

# 基于连续映射小脑模型的广义预测控制快速算法

谢永斌 罗 忠 冯祖仁 胡保生

(西安交通大学系统工程研究所·西安, 710049)

**摘要:** 文中把改进后的小脑模型(CMAC)应用于广义预测控制(GPC)算法中, 避免了该算法需实时进行的复杂计算. 本文的方法对 GPC 在实际中的应用有重要的意义.

**关键词:** 小脑模型; 状态方程; 神经网络; 广义预测控制

## 1 引 言

由 Clarke<sup>[1,2]</sup>提出的广义预测控制 GPC(Generalized Predictive Control)算法由于其具有良好的控制性能而获得了众多研究者的关注. 但 GPC 算法实时计算量较大, 这在应用中受到一定限制. Albus<sup>[3]</sup>提出的小脑模型结构 CMAC(Cerebellar Model Articulation Control)可认为是一个具有三层结构的神经元网络. CMAC 结构简单, 可完成任意多输入多输出非线性函数的映射, 学习速度快, 便于在实际中应用.

本文首先以状态方程的形式, 推证了 GPC 算法的一种新的计算方法, 并利用文中给出的连续映射 CMAC 离线完成 GPC 算法的绝大部分计算任务, 从而极大地提高了 GPC 算法的在线计算速度.

## 2 GPC 算法的一种状态方程实现

设系统的模型为

$$A'(z^{-1})y(t) = B'(z^{-1})U(t-1) + \xi(t)/\Delta. \quad (2.1)$$

其中  $A'(z^{-1}), B'(z^{-1})$  为后移算子  $z^{-1}$  的实系数多项式,  $\xi(t)$  为零均值白噪声,  $\Delta = 1 - z^{-1}$ . 若  $A(z^{-1}) = A'(z^{-1})\Delta, B(z^{-1}) = z^{-1}B'(z^{-1})$  则(2.1)式也可写成

$$A(z^{-1})y(t) = B(z^{-1})\Delta U(t) + \xi(t). \quad (2.2)$$

其中  $A(z^{-1}) = 1 + a_1z^{-1} + \dots + a_{na}z^{-na}, B(z^{-1}) = b_1z^{-1} + \dots + b_{nb}z^{-nb}$ .

(2.2)式可写成如下的等价状态方程形式

$$X(k+1) = AX(k) + b\Delta U(k), \quad (2.3)$$

$$y(k) = C^T X(k) + \xi(k). \quad (2.4)$$

其中  $b^T = [b_1, 0, \dots, 0, 1, 0, \dots, 0]_{1 \times (na+nb-1)}, C^T = [1, 0, \dots, 0]_{1 \times (na+nb-1)},$

$X(k) = [y(k), \dots, y(k-na+1), \Delta U(k-1), \dots, \Delta U(k-nb+1)]_{1 \times (na+nb-1)},$

$$A = \begin{bmatrix} a_1 & a_2 & \dots & a_{na-1} & a_{na} & b_2 & b_3 & \dots & b_{nb-1} & b_{nb} \\ 1 & 0 & \dots & 0 & 0 & & & & & \\ & \ddots & \ddots & \vdots & \vdots & & 0 & & & \\ & & 0 & \ddots & \ddots & & & & & \\ & & & & 1 & 0 & & & & \\ \hline & & & & & 0 & 0 & \dots & 0 & 0 \\ & & & & & 1 & 0 & \dots & 0 & 0 \\ & & & & & & \ddots & \ddots & \vdots & \vdots \\ & & & & & & 0 & \ddots & \ddots & \vdots \\ & & & & & & & & 1 & 0 \end{bmatrix}.$$

则(2.3),(2.4)式系统在无干扰,初始状态为零的单位阶跃响应系数为

$$t_i = C^T A^{i-1} b, \quad i = 1, 2, \dots, \quad (2.5)$$

若在第  $k$  个时刻,未来控制输入增量为零,即  $\Delta U(k) = \Delta U(k+1) = \dots = 0$ , 则系统未来输出的最优预测为

$$\hat{y}(k+m|k) = C^T A^m X(k), \quad m = 1, 2, \dots. \quad (2.6)$$

在针对(2.1)式的 GPC 算法中,若优化时域的始值为 1,终值为  $P$ ,控制时域为  $M$ ,  $\lambda$  为控制加权系数,定值控制系统的参考轨迹为

$$w(k+i) = \alpha^i y(k) + (1 - \alpha^i) L, \quad i = 1, 2, \dots, P, \quad (2.7)$$

$y(k)$  为系统的实际输出值,  $L$  为设定值,  $\alpha (0 \leq \alpha < 1)$  为设定值跟随系数,则 GPC 的最优控制增量为<sup>[2]</sup>

$$\Delta U(k) = g^T (W - F). \quad (2.8)$$

其中  $g^T$  为矩阵  $(G^T G + \lambda I)^{-1}$  的第一行,而  $G$  阵为系统(2.3),(2.4)的单位阶跃响应系数阵,综合(2.5)式其相应的形式为

$$G = \begin{bmatrix} C^T b & 0 & \dots & \dots & 0 \\ C^T A b & C^T b & 0 & \dots & 0 \\ C^T A^2 b & C^T A b & C^T b & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ C^T A^{P-1} b & \dots & \dots & \dots & C^T A^{P-m} b \end{bmatrix}_{P \times M}, \quad (2.9)$$

$$(2.8) \text{ 式中} \quad W = [w(k+1), \dots, w(k+P)]_{1 \times P}^T, \quad (2.10)$$

$$F = [\hat{y}(k+1|k), \dots, \hat{y}(k+P|k)]_{1 \times P}^T. \quad (2.11)$$

综合(2.6),(2.7),(2.10),(2.11)有

$$W - F = T_1 X(k) + T_2 L. \quad (2.12)$$

其中

$$T_1 = \begin{bmatrix} \alpha & 0 & \dots & 0 \\ \alpha^2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha^P & 0 & \dots & 0 \end{bmatrix} - \begin{bmatrix} C^T A \\ C^T A^2 \\ \vdots \\ C^T A^P \end{bmatrix}, \quad (2.13)$$

$$T_2 = [1 - \alpha, 1 - \alpha^2, \dots, 1 - \alpha^P]_{1 \times P}^T, \quad (2.14)$$

则(2.8)式也可写成

$$\Delta U(k) = d^T X(k) + r_{na+nb} L. \quad (2.15)$$

其中

$$d^T = [r_1, r_2, \dots, r_{na+nb-1}] = g^T T_1, \quad (2.16)$$

$$r_{na+nb} = g^T T_2, \quad (2.17)$$

(2.15) 式可进一步写成

$$\Delta U(k) = \gamma X^0(k). \quad (2.18)$$

其中

$$\begin{aligned} X^0(k) &= \begin{bmatrix} X(k) \\ L \end{bmatrix} \\ &= [y(k), \dots, y(k-na+1), \Delta U(k-1), \dots, \Delta U(k-nb+1), L]^T, \\ \gamma &= [r_1, r_2, \dots, r_{na+nb}]. \end{aligned} \quad (2.19)$$

### 3 连续映射的 CMAC

为了说明其主要思想,以一维输入输出映射为例,连续映射 CMAC 的结构如图 1.

设输入变量  $X \in [a, b] \subset S$ ,  $S$  为输入空间,  $a, b \in \mathbb{R}$ , 把  $[a, b]$  等间隔离散化为  $n$  个值  $S_1, S_2, \dots, S_n$  ( $S_1 = a, S_n = b, S_{i+1} > S_i, i = 1, 2, \dots, n-1$ ), 并将其作为学习点集. 又设任一学习点集中的输入值  $S_i$  可激活单元的个数为  $K$ , 且其地址为  $a_i, a_{i+1}, \dots, a_{i+K-1}$ . 在学习点处, 若输入点为  $S_i$ , 则映射输出

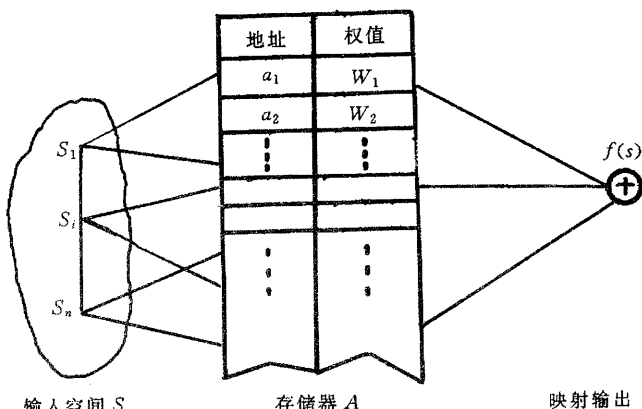


图 1 CMAC 结构

$$f(S_i) = \sum_{j=0}^{K-1} W_{i+j}. \quad (3.1)$$

其中  $W_i$  为地址为  $a_i$  单元存储的权值, 在非学习点处, 其映射输出(泛化输出)由下列内插公式给出, 若任意输入点  $S' \in [S_i, S_{i+1}]$ , 则泛化输出为

$$f(S') = \frac{S_{i+1} - S'}{S_{i+1} - S_i} W_i + \sum_{j=1}^{K-1} W_{i+j} + \frac{S' - S_i}{S_{i+1} - S_i} W_{i+K}. \quad (3.2)$$

连续映射 CMAC 的学习方法与 CMAC 完全一样, 若学习输入值为  $S_i$ , 其希望的映射输出值为  $f_i$ , 则将  $\Delta = (f_i - f(S_i))/K$  分别加入  $S_i$  可激活的  $K$  个单元,  $f(S_i)$  利用以前学习的权值由 (3.1) 式求得. 这个过程一直到在所有的学习点处映射输出误差  $K\Delta$  达到希望的精度为止. 具有 Hash 编码的 CMAC 其连续映射原理与上述方法类似.

### 4 GPC 快速算法的实现步骤

对于 (2.1) 式所表示的系统, 首先选取合适的 GPC 优化时域  $P$ , 控制时域  $M$ , 控制加权系数  $\lambda$  及设定值跟随系数  $\alpha$ .

第一步 离线构造并训练连续映射 CMAC.

1) 系统所有未知或不定参数的变化范围作为 CMAC 的输入空间, 未知或不定参数的总数即为 CMAC 的输入维数. 在所有输入维数上分别进行等间隔离散化构成学习输入点集.

2) (2.18) 式中  $\gamma$  向量作为 CMAC 的  $na + nb$  维输出空间. 选择合适的每一学习点可激活的存储单元个数  $K$ , 从而构造出相应的 CMAC 结构.

3) 在 CMAC 训练时, 希望输出值的计算按以下顺序: 给出一个具体的学习输入点(即某一确定的系统参数点), 由 (2.5) 式求取  $t_i (i = 1, 2, \dots, P)$ , 由 (2.9) 式求取  $g^T$ , 由 (2.13), (2.14) 求取  $T_1, T_2$ , 由 (2.16), (2.17) 求取向量  $d^T$  及标量  $r_{na+nb}$ , 从而获得输出希望向量  $\gamma$ .

4) 训练所有输入空间的等间隔离散学习点, 直到所有学习点处收敛到给定的精度为止.

第二步 实时控制.

1) 在第  $k$  个时刻, 据最新数据, 用递推公式估计系统未知或时变参数  $\hat{\Theta}(k)$ .

2) 以  $\hat{\Theta}(k)$  作为训练好的 CMAC 的输入, 由 (3.1) 或 (3.2) 式得到输出向量  $\gamma$ .

3) 用 (2.18) 式计算控制输出增量  $\Delta U(k)$ , 并实施控制.

4) 整理有关数据结构, 返 1) 进行  $k+1$  时刻的计算与控制.

5 仿真研究

仿真模型为  $G(z^{-1}) = (0.5z^{-1} + bz^{-2})/(1 - az^{-1})$ ，则进一步可写为

$$y(k) = (1 + a)y(k - 1) - ay(k - 2) + 0.5\Delta u(k - 1) + b\Delta u(k - 2).$$

其中  $a \in [0, 0.9]$ ,  $b \in [0, 0.5]$ .

在 GPC 算法中取  $\lambda = 0, P = 6, M = 2, a = 0.3$ . 把  $a, b$  在其取值范围内分别等间隔离散成 25 个值, 构成 625 个学习输入点, 每个学习点激活 5 个存储单元, 则相应的二输入  $(a, b)$ , 四输出  $(r_1, r_2, r_3, r_4)$  的连续映射 CMAC 训练结果如图 2, 若  $L$  为设定值, 则实时的控制计算仅为

$$\Delta U(k) = r_1y(k) + r_2y(k - 1) + r_3\Delta u(k - 1) + r_4L.$$

在仿真中, 系统的实际参数为  $a = 0.8, b = 0.4$ , 参数递推估计的初值为  $a = 0.4, b = 0$ , 如图 3 为 200 个时刻的实时控制相应情况.

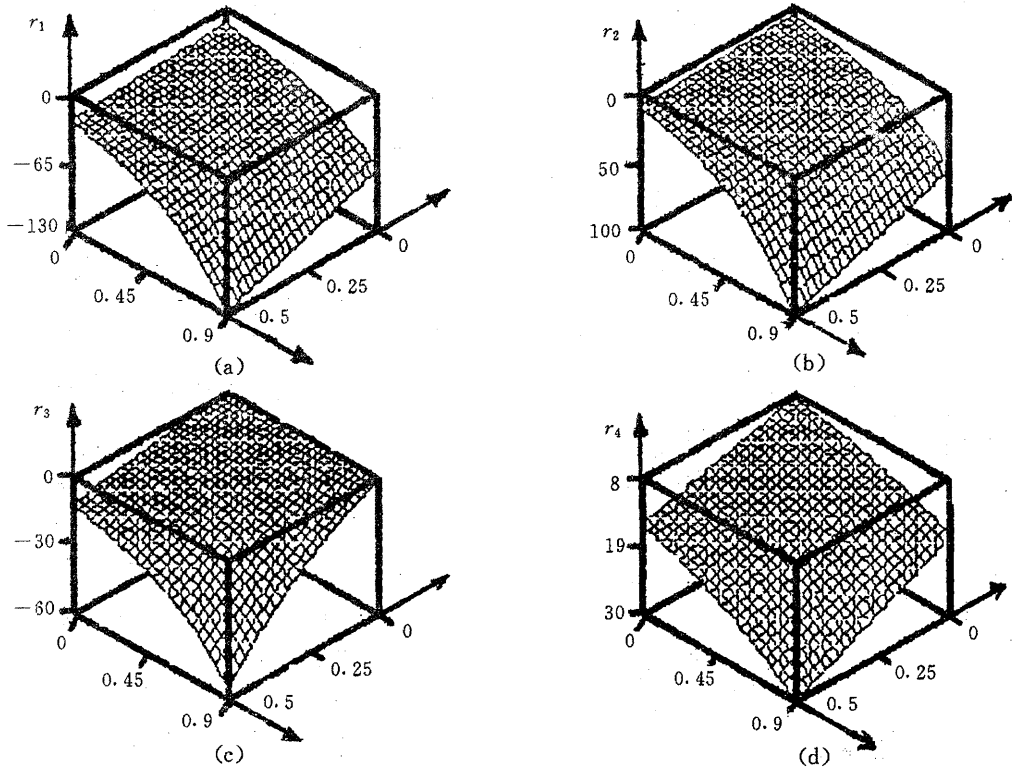
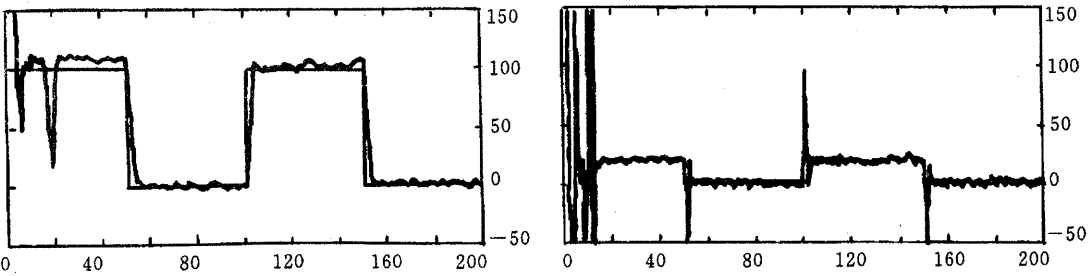


图 2 CMAC 训练结果



(a) 设定值与控制响应曲线 (b) 控制输出量曲线

图 3 GPC 控制响应等曲线

## 6 结论及推广

本文利用连续映射 CMAC 有效地解决了 GPC 在线计算量大的缺点,对 GPC 算法的实际应用有重要意义。

本文的方法立即可推广到如下的情况:GPC 算法中优化时域值不为 1 的情况或 GPC 的优化指标中加入误差权矩阵  $Q$  及控制权矩阵  $R$  的情况.在 GPC 算法中加入类似动态矩阵控制<sup>[1]</sup>的实时在线反馈的情况,非定值控制的情况,以及多输入多输出的情况等。

## 参 考 文 献

- 1 席裕庚. 预测控制. 北京:国防工业出版社,1993
- 2 Clark, D. W., Mohtadi, C. and Tuffs, P. S.. Generalized predictive control part I the basic algorithm. *Automatica*, 1987,23(2):137-148
- 3 Albus, J. S.. A new approach to manipulator control: the cerebellar model articulation controller (CMAC). *Jour. of Dynamic Sys., Measurement and Control*, Sept. 1975,220-233

## A Fast Algorithm for Generalized Predictive Control Based on the CMAC

XIE Yongbin, LUO Zhong, FENG Zuren and HU Baosheng

(Institute of Systems Engineering, Xi'an Jiaotong University • Xi'an, 710049, PRC)

**Abstract:** In this article, the CMAC is applied to the generalized predictive control algorithm for the purpose of saving complex computation. Our method has high practical significance to the applications of the GPC.

**Key words:** cerebellar model articulation control; state equations; neural network; generalized predictive control

## 本文作者简介

谢永斌 见本刊 1997 年第 3 期第 402 页.

罗 忠 见本刊 1997 年第 3 期第 402 页.

冯祖仁 见本刊 1997 年第 3 期第 402 页.

胡保生 见本刊 1997 年第 3 期第 402 页.