



基于广义基函数的 CMAC 学习算法的 改进及收敛性分析

段培永 邵惠鹤

(上海交通大学自动化系 上海 200030)

摘 要 基于广义基函数的 CMAC (Cerebellar Model Articulation Controller) 学习算法 (称 C-L 算法) 收敛条件依赖于基函数和学习样本, 很难同时满足学习快速性与收敛性. 提出了一种改进学习算法, 并证明改进算法是收敛的, 而且收敛条件不依赖于基函数和学习样本. 仿真结果表明改进算法优于 C-L 算法和标准的 Albus 算法.

关键词 CMAC, 学习算法, 基函数网络.

IMPROVED ALGORITHM OF CMAC WITH GENERAL BASIS FUNCTION AND ITS CONVERGENCE ANALYSIS

DUAN Peiyong SHAO Huihe

(Department of Automation, Shanghai Jiaotong University, Shanghai 200030)

Abstract The convergence of the learning algorithm for the CMAC (Cerebellar Model Articulation Controller) with general basis functions, presented by Chiang & Lin, is associated with the selected basis functions and the sample data during learning. Therefore, it is difficult for the algorithm to obtain high learning speed as well as learning convergence. In this paper, we propose an improved algorithm and prove that its convergence does not depend upon the choice of basis functions or sample data. The simulation results demonstrate that the improved algorithm has a better learning performance than C-L algorithm and general Albus algorithm.

Key words CMAC, learning algorithm, basis function networks.

1 C-L 算法及其局限性

传统的 CMAC^[1] 基函数为常数, 只能记忆静态信息, 而且泛化性能差^[2]. Chiang & Lin 采用可微分基函数, 不但提高了 CMAC 的输出精度, 而且增强了泛化能力, 同时还可

获得微分信息^[3]. m 维紧集 $U = A_1 \times A_2 \times \dots \times A_m$, 满足 $\forall \mathbf{u}_k = [u_{k1}, u_{k2}, \dots, u_{km}]^T \in U$, 有 $u_{ki} \in [u_i^-, u_i^+] = A_i \subset R, (i=1, 2, \dots, m)$, R 为实数集, u_i^-, u_i^+ 为 A_i 的下、上界. 样本输出 $y_k \in R$, 样本集 $S = \{(\mathbf{x}_k, y_k)\}, k=1, 2, \dots, N_s, \mathbf{x}_k = [x_{k1}, x_{k2}, \dots, x_{km}]^T \in U$, 输出样本 $y_k \in R$. 把 U 量化为 L 个超立方体, 记为 $C_j (j=1, 2, \dots, L)$, 基函数 $b_j(\cdot)$ 定义在 C_j 上. $\forall \mathbf{x}_k \in S$, 在 U 上有 N_e 个超立方体包含 \mathbf{x}_k , 即 $\mathbf{x}_k \in C_{q_k} \cap C_{q_k} \cap \dots \cap C_{r_k}$, 记指标集 $U_k = \{p_k, q_k, \dots, r_k\}$, 则 CMAC 输出是超立方体簇 $\{C_l | l \in U_k\}$ 上基函数的线性组合 $\hat{y}_k = \mathbf{S}_k^T B(\mathbf{x}_k) \mathbf{q} = \sum_{l=1}^L s_{ll} b_l(\mathbf{x}_k) q_l = \sum_{l \in U_k} b_l(\mathbf{x}_k) q_l$, 其中 $B(\mathbf{x}_k) = \text{diag}[b_1(\mathbf{x}_k), b_2(\mathbf{x}_k), \dots, b_L(\mathbf{x}_k)]$, $\mathbf{q} = [q_1, q_2, \dots, q_L]^T$ 为权系数向量, \mathbf{S}_k 为权系数选择向量, 有 N_e 个元素为 1, 其余为 0. 记 $e_k = y_k - \hat{y}_k$, 则 C-L 学习算法为^[3]

$$\Delta \mathbf{q}_{k-1} = \frac{\alpha e_{k-1} B(\mathbf{x}_{k-1}) \mathbf{S}_{k-1}}{N_e}, \quad (1)$$

其收敛条件为: $0 < \alpha < \min_{k \in \{1, 2, \dots, N_s\}} \frac{2N_e}{\sum_{l \in U_k} b_l^2(\mathbf{x}_k)}$.

显然, α 的选取受基函数与学习样本的限制. 实际上, 在基函数已确定的情况下选择 α 时有两种情况: 1) 样本在学习前已知, α 的上限尽管可以求出, 但很繁琐; 2) 样本在学习前未知, 如在线学习时, α 上限事先无法确定. 此时, 为了保证收敛, 使 α 较小, 从而势必大大减小学习速度. 实际应用表明 C-L 算法存在收敛速度慢的问题.

2 改进的 C-L 算法及其收敛性分析

为了克服 C-L 算法收敛条件存在的缺陷, 受参数自适应算法^[4]的启发, 提出了一种改进的 CMAC 学习算法, 即把式(1)改写为

$$\Delta \mathbf{q}_{k-1} = \frac{\alpha e_{k-1} B(\mathbf{x}_{k-1}) \mathbf{S}_{k-1}}{\beta + \mathbf{S}_{k-1}^T B(\mathbf{x}_{k-1}) B^T(\mathbf{x}_{k-1}) \mathbf{S}_{k-1}},$$

其中 α, β 为常数, 记 $f_{k-1} = \mathbf{S}_{k-1}^T B(\mathbf{x}_{k-1}) B^T(\mathbf{x}_{k-1}) \mathbf{S}_{k-1} = \sum_{l \in U_s} b_l^2(\mathbf{x}_{k-1})$. 对样本数据的学习采用批量学习, 即把所有样本学习一遍后, 再从第一个样本开始把每个样本学习一遍... 依此类推, 直到 CMAC 的输出满足要求为止. 每学习一遍称为一个循环. 记第 k 个样本在第 i 次循环学习时, CMAC 的输出误差为 $e_k^{(i)} = y_k - \hat{y}_k^{(i)}$, y_k 为期望样本输出值, $\hat{y}_k^{(i)}$ 为输入为 \mathbf{x}_k 时 CMAC 联想输出值, $\mathbf{q}_k^{(i)}$ 为第 k 个样本在第 i 次循环学习前的权系数向量. 特别地, $\mathbf{q}_1^{(0)} = \mathbf{q}_0$ 为样本学习初始值. 记

$$\mathbf{q}_k^{(i+1)} = \mathbf{q}_k^{(i)} + D \mathbf{q}_k^{(i)}, \quad (2)$$

$$D \mathbf{q}_k^{(i)} = \mathbf{q}_k^{(i+1)} - \mathbf{q}_k^{(i)} = \mathbf{q}_{k-1}^{(i+1)} + \Delta \mathbf{q}_{k-1}^{(i+1)} - (\mathbf{q}_{k-1}^{(i)} + \Delta \mathbf{q}_{k-1}^{(i)}) = (\mathbf{q}_{k-1}^{(i+1)} - \mathbf{q}_{k-1}^{(i)}) + (\Delta \mathbf{q}_{k-1}^{(i+1)} - \Delta \mathbf{q}_{k-1}^{(i)}) = \left[I - \frac{\alpha B(\mathbf{x}_{k-1}) \mathbf{S}_{k-1} \mathbf{S}_{k-1}^T B(\mathbf{x}_{k-1})}{\beta + f_{k-1}} \right] D \mathbf{q}_{k-1}^{(i)}. \quad (3)$$

令 $E_k = I - \frac{\alpha}{\beta + f_k} B(\mathbf{x}_k) \mathbf{S}_k \mathbf{S}_k^T B(\mathbf{x}_k)$, 式(3)可写为

$$D \mathbf{q}_k^{(i)} = E_{k-1} D \mathbf{q}_{k-1}^{(i)} = (E_{k-1} E_{k-2} \dots E_1) D \mathbf{q}_1^{(i)} = (E_{k-1} E_{k-2} \dots E_1 E_{N_s} E_{N_s-1} \dots E_k) D \mathbf{q}_k^{(i-1)} =$$

$$(E_{k-1}E_{k-2}\cdots E_1E_{N_s}E_{N_s-1}\cdots E_k)^i Dq_k^{(0)} = R_k^i Dq_k^{(0)},$$

其中 $R_k = E_{k-1}E_{k-2}\cdots E_1E_{N_s}E_{N_s-1}\cdots E_k$. 由式(1),(2)可知

$$Dq_k^{(0)} = q_k^{(1)} - q_k^{(0)} = \Delta q_{k-1}^{(1)} + \Delta q_{k-2}^{(1)} + \cdots + \Delta q_1^{(1)} + \Delta q_{N_s}^{(0)} + q_{N_s-1}^{(0)} + \cdots + \Delta q_k^{(0)} = \sum_{j=1}^{k-1} \frac{\alpha e_j^{(1)} B(x_j) S_j}{\beta + f_j} + \sum_{j=k}^{N_s} \frac{\alpha e_j^{(0)} B(x_j) S_j}{\beta + f_j}.$$

那么

$$Dq_k^{(i)} = R_k^i \left(\sum_{j=k}^{N_s} \frac{\alpha e_j^{(0)} B(x_j) S_j}{\beta + f_j} + \sum_{j=1}^{k-1} \frac{\alpha e_j^{(1)} B(x_j) S_j}{\beta + f_j} \right), \tag{4}$$

E_k 为对称阵,表达式为

$$E_k = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & & \vdots & & & & \vdots & & & & & \vdots & & & \vdots \\ \vdots & & 1 & 0 & & & & 0 & & & & & 0 & & & \vdots \\ 0 & \cdots & 0 & e_{pp} & 0 & \cdots & 0 & e_{pq} & 0 & \cdots & \cdots & 0 & e_{pr} & 0 & \cdots & 0 \\ \vdots & & & 0 & 1 & & & 0 & & & & & 0 & & & \vdots \\ \vdots & & & \vdots & & \ddots & & \vdots & & & & & \vdots & & & \vdots \\ \vdots & & & 0 & & & 1 & 0 & & & & & 0 & & & \vdots \\ 0 & \cdots & 0 & e_{qp} & 0 & \cdots & 0 & e_{qq} & 0 & \cdots & \cdots & 0 & e_{qr} & 0 & \cdots & 0 \\ \vdots & & & 0 & & & & 0 & 1 & & & & 0 & & & \vdots \\ \vdots & & & \vdots & & & & \vdots & & \ddots & & & \vdots & & & \vdots \\ \vdots & & & \vdots & & & & \vdots & & \ddots & & & \vdots & & & \vdots \\ \vdots & & & 0 & & & & 0 & & & & & 1 & 0 & & \vdots \\ 0 & \cdots & 0 & e_{rp} & 0 & \cdots & 0 & e_{rq} & 0 & \cdots & \cdots & 0 & e_{rr} & 0 & \cdots & 0 \\ \vdots & & & 0 & & & & 0 & & & & & 0 & 1 & & \vdots \\ \vdots & & & \vdots & & & & \vdots & & & & & & & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \cdots & \cdots & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & 0 & \cdots & \cdots & 1 \end{bmatrix}$$

$$e_{ii} = \begin{cases} 1, & s_{k,i} = 0, \\ 1 - \frac{\alpha}{\beta + f_k} b_i^2(x_k), & s_{k,i} = 1 \end{cases}$$

$$e_{ij} = \begin{cases} 0, & s_{k,i} = 0 \text{ or } s_{k,j} = 0, \\ -\frac{\alpha}{\beta + f_k} b_i(x_k) b_j(x_k), & s_{k,i} = 1 \text{ and } s_{k,j} = 1, i \neq j. \end{cases}$$

为便于论述,引入算子 $\text{row}^i(A)$,表示取矩阵 A 的第 i 个行向量. 有下面两个引理:

引理1. 对任一矩阵 $A = \{a_{ij}\}$, $b_i = b_i(x_k)$ 为基函数. 当 $0 < \alpha < 2, \beta > 0$ 时, $\forall k \in \{1, 2, \dots, N_s\}$ 当且仅当 $(\sum_{l \in U_k} a_{il} b_l)^2 = 0$ 时, $\|\text{row}^i(AE_k)\| = \|\text{row}^i(A)\|$ 成立.

引理2. 当 $0 < \alpha < 2, \beta > 0$ 时, $\forall i \in \{1, 2, \dots, N_s\}$, $\lim_{j \rightarrow +\infty} R_k^j E_i = \lim_{j \rightarrow +\infty} R_k^j$.

限于篇幅,上述引理的证明从略. 由引理可得下述定理.

定理. 当 $0 < \alpha < 2, \beta > 0$ 时, $\lim_{j \rightarrow +\infty} Dq_k^{(j)} = 0$.

证明. 记 $\lim_{j \rightarrow +\infty} r_k^T(j) = \lim_{j \rightarrow +\infty} \text{row}^i(R_k^j)$, 由引理1和引理2, 当 $j \rightarrow +\infty$ 时, 有

$$\left(\sum_{l \in U_n} r_{il} b_l(\mathbf{x}_n) \right)^2 = 0, \quad n = 1, 2, \dots, N_s.$$

$$\lim_{j \rightarrow +\infty} R_k^i B(\mathbf{x}_n) \mathbf{S}_n = \left[\lim_{j \rightarrow +\infty} \sum_{l \in U_n} r_{1l}(j) b_l(\mathbf{x}_n), \lim_{j \rightarrow +\infty} \sum_{l \in U_n} r_{2l}(j) b_l(\mathbf{x}_n), \dots, \right. \\ \left. \lim_{j \rightarrow +\infty} \sum_{l \in U_n} r_{Ll}(j) b_l(\mathbf{x}_n) \right]^T = 0 \quad (5)$$

由式(4),(5)得

$$\lim_{i \rightarrow +\infty} D\mathbf{q}_k^{(i)} = \lim_{i \rightarrow +\infty} \left[\sum_{j=1}^{k-1} \frac{\alpha e_j^{(1)} R_k^i B(\mathbf{x}_j) \mathbf{S}_j}{\beta + f_j} + \sum_{j=k}^{N_s} \frac{\alpha e_j^{(0)} R_k^i B(\mathbf{x}_j) \mathbf{S}_j}{\beta + f_j} \right] = 0.$$

上述定理表明改进的学习算法是收敛的.

3 仿真实例

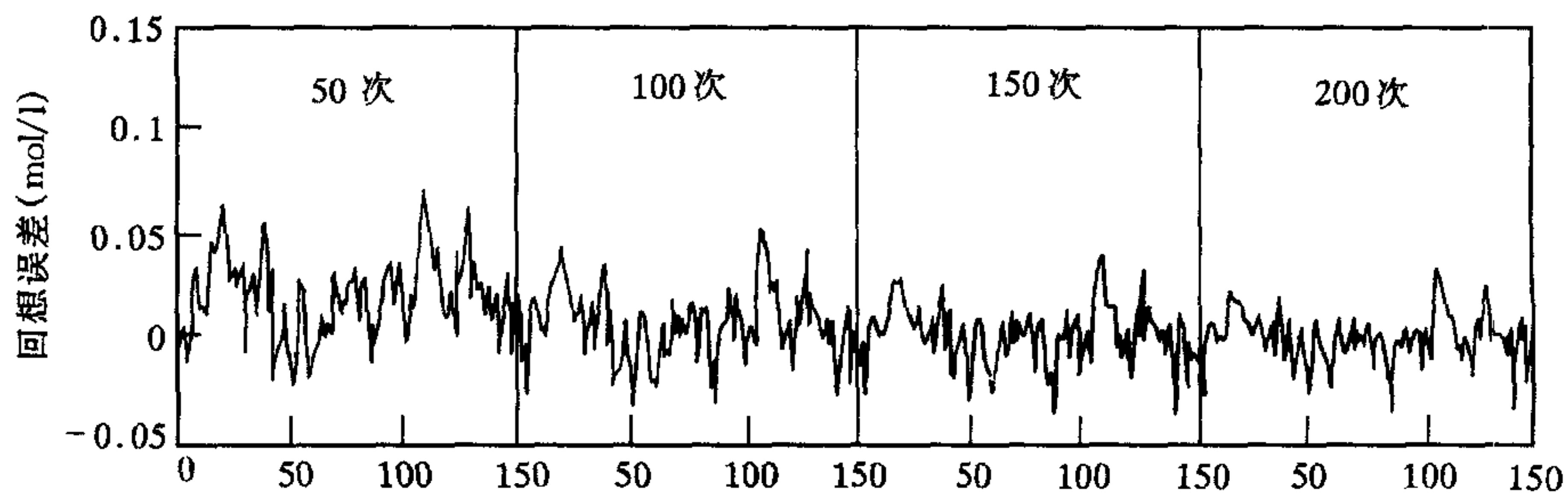
基函数采用高斯基函数, $b_i(\cdot)$ 定义在第 i 个超立方体 C_i 上, 对 $\forall \mathbf{x} \in C_i, b_i(\mathbf{x}) = \exp\left(-\sum_{j=1}^m (x_j - \bar{x}_{ij})^2 / \sigma_{ij}^2\right)$, 其中 \bar{x}_{ij} 为均值, σ_{ij} 为方差. CSTR 系统的模型是两个非线性微分方程^[5]

$$\dot{C}_A = \frac{q}{V} (C_{Af} - C_A) - k_0 C_A \exp\left(-\frac{E}{RT}\right),$$

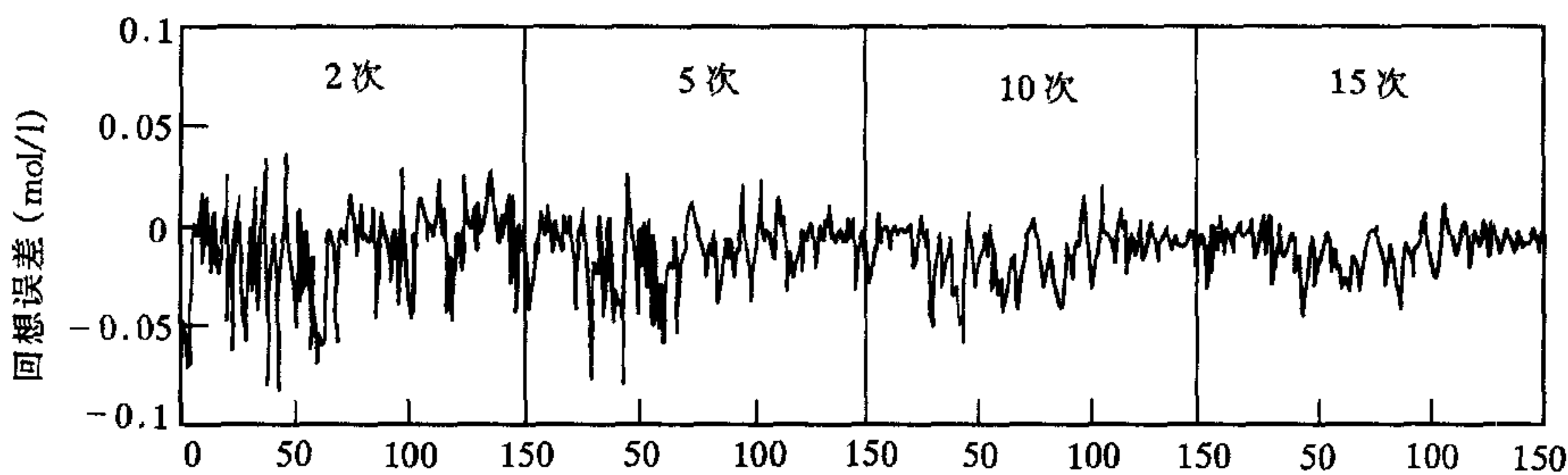
$$\dot{T} = \frac{q}{V} (T_f - T) + \frac{(-\Delta H)k_0 C_A \exp\left(-\frac{E}{RT}\right)}{\rho C_p} + \frac{\rho_c C_{pc}}{\rho C_p V} q_c \left[1 - \exp\left(-\frac{hA}{q_c \rho_c C_{pc}}\right)\right] (T_{cf} - T),$$

式中 C_A 是 A 组分流出时的浓度, T 是反应器的温度, q 是进料流速, q_c 是冷却剂流速. 模型中剩余参数的含义及其在正常工作条件下的数值见文献[5]. 控制量为 q_c 输出为 C_A . 设计 CMAC 的输入为 $\mathbf{x}_k = [C_A(k-1), C_A(k-2), C_A(k-3), q_c(k-1), q_c(k-2), q_c(k-3)]$. 选择 CMAC 输入空间为 $U = [0.04, 0.14] \times [0.04, 0.14] \times [0.04, 0.14] \times [94, 114] \times [94, 114] \times [94, 114]$. 用伪随机信号激励 CSTR, 通过采样(周期为 0.1min)获得一批输入输出数据 $\{\mathbf{x}_k, C_A(k)\}$, 从中选择 300 个, 平分为两组: 样本 A 和样本 B. 分别采用 C-L 算法、标准的 Albus 算法和改进的 C-L 算法, 批量学习样本 A. C-L 算法中 α 取 1.2; 改进的 C-L 算法中 α 取 1.2, β 取 1. 图 1(a)、(b)、(c) 分别用三种不同学习算法学习样本 A 时, 在不同学习阶段的回想(recall)误差 $e_k = C_A(k) - \text{CMAC}(\mathbf{x}_k)$. 图 1(d) 为改进的 C-L 算法学习样本 B 时在不同学习阶段的回想误差. 比较图(a)、(b)、(c) 可以看出, 改进的 C-L 算法学习速度明显快于 C-L 算法, 同时, 也比标准的 Albus 算法快, 而且批量学习效果很好, 优于原算法. 由图(c)和(d)可以看出, 学习算法参数固定, 学习不同的样本 A 和 B, 都能很快收敛, 即收敛性与学习样本无关. 改进的 C-L 算法计算较复杂, 但软件实现 CMAC 时, 增加的计算量不会给实际应用带来麻烦.

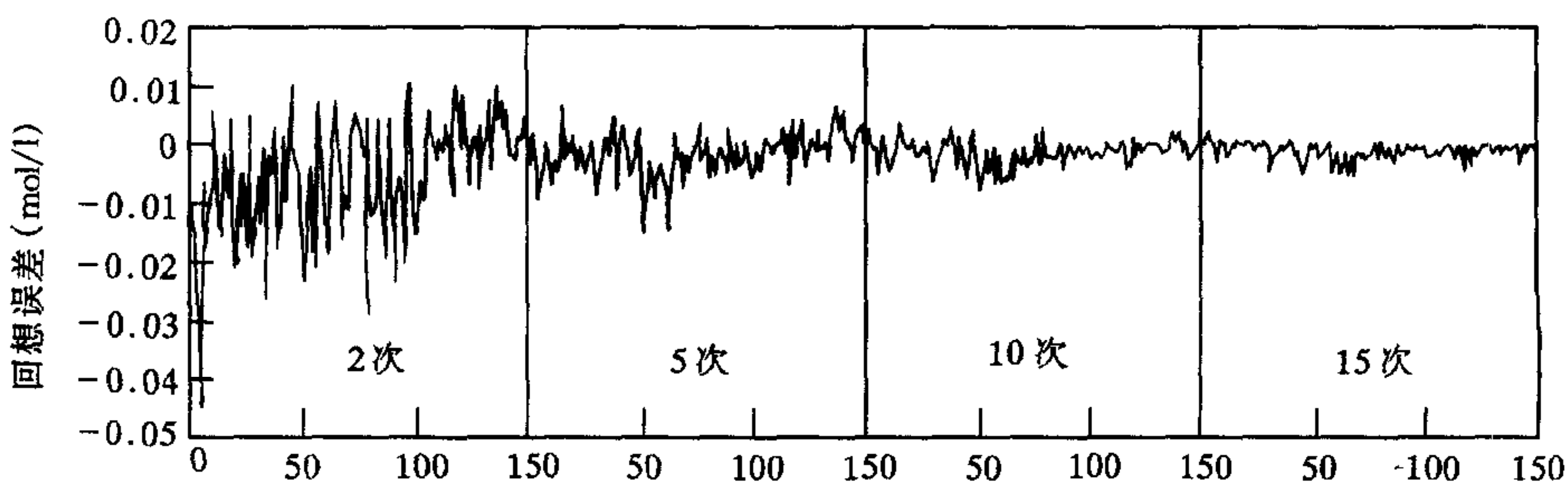
本文指出了基于广义基函数的 C-L 算法收敛条件在实际应用时的缺陷, 并提出了改进算法, 使收敛条件不依赖于基函数和样本数据, 且仿真实例说明改进算法比 C-L 算法学习速度快, 具有广泛的应用前景.



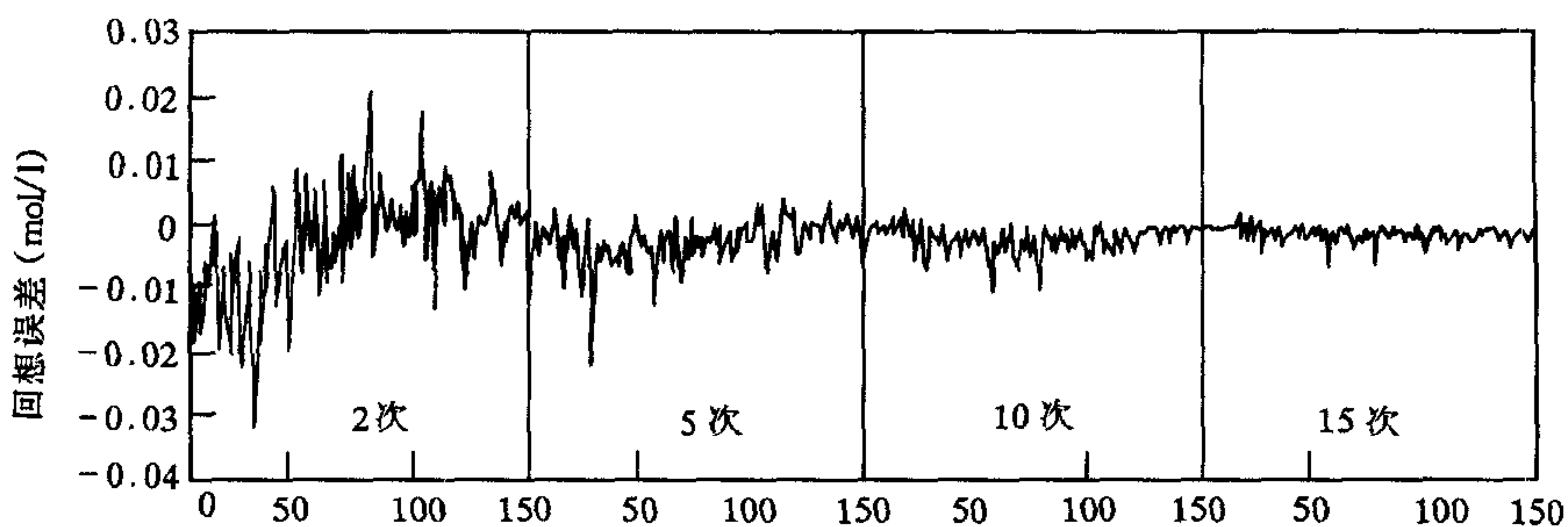
(a) C-L 算法, 学习样本 A



(b) 标准 Albus 算法, 学习样本 A



(c) 改进的 C-L 算法, 学习样本 A



(d) 改进的 C-L 算法, 学习样本 B

图1 学习样本, 批量学习算法、学习次数与回想误差的关系

参 考 文 献

- 1 Albus J S. Data storage in the cerebellar model articulation controller (CMAC). *Journal of Dynamic Systems, Measurement, and Control, Transaction of ASME*, 1975, **97**(3):228—233
- 2 欧阳楷等. 神经计算中坐标变换(CMAC)的泛化特性. *自动化学报*, 1997, **23**(4):475—481
- 3 Chiang C T, Lin C S. CMAC with general basis functions. *Neural Networks*, 1996, **9**(7):1199—1211

- 4 吴宏鑫. 全系数自适应控制理论与应用. 北京:国防工业出版社,1990
- 5 Nahash *et al.* Nonlinear internal model control strategy for neural network models. *Computer Chm. Engng.*, 1992, 16(12):1039—1057

段培永 1968年生,1991年毕业于山东建工学院. 现为上海交通大学自动化系博士生. 主要从事人工神经网络、模糊控制、自适应控制等理论的研究及应用.

邵惠鹤 见本刊 Vol. 22, No. 2, P. 213.

(上接241页)

中国自动化学会1999年一般专题学术活动计划

流程工业过程优化控制专题研讨会	专题研讨	7月	80	安徽	孙德敏 合肥中国科技大学自动化系 230026
全国第3届DCS、PLC和现场总线学术交流会	专题研讨	7月	100	太原	张忠怀 山西太原理工大学自动化系 邮编 030024
第2届板形测量与控制学术研讨会	交流近年来板材的板型测量与控制成果,研讨如何尽快提高钢板、带材的质量	10月	100	北京	杨溪林 北京919信箱 邮编 100071
医药工业生产过程自动化技术专题研讨会	专题研讨	4季度	100	待定	吴斌昌 上海市漕宝路103号 邮编 200233
工程控制网和网络组套研讨会	国际上工控网络和网络组套最新动态及应用、并进行研讨	4季度	50	哈尔滨	王炎 哈尔滨工业大学机器人研究所 150006

(待续)