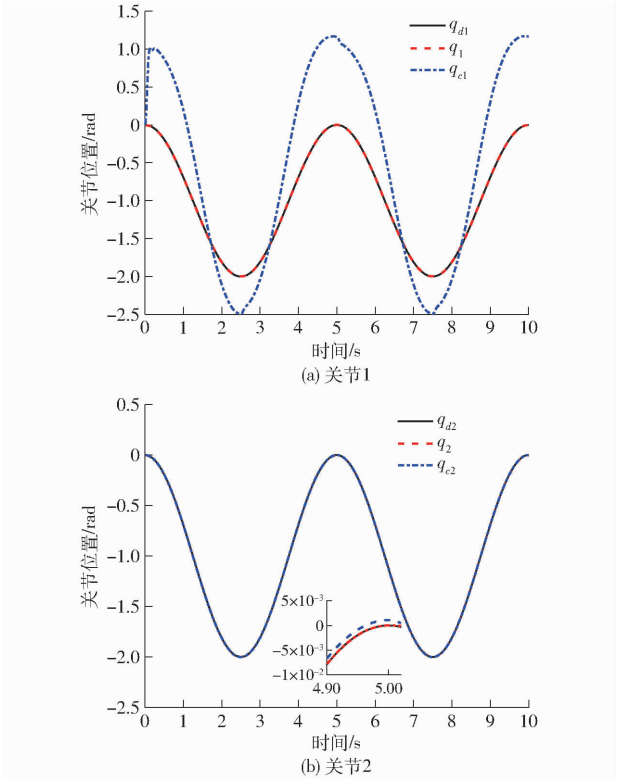


图 2 关节轨迹跟踪曲线

Fig. 2 Joint trajectory tracking curves

有重力矩作用的原因。因为初始阶段,所提方法作用效果是一个由弱到强的过程,因此它需要一段时间来抵消掉重力矩的影响。图 4 为指令误差差分,可以看出它与跟踪误差除了幅度以外,整体的变化趋势十分接近。

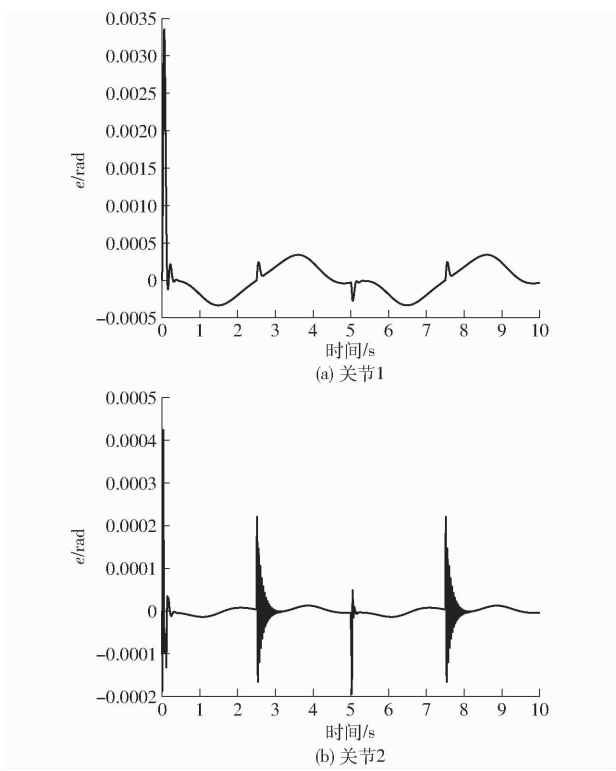


图 3 关节轨迹的跟踪误差

Fig. 3 Tracking errors of joint trajectory

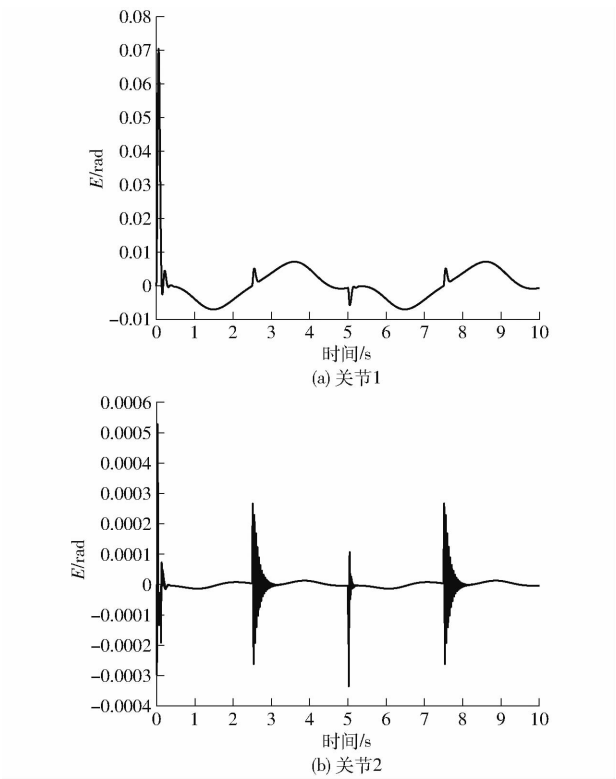


图 4 关节指令误差差分

Fig. 4 Numerical difference of command error for joints

3.2.2 轨迹 2 跟踪结果

轨迹 2 参数分别设为 $A = (1, 1)$ rad, $q_d^0 = (-1, -1)$ rad, $f_1 = (0.05, 0.075)$, $f_2 = (0.015, 0.015)$ 。整条轨迹的执行时间设为 30 s。

由于 P(I)D 主要通过增加比例项 K_p 和积分项 K_i 来提高跟踪精度,因此,上一节设置的 K_d 保持不变,也可更直观地比较本文方法与 P(I)D 的性能。通过人工调节, P(I)D 的另外两个参数被确定为 $K_p = \text{diag} \{ 62\ 000, 41\ 800 \}$, $K_i = \text{diag} \{ 62\ 000, 60\ 000 \}$ 。

图 5 为本文方法与 P(I)D 跟踪轨迹 2 得到的跟踪误差。在图 5 中,本文方法在前 0.2 s 出现了较大的尖峰,随后会变周期性地出现较小的尖峰,但整体比较稳定。单纯的 PD 控制律在前 1.2 s 存在剧烈的振荡,而 PID 的振荡持续了约 2 s。为了更好地对比性能,利用 2 s 后得到的跟踪误差绝对平均值来衡量跟踪精度,结果见表 1。由表 1 可知,本文方法对两个关节的跟踪精度比 PD 分别提高 93.14% 和 98.38%;比 PID 分别提高 79.39% 和 95.35%。

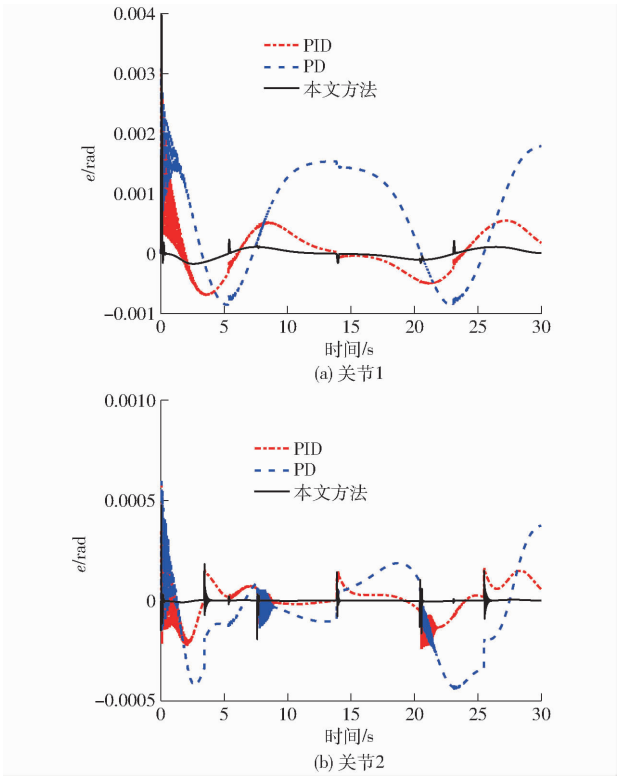


图 5 本文方法和 P(I)D 跟踪轨迹 2 的误差对比

Fig. 5 Errors obtained by proposed method and P(I)D tracking trajectory 2

图 6 为 3 个算法计算得到的控制输入力矩。本文方法在关节 1 前 0.2 s 内存在一个突变,需要提供比 P(I)D 更大的控制输入。随后三者的控制输入相差不大。P(I)D 的 K_p 较大,可以很好地抑制重力

表 1 3 个算法跟踪误差绝对平均值

Tab.1 Absolute average value of tracking error
obtained by three algorithmsrad

算法	关节	绝对平均值
PID	1	0.000 303
	2	0.000 059
PD	1	0.000 909
	2	0.000 169
本文方法	1	0.000 062
	2	0.000 002

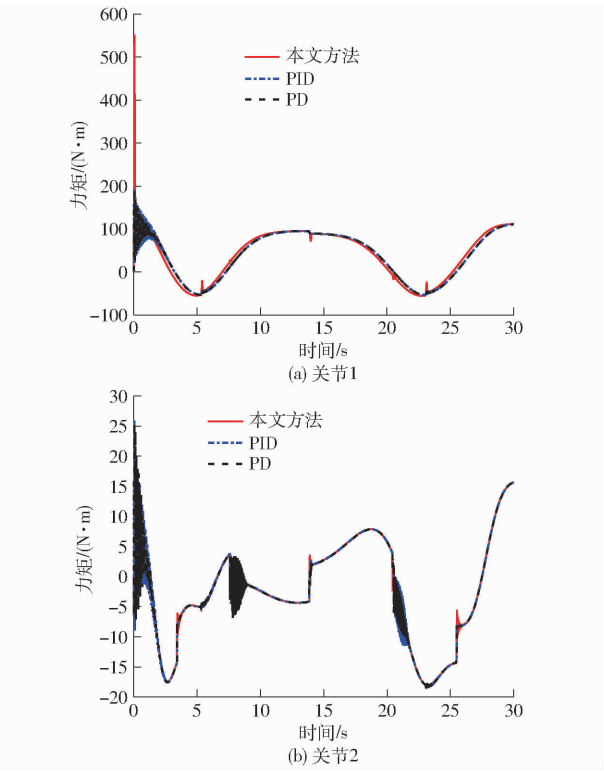


图 6 本文方法和 P(I)D 跟踪轨迹 2 的控制输入
Fig.6 Control inputs obtained by proposed method and P(I)D

矩和摩擦力的作用,但是也会使它们在控制初始阶段出现振荡。

综上所述,本文方法的收敛速度、稳定性和精度均较 P(I)D 更好,但是初始阶段在克服重力矩影响时,需要提供更大的控制力矩。

3.2.3 重力矩补偿测试

根据提供的仿真模型,可将 2 个关节都在 0 rad 时计算得到的重力矩看作真实值,它们分别为 110.36、12.26 N·m。实际操作中,很难准确估算出这两个值。因此,取它们的 80% 作为估算值补偿在式(17)右侧。图 7、8 分别为重力矩补偿前后所提方法的跟踪误差和控制输入。由图 7、8 可看出,所提的这个补偿方案可行且效果很好。初始阶段的控制输入显著下降,跟踪精度也得到了提高。

3.3 实验

机器人初始构型如图 9 所示,其各个关节由分辨

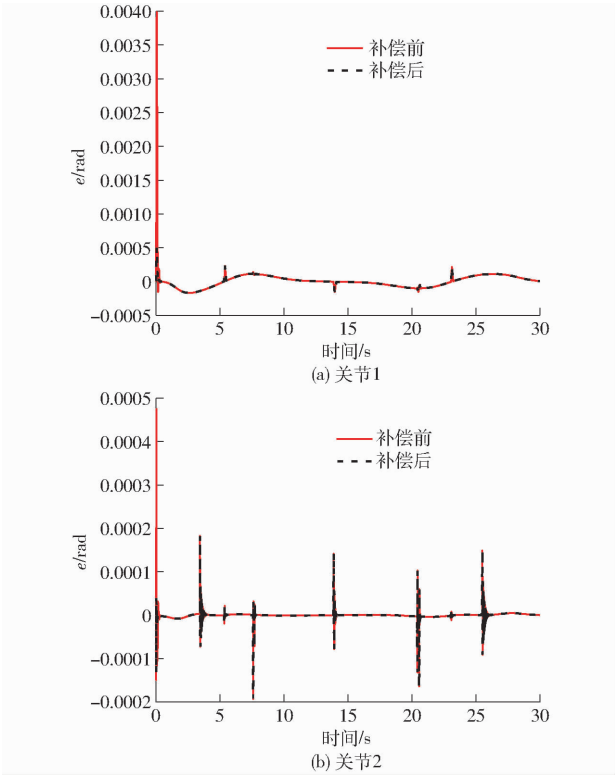


图 7 本文方法重力矩补偿前后的跟踪误差
Fig.7 Tracking error of proposed method before and after gravity compensation

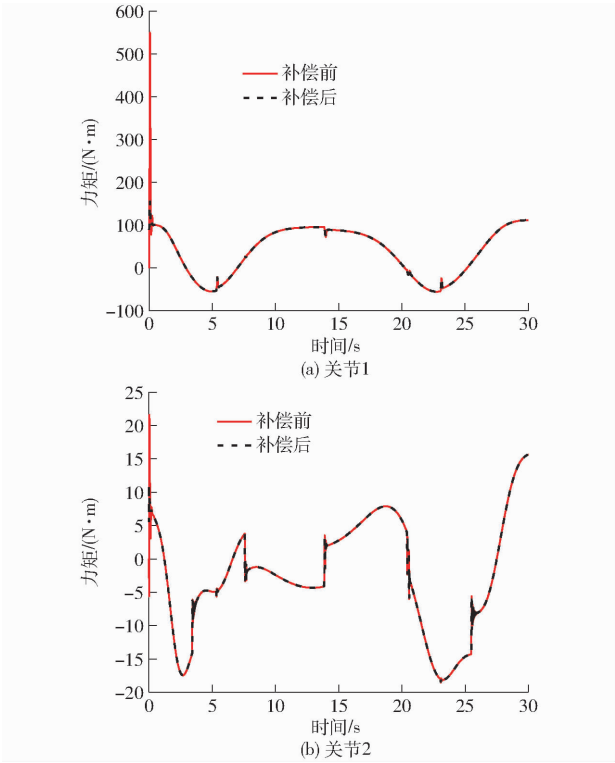


图 8 本文方法重力矩补偿前后的控制输入
Fig.8 Control inputs of proposed method before and after gravity compensation

率为 0.001 5 rad/p 的 Dynamixel XM430 - W350 型舵机驱动。该款机器人总质量仅 0.7 kg,各关节受重力矩影响小。此外,其各关节受摩擦力的影响较小。因

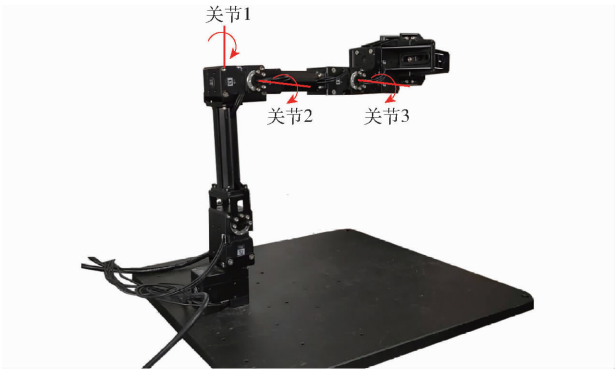


图 9 实物机器人
Fig. 9 Physical robot

此,该机器人实验能更好地反映所提方法的有效性。本文控制算法均在 Arduino 中编写,然后被上传至

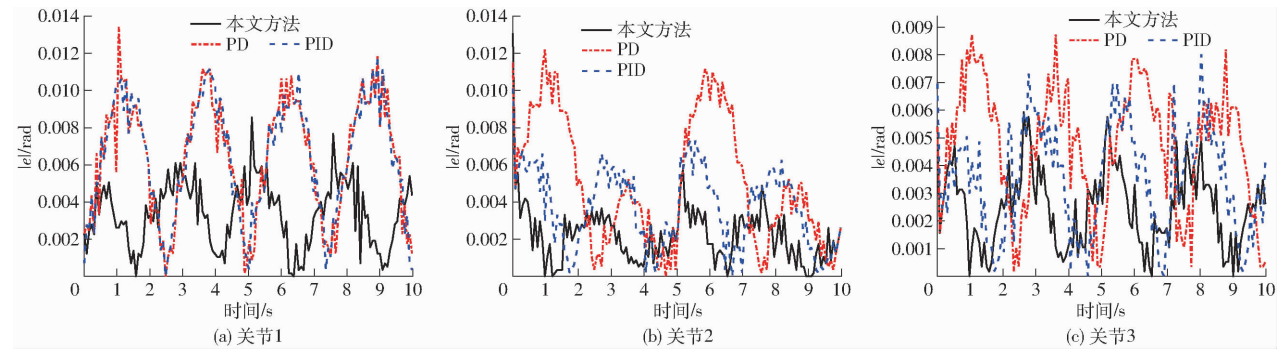


图 10 3 个算法在实物机器人上跟踪轨迹 1 的跟踪误差绝对值
Fig. 10 Absolute value of tracking errors obtained by three algorithms tracking trajectory 1 on physical robot

由于各个关节的实际初始位置不处于真正的机械零位,因此对它们在 0.08 s 后的绝对平均值和最大绝对值进行比较,结果见表 2。

轨迹 2 用作测试和对比 3 个算法对不同轨迹的适应能力。其参数为 $A = (\pi/6, \pi/6, -\pi/6) \text{ rad}$, $q_d^0 = (-\pi/6, -\pi/6, \pi/6) \text{ rad}$, $f_1 = (0.05, 0.05, 0.05)$, $f_2 = (0.015, 0.015, 0.015)$ 。轨迹执行时间为 30 s。图 11 为 3 个算法跟踪轨迹 2 得到的误差绝对值,表 3 为其在 0.08 s 后得到的绝对平均值和最大绝对值。

由图 10、11 可知,所提方法只需很小的参数便可达到比 P(I)D 更小的跟踪误差。轨迹 2 比轨迹 1

OpenCR1.0 板以控制机器人的关节运动。所有算法在该控制板上的采样时间 T 设为 0.005 ms。

轨迹 1 用于调节本文方法和 P(I)D 3 个算法的参数,并比较算法的性能。该轨迹的各项参数为 $A = (\pi/4, \pi/6, \pi/6) \text{ rad}$, $q_d^0 = (-\pi/4, -\pi/6, -\pi/6) \text{ rad}$, $\omega = (0.4\pi, 0.4\pi, 0.4\pi)$ 。轨迹执行总时间设为 10 s。为了更直观地对比 3 个算法,它们的 $K_d = \text{diag}\{20, 20, 20\}$ 。所提方法的另外两个参数采用第 2 节中的方案 2 调节,经人工试错确定为 $K_p = \text{diag}\{60, 60, 60\}$, $K_c = \text{diag}\{3.2, 4, 3\}$ 。P(I)D 的另外两个参数确定为 $K_p = \text{diag}\{18\,000, 18\,000, 18\,000\}$, $K_I = \text{diag}\{18\,000, 18\,000, 18\,000\}$ 。3 个算法用跟踪误差的绝对值衡量,如图 10 所示。

表 2 3 个算法跟踪轨迹 1 跟踪误差的绝对平均值和最大绝对值			
Tab.2 Absolute average value and maximum absolute value of tracking errors obtained by three algorithms tracking trajectory 1			
tracking trajectory 1 rad			
算法	关节	绝对平均值	最大绝对值
PID	1	0.006 5	0.013 1
	2	0.003 7	0.008 6
	3	0.003 4	0.008 9
PD	1	0.006 5	0.013 6
	2	0.005 0	0.012 6
	3	0.004 6	0.009 3
本文方法	1	0.003 4	0.008 7
	2	0.002 1	0.007 3
	3	0.002 6	0.006 8

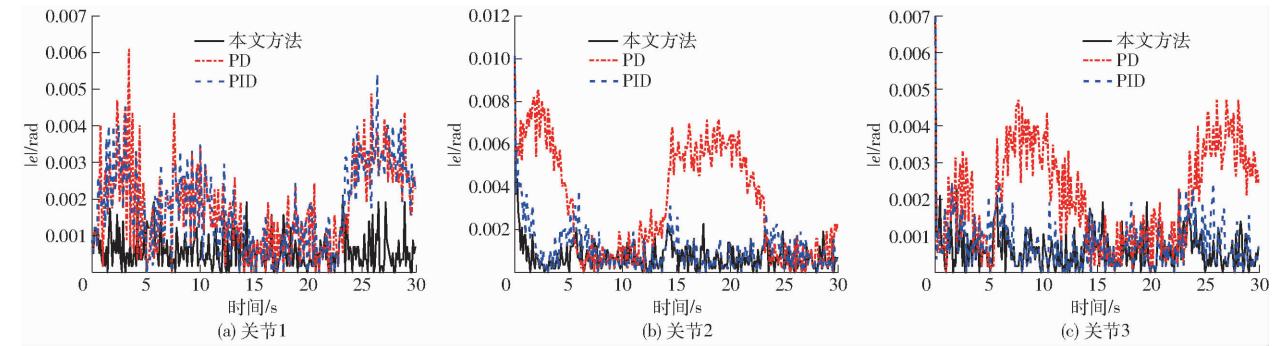


图 11 3 个算法在实物机器人上跟踪轨迹 2 的跟踪误差绝对值
Fig. 11 Absolute values of tracking errors obtained by three algorithms tracking trajectory 2 on physical robot

表 3 3 个算法跟踪轨迹 2 得到的跟踪误差的绝对平均值
和最大绝对值

Tab.3 Absolute average value and maximum absolute
value of tracking errors obtained by three algorithms
tracking trajectory 2 rad

算法	关节	绝对平均值	最大绝对值
PID	1	0.001 8	0.006 3
	2	0.001 1	0.006 8
	3	0.000 8	0.003 8
PD	1	0.001 8	0.007 5
	2	0.003 3	0.009 3
	3	0.002 0	0.005 6
本文方法	1	0.000 6	0.003 5
	2	0.000 7	0.005 9
	3	0.000 6	0.003 3

的速度慢,结果表明所提方法跟踪速度越慢的期望
轨迹,跟踪误差越小。

4 结 论

- (1)将轨迹跟踪过程中已被跟踪点的误差累加
起来作为前馈补偿至即将被跟踪的点上的方法,对
机器人的轨迹跟踪控制问题可行。
- (2)在控制器中的算法使速度误差稳定的前提
下,所提方法可以令轨迹跟踪误差收敛。
- (3)所提方法可以起到自动调节 PD 控制律中
参数的作用。与 PID 一样有 3 个参数需要整定,但
是收敛速度更快,跟踪误差更小。
- (4)由于控制器采用 PD 算法,所提方法的跟踪
效果也会受其影响:机器人期望轨迹的速度越小,跟
踪误差会越小;反之,跟踪误差也会增大。
- (5)将初始位置的重力矩作为常数补偿,可以
有效地减小所提方法初始阶段的误差和控制输入。

参 考 文 献

[1] XIAO F, LI G, JIANG D, et al. An effective and unified method to derive the inverse kinematics formulas of general six-DOF manipulator with simple geometry[J]. Mechanism and Machine Theory, 2021, 159: 104265.

[2] LLAMA M A, KELLY R, SANTIBANWZ V. Stable computed-torque control of robot manipulators via fuzzy self-tuning[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2000, 30(1): 143 – 150.

[3] WANG H. Adaptive control of robot manipulators with uncertain kinematics and dynamics[J]. IEEE Transactions on Automatic Control, 2016, 62(2): 948 – 954.

[4] 郭庆贺,赵丁选,赵小龙,等. 基于内模控制的主动悬挂电液伺服作动器位置控制研究[J]. 农业机械学报,2020,51(12): 394 – 404.

GUO Qinghe, ZHAO Dingxuan, ZHAO Xiaolong, et al. Internal model control in position control of active suspension electro-hydraulic servo actuator[J]. Transactions of the Chinese Society for Agricultural Machinery, 2020, 51(12): 394 – 404. (in Chinese)

[5] LIU H, TIAN X, WANG G, et al. Robust H_{∞} finite-time stability control of a class of nonlinear systems[J]. Applied Mathematical Modelling, 2016, 40(7 – 8): 5111 – 5122.

[6] 陈庆诚,朱世强,蒋瑜,等. 液压外骨骼机器人非线性模糊滑模控制方法[J]. 农业机械学报,2017,48(4):355 – 361,405.

CHEN Qingcheng, ZHU Shiqiang, JIANG Yu, et al. Nonlinear fuzzy SMC algorithm for hydraulic-driven exoskeleton robot system[J]. Transactions of the Chinese Society for Agricultural Machinery, 2017, 48(4): 355 – 361, 405. (in Chinese)

[7] FALLAHA C J, SAAD M, KANAAN H Y, et al. Sliding-mode robot control with exponential reaching law[J]. IEEE Transactions on Industrial Electronics, 2010, 58(2): 600 – 610.

[8] BAEK J, JIN M, HAN S. A new adaptive sliding-mode control scheme for application to robot manipulators[J]. IEEE Transactions on Industrial Electronics, 2016, 63(6): 3628 – 3637.

[9] HE W, OUYANG Y, HONG J. Vibration control of a flexible robotic manipulator in the presence of input deadzone[J]. IEEE Transactions on Industrial Informatics, 2016, 13(1): 48 – 59.

[10] WAI R J, MUTHUSAMY R. Design of fuzzy-neural-network-inherited backstepping control for robot manipulator including actuator dynamics[J]. IEEE Transactions on Fuzzy Systems, 2013, 22(4): 709 – 722.

[11] LIU C, ZHAO Z, WEN G. Adaptive neural network control with optimal number of hidden nodes for trajectory tracking of robot manipulators[J]. Neurocomputing, 2019, 350: 136 – 145.

[12] ALVAREZ-RAMIREZ J, KELLY R, CERVANTES I. Semiglobal stability of saturated linear PID control for robot manipulators[J]. Automatica, 2003, 39(6): 989 – 995.

[13] ARMSTRONG B S R, GUTIERREZ J A, WADE B A, et al. Stability of phase-based gain modulation with designer-chosen switch functions[J]. The International Journal of Robotics Research, 2006, 25(8): 781 – 796.

[14] TOMEI P. Adaptive PD controller for robot robots[J]. IEEE Transactions on Robotics and Automation, 1991, 7(4): 565 – 570.

[15] MONTALVO W, ESCOBAR-NARANJO J, GARCIA C A, et al. Low-cost automation for gravity compensation of robotic arm[J]. Applied Sciences, 2020, 10(11): 3823.

[16] ROCCO P. Stability of PID control for industrial robot arms[J]. IEEE Transactions on Robotics and Automation, 1996, 12(4): 606 – 614.

- [17] KHOOBAN M H, NIKNAM T, SHA-SADEGHI M. Speed control of electrical vehicles: a time-varying proportional - integral controller-based type-2 fuzzy logic[J]. IET Science, Measurement & Technology, 2016, 10(3): 185 - 192.
- [18] VASU S. Fuzzy PID based adaptive control on industrial robot system[J]. Materials Today: Proceedings, 2018, 5(5): 13055 - 13060.
- [19] YUN J, SUN Y, LI C, et al. Self-adjusting force/bit blending control based on quantitative factor-scale factor fuzzy-PID bit control[J]. Alexandria Engineering Journal, 2022, 61(6): 4389 - 4397.
- [20] TAVANA M R, KHOONAN M H, NIKNAM T. Adaptive PI controller to voltage regulation in power systems: STATCOM as a case study[J]. ISA Transactions, 2017, 66: 325 - 334.
- [21] FANG M C, ZHUO Y Z, LEE Z Y. The application of the self-tuning neural network PID controller on the ship roll reduction in random waves[J]. Ocean Engineering, 2010, 37(7): 529 - 538.
- [22] JAISUMROUM N, CHOTIPRAYANAKUL P, LIMNARARAT S. Self-tuning control with neural network for robot manipulator [C]//2016 16th International Conference on Control, Automation and Systems (ICCAS). IEEE, 2016: 1073 - 1076.
- [23] SUN D, HU S, SHAO X, et al. Global stability of a saturated nonlinear PID controller for robot manipulators[J]. IEEE Transactions on Control Systems Technology, 2009, 17(4): 892 - 899.
- [24] KELLY R. Global positioning of robot manipulators via PD control plus a class of nonlinear integral actions[J]. IEEE Transactions on Automatic Control, 1998, 43(7): 934 - 938.
- [25] GOREZ R. Globally stable PID-like control of mechanical systems[J]. Systems & Control Letters, 1999, 38(1): 61 - 72.
- [26] SU Y, ZHENG C. PID control for global finite-time regulation of robotic manipulators[J]. International Journal of Systems Science, 2017, 48(3): 547 - 558.
- [27] ARIMOTO S. A class of quasi-natural potentials and hyper-stable PID servo-loops for nonlinear robotic systems [J]. Transactions of the Society of Instrument and Control Engineers, 1994, 30(9): 1005 - 1012.
- [28] 沈国良, 赵均, 钱积新. 基于期望响应的预测函数控制参考轨迹自校正方法[J]. 化工学报, 2008, 59(1): 118 - 125.
SHEN Guoliang, ZHAO Jun, QIAN Jixin. Self-tuning scheme of predictive function control's reference trajectory based on expected response[J]. CIESC Journal, 2008, 59(1): 118 - 125. (in Chinese)
- [29] 闫涛, 章宏, 孙丽君. 基于参考轨迹修正的机床进给误差动态补偿研究[J]. 机电工程, 2021, 38(3): 368 - 372, 404.
YAN Tao, ZHANG Hong, SUN Lijun. Dynamic compensation of machine tool feed error based on reference trajectory correction[J]. Journal of Mechanical & Electrical Engineering, 2021, 38(3): 368 - 372, 404. (in Chinese)
- [30] 陶洪峰, 董晓齐, 杨慧中. 参考轨迹更新的点到点迭代学习控制算法优化及应用[J]. 控制理论与应用, 2016, 33(9): 1207 - 1213.
TAO Hongfeng, DONG Xiaoqi, YANG Huizhong. Optimal algorithm and application for point to point iterative learning control via updating reference trajectory[J]. Control Theory & Applications, 2016, 33(9): 1207 - 1213. (in Chinese)
- [31] WANG Z, ZHOU R, HU C, et al. Online iterative learning compensation method based on model prediction for trajectory tracking control systems[J]. IEEE Transactions on Industrial Informatics, 2021, 18(1): 415 - 425.
- [32] 朱雪枫, 王建辉. 非线性重复运动系统的双迭代优化学习控制[J]. 控制理论与应用, 2021, 38(8): 1265 - 1274.
ZHU Xuefeng, WANG Jianhui. Double iterative optimal learning control of nonlinear repetitive motion system[J]. Control Theory & Applications, 2021, 38(8): 1265 - 1274. (in Chinese)
- [33] STANCESKI S, ZHANG J. A simple and effective learning approach to motion error correction of an industrial robot[C]//2019 International Conference on Advanced Mechatronic Systems (ICAMEchS). IEEE, 2019: 120 - 125.
- [34] TAN K K, ZHAO S, XU J X. Online automatic tuning of a proportional integral derivative controller based on an iterative learning control approach[J]. IET Control Theory & Applications, 2007, 1(1): 90 - 96.
- [35] ROVEDA L, FORGIONE M, PIGA D. Robot control parameters auto-tuning in trajectory tracking applications[J]. Control Engineering Practice, 2020, 101: 104488.
- [36] KAWAMURA S, MIYAZAKI F, ARIMOTO S. Is a local linear PD feedback control law effective for trajectory tracking of robot motion? [C]//Proceedings of IEEE International Conference on Robotics and Automation. IEEE, 1988: 1335 - 1340.
- [37] SALAS F G, SANTIBANEZ V, LLAMA M A. Variable gains PD tracking control of robot manipulators: stability analysis and simulations[C]//World Automation Congress 2012. IEEE, 2012: 1 - 6.