

文章编号: 1001-0920(2003) 03-0263-04

# 即时学习算法在非线系统迭代学习控制中的应用

孙 维<sup>1</sup>, 王 伟<sup>2</sup>, 朱瑞军<sup>2</sup>

(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110004; 2. 大连理工大学 信息与控制研究中心, 辽宁 大连 116023)

**摘 要:** 运用即时学习算法来解决一类非线性系统的迭代学习控制初值问题。对于任何类型的迭代学习控制算法, 即时学习算法都能有效地估计初始控制量, 减小了初始输出误差, 加快了算法的收敛速度, 使得经过有限次迭代后系统输出能严格跟踪理想信号。对机器人系统的仿真结果表明了该方法的有效性。

**关键词:** 即时学习算法; 迭代学习控制; 初值控制量; 非线性系统

中图分类号: TP13

文献标识码: A

## Iterative learning control for nonlinear system using lazy learning method

SUN Wei<sup>1</sup>, WANG Wei<sup>2</sup>, ZHU Rui-jun<sup>2</sup>

(1. School of Information Science and Engineering, Northeastern University, Shenyang 110004, China;

2. Information and Control Research Center, Dalian University of Technology, Dalian 116023, China)

**Abstract:** A method is proposed to solve the problem of the initial control input using lazy learning method. For any kind of iterative learning control algorithms, the proposed method reduces the initial output error and increases the convergence speed greatly when tracking a desired trajectory. The simulation for a robot system shows the effectiveness of the method.

**Key words:** Lazy learning method; Iterative learning control; Initial control value; Nonlinear system

### 1 引 言

即时学习算法<sup>[1~3]</sup>是近年出现的一种新的非线性系统建模和控制方法, 在不需要系统其他先验知识的条件下, 它仅利用非线性系统大量的输入输出数据, 即可在线辨识系统模型和设计控制器参数, 以达到系统动静态品质的要求。在此过程中, 即时学习算法并不保留系统所建立的局部即时模型和控制器参数, 只存贮系统新的输入输出数据。作为多模型方法中一种新的非线性系统的建模和控制算法, 其思想简单, 实施方便, 受到了人们的普遍关注。

通过对高速运动机械手的控制而提出的迭代学习控制方法近年来得到了迅速的发展。它仅利用

被控对象先前的控制经验和较少的计算量, 通过对一个简单的初始控制信号的修正, 来控制不确定性动态系统跟踪理想的周期信号。在对各种机器人系统的控制等问题中, 各种形式的迭代学习算法得到了广泛的应用<sup>[4~7]</sup>。在迭代学习控制算法中, 初始控制信号的选择很大程度上决定了算法的收敛速度和控制精度。文献[8]利用系统已知的输入输出数据和即时学习算法来估计迭代学习控制算法的初始控制信号  $u_0(t)$ , 显著地提高了系统的收敛速度。但其中存在两个问题: 其一, 在每个采样时刻都用即时学习算法建立局部线性逆控制模型来计算初始控制信号  $u_0(t)$ , 增加了算法的复杂性; 其二, 得到的初始控

收稿日期: 2002-04-22; 修回日期: 2002-07-15。

作者简介: 孙维(1975—), 女, 山东海阳人, 博士生, 从事多模型自适应控制、计算机集成制造等研究; 王伟(1955—), 男

(满族), 辽宁鞍山人, 教授, 博士生导师, 从事自适应控制、模型预测控制等研究。 reserved. <http://www.cnki.net>

制信号  $u_0(t)$ , 没有相应的校验算法来确定其是否能够产生接近理想输出信号  $y_d(t)$ , 也无法解决数据中的“坏点”引起的建模误差。本文解决了这两个问题, 在运用  $k$ - $NN$  方法建立逆控制模型得到  $u_0(t)$  的同时, 建立系统的输入输出模型来校验  $u_0(t)$  所产生的系统输出; 在系统动特性变化缓慢的情况下, 采用自适应算法调节前一时刻系统即时模型的参数, 以减少辨识算法的在线计算量, 加快系统的响应速度。在对单关节机器人系统的仿真实验中, 验证了该方法的有效性。

## 2 迭代学习控制算法的初值问题

设非线性系统的状态方程和输出方程为

$$\begin{cases} \dot{x}(t) = f[x(t), u(t), t] \\ y(t) = h[x(t), u(t), t] \end{cases} \quad (1)$$

其中:  $x \in \mathcal{R}^n, y \in \mathcal{R}^m; u \in \mathcal{R}^k, f(\cdot), h(\cdot)$  为非线性函数。系统的控制目标是在时间  $t \in [0, T_f]$  内的输出跟踪理想输出  $y_d(t)$ 。引入迭代学习控制律, 设第  $i$  次迭代学习的控制输入为  $u_i(t)$ , 则

$$u_{i+1}(t) = u_i(t) + \delta u_i(t), \quad 0 \leq t \leq T_f \quad (2)$$

典型 PID 型迭代学习控制算法的学习律  $\delta u_i(t)$

具有如下形式

$$\begin{cases} \delta u_i(t) = g(\dot{e}_i(t), e_i(t)) = \\ G_{de}\dot{e}_i(t) + G_{pe}e_i(t) + G_{i} \int_0^t e_i(t) dt \\ e_i(t) = y_d(t) - y_i(t) \end{cases} \quad (3)$$

通过对初始控制量的多次迭代运算, 得到一个理想控制信号  $u_d(t)$ , 使被控对象产生期望的运动。设初始控制信号为  $u_0(t)$ , 则式(2)描述的迭代学习控制算法可记为

$$u_{i+1}(t) = u_0(t) + \sum_{j=0}^i g(\dot{e}_j(t), e_j(t)) \quad \forall t \in [0, T_f] \quad (4)$$

以往迭代学习控制的研究工作多集中在如何设计学习函数  $g(\cdot)$ , 使得系统的输出误差  $e_i(t)$  能够快速而平滑地收敛到零值<sup>[4-7]</sup>, 而初始控制信号  $u_0(t)$  由经验确定。若  $u_0(t)$  与理想控制信号  $u_d(t)$  之间存在较大的误差, 则初始的迭代误差也很大, 使算法的迭代次数增加, 不能达到实际过程控制要求。因此, 引入即时学习算法来估计初始控制信号  $u_0(t)$ 。

### 3 用即时学习算法估计初始控制信号

#### 3.1 即时学习算法的基本原理

对于由式(1)描述的非线性系统, 设已知系统大量的信息向量数据  $\Psi_i = [x_i, y_i, u_i], i \in [1, N]$ ,

即时学习算法所要解决的问题是: 当已知理想输出信号  $y_d(t)$  和理想状态信号  $x_d(t)$  时, 利用这些数据来估计理想的控制信号  $u_d(t)$ 。设系统的逆控制模型可描述为

$$u_d(t) = h^{-1}(x_d(t), y_d(t)) = z(x_d(t), y_d(t)) \quad (5)$$

每一采样时刻系统的理想信号为  $q(t) = [x_d(t), y_d(t)]$ , 文献[8]采用如下形式的即时模型来估计

$$u_d(t) \begin{cases} \hat{z}_n(x(t)) = \alpha_0 + \alpha_1 x(t) + \dots, \quad \text{for } n = 1, \\ z(x(t)) = \alpha_0 + \alpha_{1x_1}(t) + \alpha_{2x_2}(t) + \\ \alpha_{3x_1^2}(t) + \alpha_{4x_2^2}(t) + \\ \alpha_{5x_1x_2} + \dots, \quad \text{for } n = 2 \end{cases} \quad (6)$$

通常情况下取线性模型建立逆控制模型。设  $\hat{\alpha}^T = [\alpha_0, \alpha_1, \alpha_2]$ , 则在  $N$  组数据中选择  $k$  组最接近  $q(t)$  数据, 运用  $k$ - $NN$  的建模算法可得

$$\hat{\alpha} = [(WX)^T WX]^{-1} (WX)^T Wu \quad (7)$$

式中:  $X$  为由  $\Psi_i$  中的状态信号组成的  $k \times (n+1)$  阶矩阵;  $W$  为  $k \times k$  阶对角矩阵;  $W_{ii} = w_i$  为第  $i$  组数据赋予的权值, 通常取为  $\Psi_i$  与  $q(t)$  之间欧式距离的高斯函数

$$w_i = \frac{K(dE)}{K(dE)} = \frac{a_0 \exp\left(-\left(\frac{dE - d_0}{2\sigma_0}\right)^2\right)}{dE(\Psi_i, q(t)) = (\Psi_i - q(t))^T (\Psi_i - q(t))} \quad (8)$$

设  $\mathcal{Q}(t) = [1, x_d(t)]^T$ , 则在  $q(t)$  处可得到控制信号的估计值

$$\hat{u}_d(t) = \hat{z}(t) = \hat{\alpha}^T \mathcal{Q}(t) \quad (9)$$

通过上式得到的  $\hat{u}_d(t)$  作为迭代学习控制算法的初始控制信号  $u_0(t)$ , 能够减少算法的初始输出误差, 加快算法的收敛速度。这种算法不仅没有相应的算法校验  $\hat{u}_d(t)$  是否能够产生接近  $y_d(t)$  的输出信号, 而且在线计算量大。为此, 本文通过建立即时输入输出模型和采用自适应算法来解决这些问题。

#### 3.2 建立即时输入输出模型校验控制信号的估计值

在  $q(t) = [x_d(t), y_d(t)]$  处通过建立式(9)的即时逆控制模型得到控制信号的估计值  $\hat{u}_d(t)$  后, 运用即时学习算法中的  $k$ - $NN$  建模方法和如下形式的即时输入输出模型估计系统在  $\hat{\phi}(t) = [x_d(t), u_d(t), 1]^T$  处的输出值

$$y_d(t) = \hat{\phi}_d^T(t) [(W\hat{\Phi})^T W\hat{\Phi}]^{-1} (W\hat{\Phi})^T Wy = \hat{\phi}_d^T(t) \hat{p}(t) \quad (10)$$

式中:  $\Phi$  为  $k$  组信息向量  $\Phi = [x_i, w, 1]$  构成的  $k \times (n + p - 1)$  阶矩阵;  $W$  为  $k \times k$  阶信息向量的权值矩阵, 其参数的定义同式 (8), 为参与建模的数据分配一个与  $q(t)$  之间距离相关的权值, 可以增加与  $q(t)$  较近数据在即时模型中的比重, 使即时模型能够更有效地逼近在  $q(t)$  处系统的非线性;  $\hat{\beta}(t) = [\beta_0, \beta_1, \dots, \beta_{n+p-2}]$  为模型参数向量。若满足条件  $y_d(t) - \hat{y}_d(t) < \mu, \mu > 0$ , 则采用  $\hat{u}_d(t)$  作为此刻迭代学习控制算法的初值  $u_0(t)$ ; 若  $y_d(t) - \hat{y}_d(t) > \mu$ , 为去除建模数据中的“坏点”对估计结果的影响, 按照

$$\zeta_i(t) = \frac{y_i - \Phi_i(t) \hat{\beta}(t)}{1 - \Phi_i(t) \hat{\beta}(t)}, \quad 1 \leq i \leq k \quad (11)$$

计算每组数据向量  $\Phi$  预测值与真实值之间的误差。计算  $\zeta_i(t)$ ,  $i \in [1, k]$ , 把  $\zeta_i(t)$  最大的一组数据  $\mathcal{Q}$  去掉, 即去掉参与建模的“坏点”数据, 再在系统剩余的信息向量中选择一组最接近  $q(t)$  的数据, 运用式 (9) 和 (10) 分别计算控制信号和输出信号的估计值  $\hat{u}_d(t)$  和  $\hat{y}_d(t)$ , 直至  $y_d(t) - \hat{y}_d(t) < \mu$ , 所得的  $\hat{u}_d(t)$  为此刻迭代学习控制算法的初值  $u_0(t)$ , 即  $u_0(t) = \hat{u}_d(t), t \in [0, T_f]$ 。

### 3.3 用自适应模型减少即时学习算法的计算量

文献 [8] 中在每一采样时刻都根据不同的  $q(t) = [x_d(t), y_d(t)]$ , 依照即时学习算法建立逆控制模型来估计此刻  $u_d(t)$  值, 使算法的在线计算量大, 响应速度减慢。当系统的动特性变化缓慢时, 可以用自适应模型近似地逼近非线性系统。

设系统的采样周期为  $T_c (T_c \ll T_f)$ ,  $t$  时刻所得到的逆控制模型参数向量为  $\alpha(t)$ , 输入输出模型参数向量为  $\beta(t)$ , 则在  $t + T_c$  时刻, 采用如下自适应算法调节前一时刻系统模型参数向量的值

$$\begin{cases} \hat{\beta}(t + T_c) = \hat{\beta}(t) + \frac{y_d(t) - \Phi(t) \hat{\beta}(t)}{1 + \Phi(t) \hat{\beta}(t)} \Phi^T(t) \\ \hat{\alpha}(t + T_c) = \hat{\alpha}(t) + \frac{y_d(t) - \Phi(t) \hat{\beta}(t)}{1 + \Phi(t) \hat{\beta}(t)} \mathcal{Q}^T(t) \end{cases} \quad (12)$$

设  $\Phi(t + T_c) = [x_d(t + T_c), u_d(t + T_c), 1]^T, \mathcal{Q}(t + T_c) = [x_d(t + T_c), 1]$ , 则在  $t + T_c$  时刻由自适应模型得到的控制量和系统输出的估计值为

$$\begin{cases} \hat{u}_d(t + T_c) = \mathcal{Q}(t + T_c)^T \hat{\alpha}(t + T_c) \\ \hat{y}_d(t + T_c) = \Phi(t + T_c)^T \hat{\beta}(t + T_c) \end{cases} \quad (13)$$

设  $\epsilon > 0$  为一适当选择的正的参数, 若  $y_d(t + T_c) - \hat{y}_d(t + T_c) < \epsilon$ , 则采用自适应模型的估计值

$\hat{u}_d(t + T_c)$  作为此刻迭代学习控制算法的初始控制信号  $u_0(t + T_c)$ , 即在系统参数慢时变或非 线性特性不强的前提下, 式 (12) 描述的自适应律能够使系统跟踪理想的信号; 否则, 采用 3.2 节的方法建立即时逆控制模型和输入输出模型的方法得到  $u_d(t + T_c)$ 。

综上所述, 在给定理想输出信号  $y_d(t)$  和理想状态信号  $x_d(t)$  时, 采用本文的即时学习算法估计理想控制信号  $u_d(t)$ , 并将其作为迭代学习控制算法的初始控制信号的具体运算步骤如下:

Step1:  $t$  时刻, 在系统的信息向量中选择  $k$  组最接近  $[x_d(t), y_d(t)]$  的数据, 根据式 (9) 建立逆控制模型, 得到  $u_d(t)$ ; 然后根据式 (10) 建立即时输入输出模型, 计算  $y_d(t)$ , 给定一正数  $\mu > 0$ , 若  $y_d(t) - \hat{y}_d(t) < \mu$ , 则  $u_0(t) = u_d(t)$ , 转 Step3, 否则转 Step2;

Step2: 根据式 (11) 计算参与建立即时模型的每组数据的输出估计值与实际值之间的误差  $\zeta_i(t), i \in [1, k]$ , 计算  $J = \arg \max_{i \in [1, k]} \zeta_i(t)$ , 则去掉参与建模的第  $J$  组数据, 在系统剩余的信息向量中选择一组最接近  $[x_d(t), y_d(t)]$  的数据, 转 Step1;

Step3: 若  $t + T_c > T_f$ , 转 Step5, 否则, 根据式 (12) 自适应算法调整  $t$  时刻即时逆控制模型和输入输出模型参数向量  $\hat{\alpha}(t), \hat{\beta}(t)$ , 得到此刻参数向量的估计值  $\hat{\alpha}(t + T_c), \hat{\beta}(t + T_c)$ , 由式 (12) 可得到  $u_d(t + T_c), y_d(t + T_c)$ ;

Step4: 给定一正数  $\epsilon > 0$ , 若  $y_d(t + T_c) - \hat{y}_d(t + T_c) < \epsilon$ , 则  $u_0(t + T_c) = u_d(t + T_c)$ ; 否则, 转 Step1;

Step5: 算法结束,  $u_0(t) = u_d(t), t \in [0, T_f]$ 。

## 4 仿真实验

考虑单关节的机械手臂系统, 其连续时间域内系统的动态方程为

$$I_m \ddot{\theta}(t) + \left(\frac{1}{2}m + M\right)gl \sin(\theta(t)) = \tau(t) \quad (14)$$

式中:  $\theta(t)$  为机械手旋转角度, 单位用弧度表示,  $\tau(t)$  为作用于节点的力矩,  $m$  为机械手的质量,  $M$  为负载的质量,  $l$  为机械手的长度, 取  $M = 2 \text{ kg}, m = 1 \text{ kg}, l = 0.5 \text{ m}, g = 9.8 \text{ m/s}^2$ 。  $I_m$  为关节处的惯性力矩, 定义为

$$I_m = Ml^2 + \frac{1}{3}ml^2 \quad (15)$$

设系统的采样周期  $T_c = 0.01 \text{ s}$ , 将式 (15) 离散化后可得系统的状态方程和输出方程如下

$$\begin{cases} x_1(t+1) = x_2(t) \\ x_2(t+1) = 2x_2(t) - x_1(t) + \frac{T_c^2}{I_m}(\tau(t) - \\ (0.5m + M)g \sin(x_1(t))) \end{cases} \quad (16)$$

$$\theta(t) = x_1(t)$$

采用如下形式的 D 型迭代学习控制算法构成控制器

$$u_{i+1}(t) = u_i(t) + G_d \hat{e}_i(t) \quad (17)$$

$i$  表示算法的迭代次数,且  $e(t) = \theta_i(t) - \theta(t)$ , 为保证输出误差的平滑收敛,取  $G_d = 1$ 。设迭代学习控制算法的能量指标函数为

$$J_m = \frac{1}{T_f} \int_0^{T_f} e^2(t) dt = \frac{1}{N_m} \sum_{j=0}^{N_m} e_j^2 \quad (18)$$

首先,分别取  $\theta_i(t), t \in [0, 1]$  为正弦函数和拐点分别在 0 和 1 处的 S 函数,单独采用式(17) 所示的迭代学习控制算法控制式(16) 描述的系统,可得  $N = 1\ 100$  组数据,  $\Psi_i = [x_{1,i}, x_{2,i}, y_i, \tau_i], i \in [1, 1\ 100]$ 。

取理想输出为  $\theta(t) = t - t^2, t \in [0, 1]$ , 参数  $\mu = 0.03, \epsilon = 0.01$ , 参与建模的数据组数  $k = 13$ 。如图 1 所示,虚线为运用本文方法的估计迭代学习控

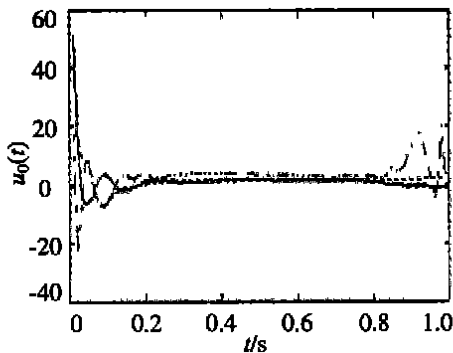


图 1 初始控制信号估计值的比较图

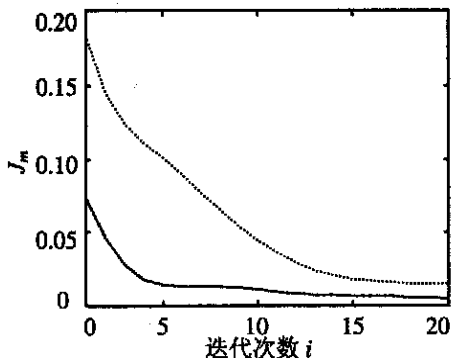


图 2 能量指标函数的比较图

制算法的初始控制量  $\hat{\tau}_0(t)$ , 实线为理想控制信号  $\tau(t)$  与运用文献[8] 算法得到的初始控制信号(点划线)的曲线。

图 2 所示为两种算法的能量指标函数的比较曲线,实线为本文算法的结果,虚线为文献[8] 算法的仿真结果。从中可以看出,本文方法不仅减小了初始时刻控制信号的误差值,而且加快了算法的收敛速度。

### 5 结 论

迭代学习控制算法的研究大多集中在迭代率的设计并证明算法收敛性问题上,而初值控制量的选择,能极大地影响算法的收敛性。本文根据迭代学习控制的特点,运用即时学习算法来解决一类非线性系统的迭代学习控制初控制量问题,在不影响迭代学习控制算法收敛性的前提下,它能有效地估计初始控制量的值,减小了初始输出误差,加快了算法的收敛速度,使得经过有限次迭代后系统输出能严格跟踪理想信号。

### 参考文献(References):

[1] Christopher G A, Andrew W M, Stefan S. Locally weighted learning [J]. *Artificial Intelligence Review*, 1997, 11(1): 11-73.

[2] Christopher G A, Andrew W M, Stefan S. Locally weighted learning for control[J]. *Artificial Intelligence Review*, 1997, 11(1): 75-113.

[3] Gianluca B, Mauro B, Hugues B. Lazy learning for local modelling and control design[J]. *Int J of Control*, 1999, 72(7/8): 643-658.

[4] Arimoto S, Kawamura S, Miyazaki F. Bettering operation of robotics by learning[J]. *J of Robotic Systems*, 1984, 1(2): 123-140.

[5] Bienz Z, Hwang D H, Oh S R. A nonlinear iterative learning method for robot path control[J]. *Robotica*, 1991, 9(4): 387-392.

[6] Xu J X. Direct learning of control efforts for trajectories with different magnitude scales[J]. *Automatica*, 1997, 33(12): 2191-2195.

[7] Xu J X, Xu J, Gao W J. A PD type fuzzy logic learning control approach for repeatable tracking control tasks[J]. *Acta Automatic Sinica*, 2001, 27(4): 434-446.

[8] Arif M, Ishihara T, Inooka H. Incorporation of experience in iterative learning controllers using locally weighted learning[J]. *Automatica*, 2001, 37(6): 881-888.