



基于神经网络的机器人自学习控制器¹⁾

王耀南

(湖南大学电气工程系 长沙 410082)

摘要 提出一种神经网络与 PID 控制相结合的机器人自学习控制器. 为加快神经网络的学习收敛性, 研究了有效的优化学习算法. 以两关节机器人为对象的仿真表明, 该控制器使机器人跟踪希望轨迹, 其系统响应、跟踪精度和鲁棒性优于常规的控制策略.

关键词 神经网络, 学习算法, 机器人控制.

1 引言

机器人控制对象是一个多关节机械手, 它的数学模型相当复杂, 与运动学和动力学密切相关, 是一个强耦合、非线性的多变量系统. 传统的机器人控制方法在理论上虽然能做到对机器人的快速、精确控制, 但实际上很难实现. 近年来神经网络控制为解决机器人控制中存在的一些问题提供了新的途径. 许多学者已用 ANN 解决机器人的复杂控制^[1-3]. 本文提出将带示教机构的逆动态控制和自校正控制两者有机地结合起来构成一种新的 ANN 自学习控制策略, 系统的控制结构如图1所示.

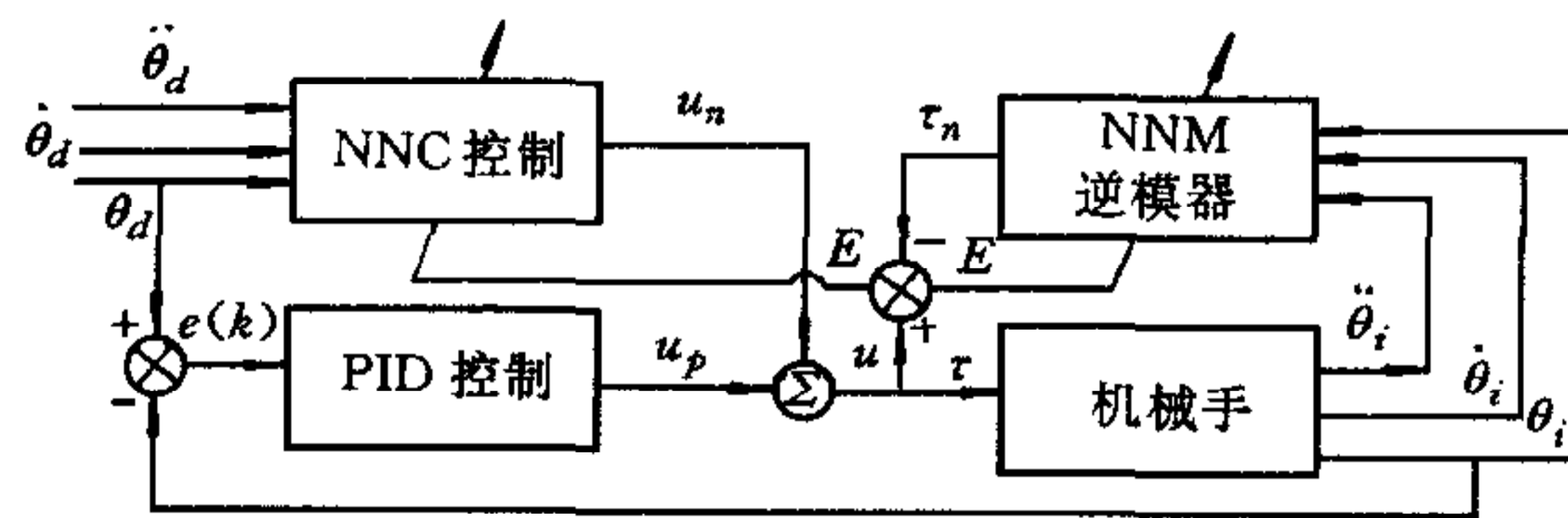


图1 神经网络机器人学习控制

图1中 NNC 神经网络用作前馈控制器, NNM 逆模神经网络用来在线学习机械手的逆动态模型. NNC 网络根据逆模器(NNM)的参数对机械手进行控制, 而 PID 控制器作为示教辅助控制器, 其主要作用是当控制开始或者系统参数发生突然变化时, 神经网络尚未学到系统的逆动态特性时对系统进行控制. 当 NNC 和 NNM 学到系统的逆动态模型之后, PID 控制的作用下降, 以至可以忽略. 这种新的控制策略比以往的神经网络机器人控制方式更为有效. 由于是直接学习系统的逆动态而不是在 PID 反馈控制器的作用指导

1) 中国博士后科学基金资助课题.

下进行学习,因而学习精度比文献[3]的带示教机构的逆动态控制方法更高.同样由于引进了常规 PID 反馈控制器作为补偿控制,无需和文献[4]的直接自校正控制器那样须先对系统的逆动态进行离线学习,而减少了学习样本选择不当对控制精度的影响,并且当系统发生较大变化时,虽然 NNC 控制器不能及时跟踪这种变化,但由于 PID 控制器的补偿,使得控制误差不会偏离过大,保证了系统的稳定性.

2 基于 ANN 的自学习控制

一个 n 个自由度的机械手封闭形式的动力学方程可以表示为

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta), \quad (1)$$

其中 $M(\theta)$ 为 $n \times n$ 维对称正定惯性矩阵, $V(\theta, \dot{\theta})$ 为 $n \times 1$ 维哥氏力和向心力矩矢量; $G(\theta)$ 为 $n \times 1$ 维重力矢量, $\theta, \dot{\theta}, \ddot{\theta}$ 分别为 $n \times 1$ 维的机械手关节位置,速度和加速度.为了简化,这里认为每一个关节只由一个驱动器单独驱动, τ 是 $n \times 1$ 维的关节控制力矩矢量.

设逆模网络 NNM 的输出为

$$\tau_n = N(\ddot{\theta}_i, \dot{\theta}_i, \theta_i, W), \quad (2)$$

控制网络 NNC 的输出 $u_n = F(\ddot{\theta}_d, \dot{\theta}_d, \theta_d, W)$, (3)

PID 控制律为

$$u_p = u_p(k-1) + k_p[e(k) - e(k-1)] + k_i e(k) + k_d[e(k) - 2e(k-1) + e(k-2)], \quad (4)$$

自学习控制律 $u = u_n + u_p$. (5)

对于 NNC 和 NNM 网络均采用三层 BP 网络,其输入输出关系为

$$o_j^{(1)}(k) = f\left(\sum_{i=1}^n W_{ij}^{(1)}(k) \cdot x_i(k)\right), \quad (6)$$

$$o_j^{(2)}(k) = \sum_{j=1}^m W_j^{(2)}(k) \cdot o_j^{(1)}(k), \quad (7)$$

$$f(x) = (1 - e^{-x}) / (1 + e^{-x}), \quad (8)$$

其中, $o_j^{(1)}$ 表示隐节点的输出, $o_j^{(2)}$ 表示输出层的输出, $W_{ij}^{(1)}(k)$, $W_j^{(2)}(k)$ 分别为连接输入层与隐层,隐层与输出层的权值, $x_i(k)$ 表示输入节点.对 NNM 网络输入为 $x = [\ddot{\theta}_i, \dot{\theta}_i, \theta_i]$, 输出为 $\tau_n = o_j^{(2)}$;对 NNC 网络输入节点为 $x = [\ddot{\theta}_d, \dot{\theta}_d, \theta_d]$, 输出节点为 $u_n = o_j^{(2)}$.

3 快速的学习算法

为加快机器人自学习控制,解决机器人实时控制的问题,本文提出一种新的变学习率优化学习算法 MLA(minimization learning algorithm).算法的基本思想是:在极小点附近用二阶 Taylor 多项式近似目标函数 $J(W)$,以求出极小点的估计值.

定义在线学习的性能指标为

$$J(w) = \frac{1}{2} \sum_{i=1}^m E_i^2(w) = \frac{1}{2} \sum_{i=1}^m (\tau - \tau_n)^2, \quad (9)$$

式中 w 表示网络权值向量 $w \in R^n$, E_i 表示期望输出 τ 与逆模网络输出 τ_n 之间的误差.

由函数极值理论知,函数 $J_k(W)$ 在极小点附近的二次近似性能指标为

$$g_k(w) = \alpha_k + (w - w_k)^T H_k^{-1} (w - w_k), \quad (10)$$

式中 α_k 为二次近似函数的极小值, w_k 为 $g_k(W)$ 的极小点, H_k^{-1} 为正定的 Hessian 矩阵, $H_k^{-1} \in R^{n \times n}$.

函数 $E_i(w)$ 在 w_k 点附近取一阶 Taylor 多项式

$$E_i(w) = E_i(w_k) + (w - w_k)^T \nabla E_i(w_k) + H \cdot O \cdot T, \quad (11)$$

式中, $\nabla E_i(w_k)$ 表示 $E_i(w_k)$ 对 w_k 的梯度.

将(11)式略去高阶项 $H \cdot O \cdot T$ 后,代入(9)式得到性能函数的一阶近似

$$J_k(w) \approx \sum_{i=1}^m [E_i(w_k) + (w - w_k)^T \cdot \nabla E_i(w_k)]^2. \quad (12)$$

在 MLA 算法中, $J_k(w)$ 的二次近似值可由(12)式展开为

$$J_{k+1}(w) = \lambda J_k(w) + [E_i(w_k) + (w - w_k)^T \cdot \nabla E_i(w_k)]^2, \quad (13)$$

式中, $0 < \lambda < 1$ 表示遗忘因子. 为了导出递归学习过程, 可将(10)式代入(13)式中的 $J_k(w)$, 并整理得

$$J_{k+1}(w) = E_i^2(w_k) + \lambda \alpha_k + 2E_i(w_k)(w - w_k)^T \cdot \nabla E_i(w_k) + (w - w_k)^T [\lambda H_k^{-1} + \nabla E_i(w_k) \cdot \nabla E_i^T(w_k)] (w - w_k)^T. \quad (14)$$

由(10)式可得

$$\begin{aligned} g_{k+1}(w) &= \alpha_{k+1} + (w - w_{k+1})^T H_{k+1}^{-1} (w - w_{k+1}) \\ &= [\alpha_{k+1} + (w_k - w_{k+1})^T H_{k+1}^{-1} (w_k - w_{k+1})] + 2(w - w_k)^T H_{k+1}^{-1} (w_k - w_{k+1}) \\ &\quad + (w - w_k)^T H_{k+1}^{-1} (w - w_k). \end{aligned} \quad (15)$$

令 $J_{k+1}(w) = g_{k+1}(w)$, 得到

$$E_i^2(w_k) + \lambda \alpha_k = \alpha_{k+1} + (w_k - w_{k+1})^T H_{k+1}^{-1} (w_k - w_{k+1}), \quad (16)$$

$$H_{k+1}^{-1} = \lambda H_k^{-1} + \nabla E_i(w_k) \cdot \nabla E_i^T(w_k), \quad (17)$$

$$E_i(w_k) \cdot \nabla E_i(w_k) = H_{k+1}^{-1} (w_k - w_{k+1}). \quad (18)$$

利用矩阵逆定理, 由(17)式得

$$H_{k+1} = \lambda^{-1} (H_k - \frac{H_k \nabla E_i(w_k) \cdot \nabla E_i^T(w_k) H_k}{\beta_k}), \quad (19)$$

其中 $\beta_k = \lambda + \nabla E_i^T(w_k) H_k \cdot \nabla E_i(w_k)$. (20)

在(17)式中, 两边同乘以 H_k 得 $H_{k+1}^{-1} = \beta_k H_k^{-1}$, 并代入(18)式, 经整理后, 得到最终神经网络权值的 MLA 学习算法

$$\left. \begin{aligned} w_{k+1} &= w_k - H_k E_i(w_k) \cdot \nabla E_i(w_k) / \beta_k \\ H_{k+1} &= \lambda^{-1} (H_k - H_k \nabla E_i(w_k) \cdot \nabla E_i^T(w_k) H_k) / \beta_k \\ \beta_k &= \lambda + \nabla E_i^T(w_k) \cdot H_k \cdot \nabla E_i(w_k), H_1 = I (\text{单位阵}) \end{aligned} \right\} \quad (21)$$

其中 $\nabla E(w_k^{(2)}) = \frac{\partial E_i}{\partial w^{(2)}} = -(\tau - \tau_n) \frac{\partial O_j^{(2)}}{\partial w^{(2)}} = -(\tau - \tau_n) \cdot O_j^{(1)}(k)$, (22)

$$\nabla E(w_k^{(1)}) = \frac{\partial E}{\partial O_j^{(2)}} \cdot \frac{\partial O_j^{(2)}}{\partial O_j^{(1)}} \cdot \frac{\partial O_j^{(1)}}{\partial w_k^{(1)}} = -(\tau - \tau_n) w_j^{(2)} \cdot f'(I^{(1)}) \cdot \chi_i(k), \quad (23)$$

$f'(I^{(1)}) = \frac{1}{2} [1 - f^2(I^{(1)})]$, $\chi_i(k)$ 为网络输入.

4 仿真实验结果

为了证实所提控制策略和神经网络学习算法的有效性,本文对两关节的机器人模型进行数字仿真.考虑两关节机器人模型^[5]为

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}, \quad (24)$$

其中

$$M_{11} = m_1 l_1^2 + m_2 (l_1^2 + l_2^2 + 2l_1 l_2 \cos(\theta_2)),$$

$$M_{12} = M_{21} = m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2),$$

$$M_{22} = m_2 l_2^2,$$

$$V_1 = -m_2 l_1 l_2 \sin(\theta_2) \cdot \dot{\theta}_2 - 2m_2 l_1 l_2 \sin(\theta_2) \cdot \dot{\theta}_1 \dot{\theta}_2,$$

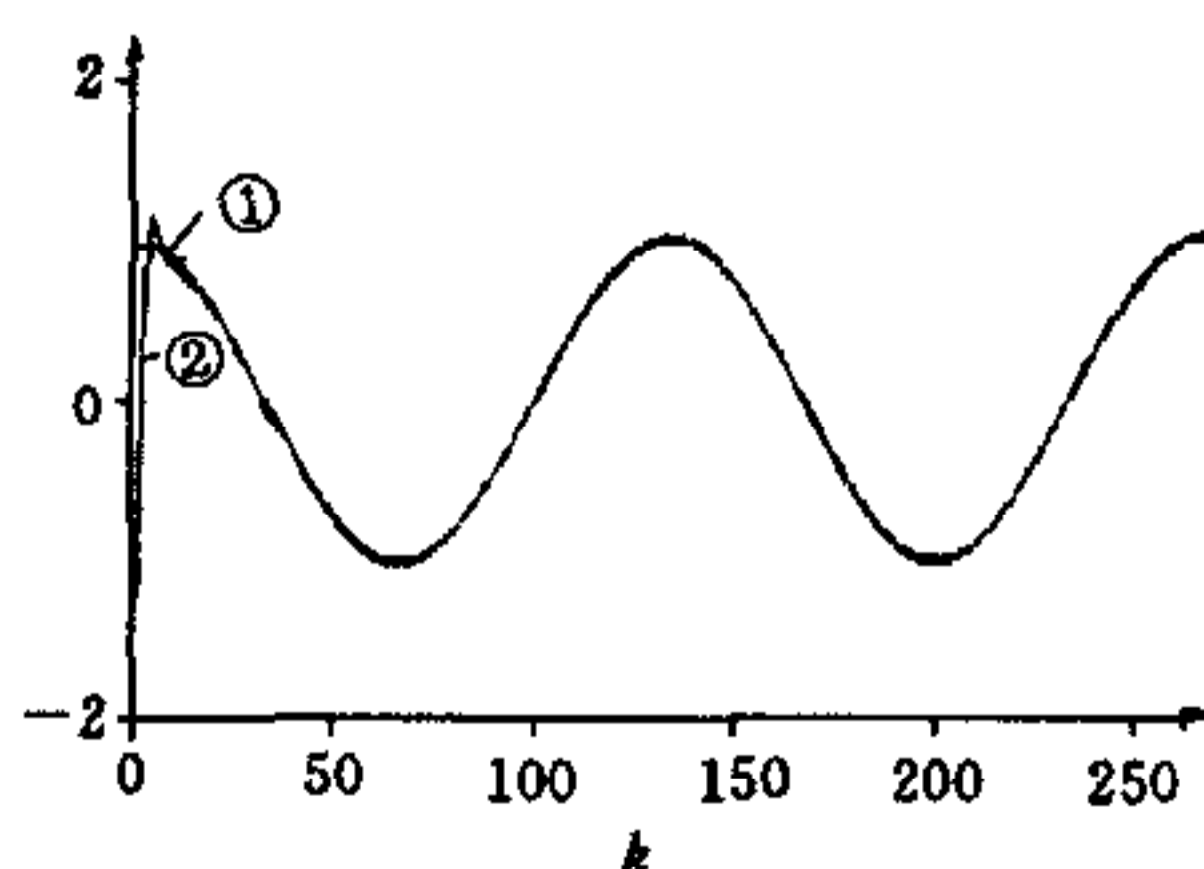
$$V_2 = m_2 l_1 l_2 \sin(\theta_2) \cdot \dot{\theta}_1^2,$$

$$G_1 = m_2 l_2 g \cos(\theta_1 + \theta_2) + (m_1 + m_2) l_2 g \cdot \cos(\theta_1),$$

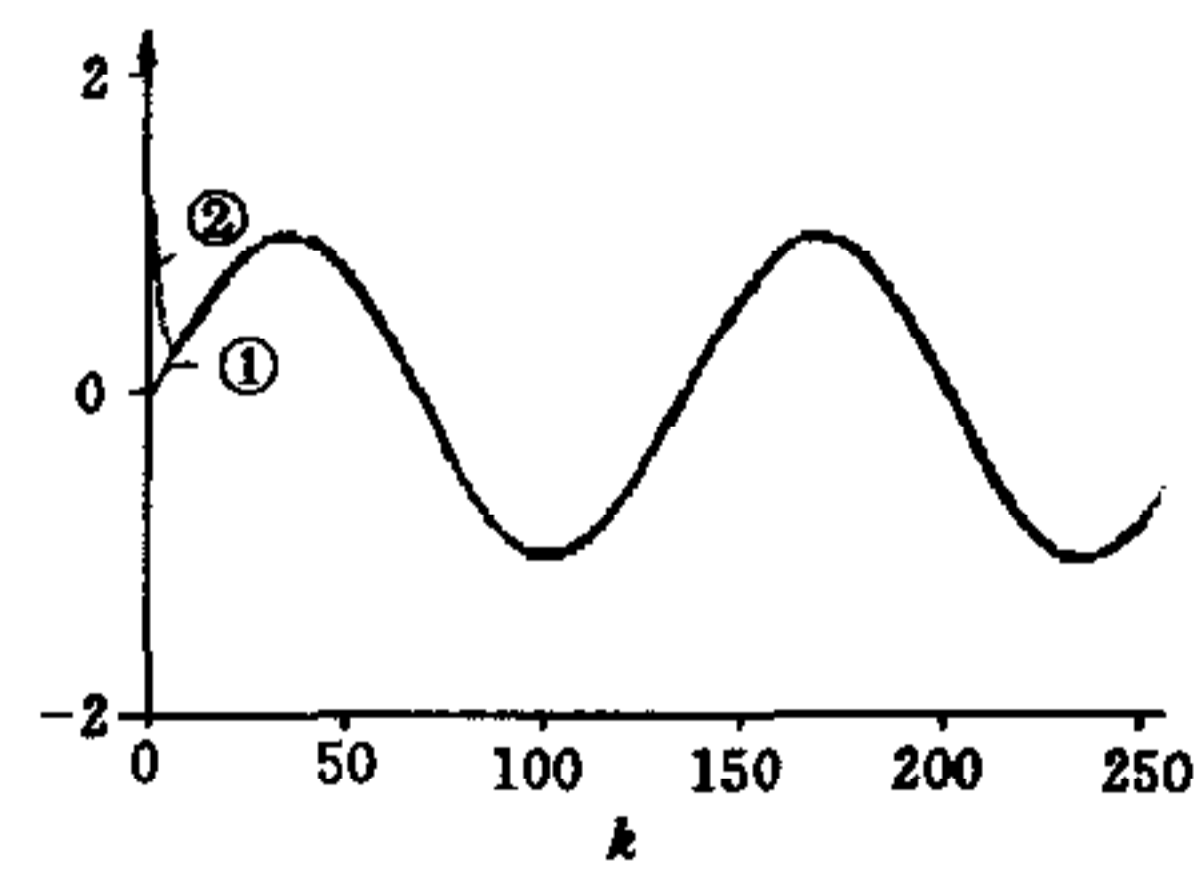
$$G_2 = m_2 l_2 g \cos(\theta_1 + \theta_2).$$

仿真中,采样周期 $T=0.005s$, $K_p=[1000,2000]^T$, $k_d=[50,60]^T$, $k_i=[10,10]^T$.神经网络结构: NNC_1 为3-6-1节点数,输入 $\{\ddot{\theta}_{d1}, \dot{\theta}_{d1}, \theta_{d1}\}$,输出 u_{n1} ; NNM_1 为3-8-1,输入 $\{\ddot{\theta}, \dot{\theta}, \theta\}$,输出 τ_{n1} ; NNC_2 为3-6-1,输入 $\{\ddot{\theta}_{d2}, \dot{\theta}_{d2}, \theta_{d2}\}$,输出 u_{n2} ; NNM_2 为3-8-1,输入 $\{\ddot{\theta}_2, \dot{\theta}_2, \theta_2\}$,输出 τ_{n2} .取不同参数 l_1, l_2, m_1, m_2 ,均能取得良好的控制效果.

例如,一组参数 $l_1=1.2m$, $l_2=0.8m$, $g=9.8m/s^2$,质量 $m_1=10kg$, $m_2=5kg$,选取最大控制力矩 $|u_{1max}|=1000N \cdot M$, $|u_{2max}|=500N \cdot M$,给定期望轨迹: $\theta_{d1}=\cos(2\pi k)$, $\dot{\theta}_{d1}=-2\pi\sin(2\pi k)$, $\theta_1(0)=-1.2$, $\theta_{d2}=\sin(2\pi k)$, $\dot{\theta}_{d2}=2\pi\cos(2\pi k)$, $\theta_2(0)=1.2$ (便于观察),仿真结果如图2所示.从图中可以看出,NNC和NNM网络经过85次学习后,实际输出 θ_1, θ_2 接近期望轨迹.其它仿真结果略.



(a) ①— θ_{d1} , ②— $\theta_1(k)$



(b) ①— θ_{d2} , ②— $\theta_2(k)$

图2 位置跟踪响应曲线

5 结束语

本文提出的自学习控制器利用了ANN的自学习和自适应特点,通过在线边学习边控制,达到很高的跟踪精度控制,同时利用了MLA学习算法提高了网络的学习收敛性,适合于机器人的快速实时控制.

参 考 文 献

- [1] Sun Yuan Kung, Teng Heng Hwang. Neural network architectures for robotic application. *IEEE Trans on Robotics and Automation*, 1989, **5**(5): 641—650.
- [2] Albert Y, Zomaya, Tarek M Nabhan. Centralized and decentralized neuro-adaptive robot controllers. *Neural Network*, 1993, **6**(2): 223—244.
- [3] Mitsuo, Kawato, Yoji uno *et al.* Hierarchical neural network model for voluntary movement with application to robotics. *IEEE control system Magazine*. 1988, **13**(2): 1129—1134.
- [4] 王耀南. 智能控制系统——模糊逻辑、专家系统、神经网络控制. 长沙: 湖南大学出版社, 1996. 204—207.
- [5] 周远清, 张再兴. 智能机器人系统. 北京: 清华大学出版社, 1989, 25—48.

SELF-LEARNING CONTROLLER BASED ON NEURAL NETWORKS FOR ROBOTIC MANIPULATOR

WANG YAONAN

(Department of Electrical Engineering, Huanan University, Changsha 410082)

Abstract This paper presents a new self-learning controller based on neural networks for robotic manipulator. A fast learning algorithm of neural networks is proposed to improve both speed and convergence of the learning process. Simulation results of a two-link robot show that the proposed method can give more significant performance and robustness than conventional approaches.

Key words Neural networks, learning algorithm, robot control.