第22卷 第1期
1996年1月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 22, No.1
Jan., 1996

短 文

广义预测控制的并行算法[1)]

王 轶 席裕庚

(上海交通大学自动化研究所 200030)

摘 要

研究广义预测控制(GPC)的并行算法. 常规对 GPC 的设计, 面临的在线矩阵求逆的耗时和控制灵活性的矛盾很难较好的解决. 本文通过在线并行实现, 有效地提高了算法的实时性和对复杂对象的适应性.

关键词: 广义预测控制、并行算法、在线实现.

1 引言

广义预测控制(GPC)是在自适应控制的研究中发展起来的. 由于它采用长时段优化改善了控制的鲁棒性, 因而受到控制界的广泛重视. 但是, 迄今为止的控制理论及算法, 都建立在 Von Neumann 串行结构计算机基础上, GPC 也不例外. 这种以串行递推为基础的控制算法, 在面对实时控制的大规模计算和信息处理时, 出现了根本性的困难和局限. 并行机的诞生以及并行处理技术的发展, 为发展基于并行处理的复杂系统控制技术提供了可能. 针对 GPC 算法, 引入并行算法, 快速实现了控制器的参数自校正; 通过在线运算的阵列结构, 提高了实时控制的速度, 从而有效提高了算法的实时性及对复杂对象的适应性.

2 广义预测控制(GPC)的并行算法及结构

Clarke 等最早在文[1]中提出 GPC 算法, 为节省篇幅, 本文不再重复[1]中的算法介绍, 而是结合算法的并行化直接进行讨论.

GPC 算法主要包括参数自校正、柔化、调节和预测四部分. 其中后三部分为向量运算, 可直接并行实现. 因此, 下面重点讨论参数自校正部分的并行实现.

2.1 最小二乘估计模型参数

GPC 的预测模型为 CARIMA 模型

$$A(z^{-1})y(t)=B(z^{-1})u(t-1)+\xi(t)/(1-z^{-1}), \qquad (2-1)$$

其中 $\xi(t)$是均值为零的白噪声序列,

$$A(z^{-1})=1+a_1z^{-1}+\cdots a_nz^{-n},\ B(z^{-1})=b_0+b_1z^{-1}+\cdots b_{n_b}z^{-n_b}.$$

1) 国家教委资助回国留学人员和优秀青年教师基金国家自然科学基金资助项目.
本文于 1993 年 7 月 5 日收到

为使模型准确反映实际，须采用最小二乘实时辨识 $A(z^{-1})$、$B(z^{-1})$，参数递推辨识的公式为

$$\begin{cases} \hat{\tilde{\theta}}(k)=\hat{\tilde{\theta}}(k-1)+\tilde{K}(k)[\Delta y(k)-\tilde{\varphi}^{\mathrm{T}}(k)\hat{\tilde{\theta}}(k-1)]\ , \\ \tilde{K}(k)=\tilde{P}(k-1)\tilde{\varphi}(k)[\tilde{\varphi}^{\mathrm{T}}(k)\tilde{P}(k-1)\tilde{\varphi}(k)+\mu]^{-1}\ , \\ \tilde{P}(k)=\mu^{-1}[\tilde{I}-\tilde{K}(k)\tilde{\varphi}^{\mathrm{T}}(k)]\,\tilde{P}(k-1). \end{cases} \tag{2-2}$$

其中

$$\tilde{\theta}(k)=[a_1\ ,\ a_2\ ,\ \cdots,\ a_n\ ,\ b_0\ ,\ b_1\ ,\ \cdots, bn_b]^{\mathrm{T}}=[\theta_1^{(k)},\ \cdots,\ \theta_m^{(k)}],\ m=n+n_b+1,$$

$$\tilde{\varphi}(k)=[-\Delta y(k-1),\cdots-\Delta y(k-n),\Delta u(k-1),\cdots\Delta u(k-n_b-1)]^{\mathrm{T}},$$

$$\tilde{P}(k)=(P_{ij}^{(k)})_{m\times m},\ \tilde{K}(k)=[K_1\ ,\ K_2\ ,\cdots,K_m]^{\mathrm{T}}\ ,$$

$0<\mu<1$ 为遗忘因子，常取 $0.95<\mu<1$。

初值 $\hat{\tilde{\theta}}(0)=\tilde{0},\ \tilde{P}(0)=\alpha^2\tilde{I}$ (α 为充分大的正数 $(0,1)$)

令

$$\tilde{P}\tilde{\varphi}=X=[x_1\ ,x_2\ ,\cdots,x_m]^{\mathrm{T}},\ \tilde{\varphi}^{\mathrm{T}}\tilde{P}=Z=[z_1\ ,z_2\ ,\cdots,z_m]^{\mathrm{T}},$$

$$\tilde{\varphi}^{\mathrm{T}}\tilde{P}\tilde{\varphi}=s,\ \tilde{\varphi}^{\mathrm{T}}\hat{\tilde{\theta}}(k-1)=t.$$

则

$$x_i=\sum_{j=1}^{m}\varphi_j\cdot P_{ij},\quad z_i=\sum_{j=1}^{m}\varphi_j\cdot P_{ji},\quad (i=1,\cdots,m),$$

$$s=\sum_{i=1}^{m}\varphi_i\cdot x_i,\quad t=\sum_{i=1}^{m}\varphi_i\cdot\hat{\theta}_i^{(k-1)}\ . \tag{2-3}$$

继而推得 $(i=1,\cdots,m;\ j=1,\cdots,m)$

$$K_i=x_i/(s+\mu), \tag{2-4}$$

$$P_{ij}^{(k)}=\mu^{-1}(P_{ij}^{(k-1)}-K_i z_i), \tag{2-5}$$

$$\hat{\theta}_i^{(k)}=\hat{\theta}_i^{(k-1)}+K_i(\Delta y(k)-t). \tag{2-6}$$

式(2－4)和(2－6)对 i 可并行，(2－5)则对 i，j 都可并行．(2－3)计算式具有同样的结构而可采用向量机并行完成．以上各式都基于形如 $d=\sum_{i=1}^{m}x_i\cdot y_i$ 的点积运算，图 1 是实现该种运算的二叉树结构，用 m 台机的耗时为 $O(\log_2 m)$．(图中 $m=8$)

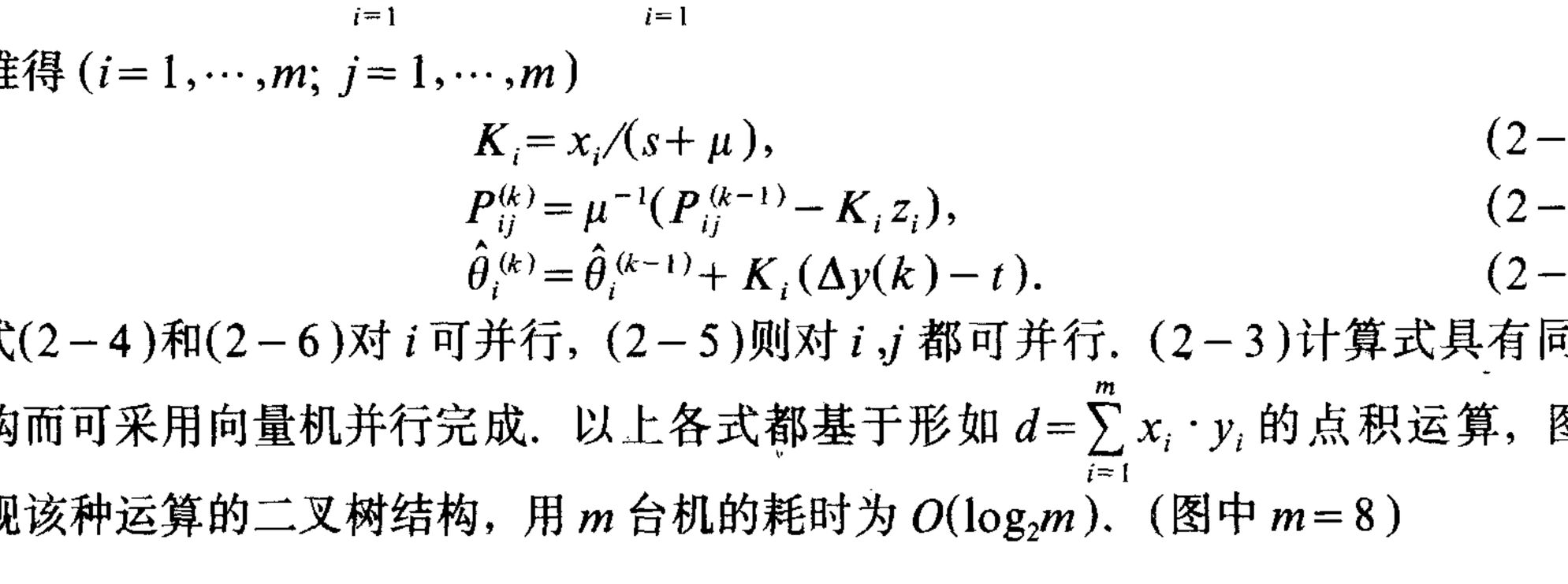

图 1　二叉树结构

2.2 由$A(z^{-1})$递推计算$E_j(z^{-1})$、$F_j(z^{-1})$.

GPC 算法需首先求解以下 Diophantine 方程

$$1=E_j(z^{-1})A(z^{-1})(1-z^{-1})+z^{-j}F_j(z^{-1}), \tag{2-7}$$

其中，$E_j(z^{-1})=e_{j0}+e_{j1}z^{-1}+\cdots+e_{j,j-1}z^{-(j-1)}$，$F_j(z^{-1})=f_{j0}+f_{j1}z^{-1}+\cdots+f_{j,n}z^{-n}$，

$E_j(z^{-1})$、$F_j(z^{-1})$的系数可由下式递推计算

$$\tilde{f}_{j+1}=A\tilde{f}_j,\ E_{j+1}=E_j+f_{j0}z^{-j}, \tag{2-8}$$

其中，$\tilde{f}_j=[f_{j0},\cdots,f_{jn}]^{\mathrm{T}}$，初值$\tilde{f}_0=[1,0,\cdots,0]^{\mathrm{T}}$.

$$\tilde{A}=\begin{bmatrix} 1-a_1 & 1 & 0 & \cdots & 0 \\ a_1-a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n-1}-a_n & 0 & 0 & \cdots & 1 \\ a_n & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

上式递推可进一步写作

$$\begin{cases} e_{j+1,i}=e_{j,i}, & (i=0,\cdots,j-1), \\ e_{j+1,j}=f_{j0} & (j=0,\cdots,N-1), \\ f_{j+1,j}=(a_i-a_{i+1})f_{j,0}+f_{j,i+1}, & (i=0,\cdots,n). \end{cases} \tag{2-9}$$

其中 $a_0=1$，$a_{n+1}=0$，上式可用图 2 的并行结构实现.

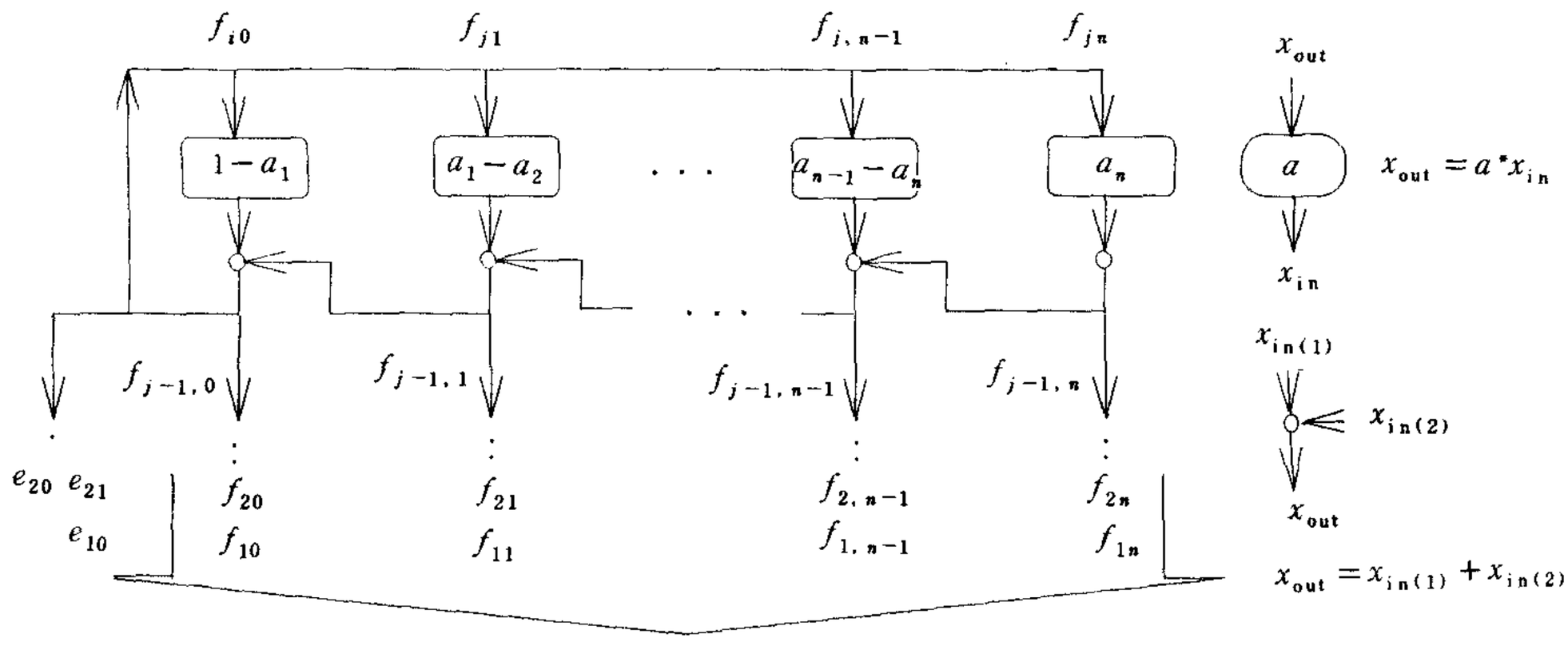

图 2 流水线实现$E_j(z^{-1})$和$F_j(z^{-1})$的递推

2.3 自校正控制器参数 $\tilde{g}^{\mathrm{T}}$、$\tilde{H}$、$\tilde{P}$.

第一步. 由 $E_j(z^{-1})B_j(z^{-1})=G_j(z^{-1})=g_{j0}+g_{j1}z^{-1}+\cdots+g_{j,j+n_b-1}z^{-(j+n_b-1)}$得

$$g_{ji}=\sum_{k=i_1}^{i_2}e_{jk}\cdot b_{i-k},\quad \begin{matrix}(i=0,\cdots,j+n_b-1,\ j=1,\cdots N),\\ i_2=\min(i,j-1),\ i_1=\max(0,i-n_b).\end{matrix} \tag{2-10}$$

利用 $\frac{N}{2}(N+3)$个处理器，经过 $O(N+n_b+1)$后即可完成，见图 3.

第二步. 构造矩阵 $\tilde{G}$、$\tilde{H}$、$\tilde{P}$.

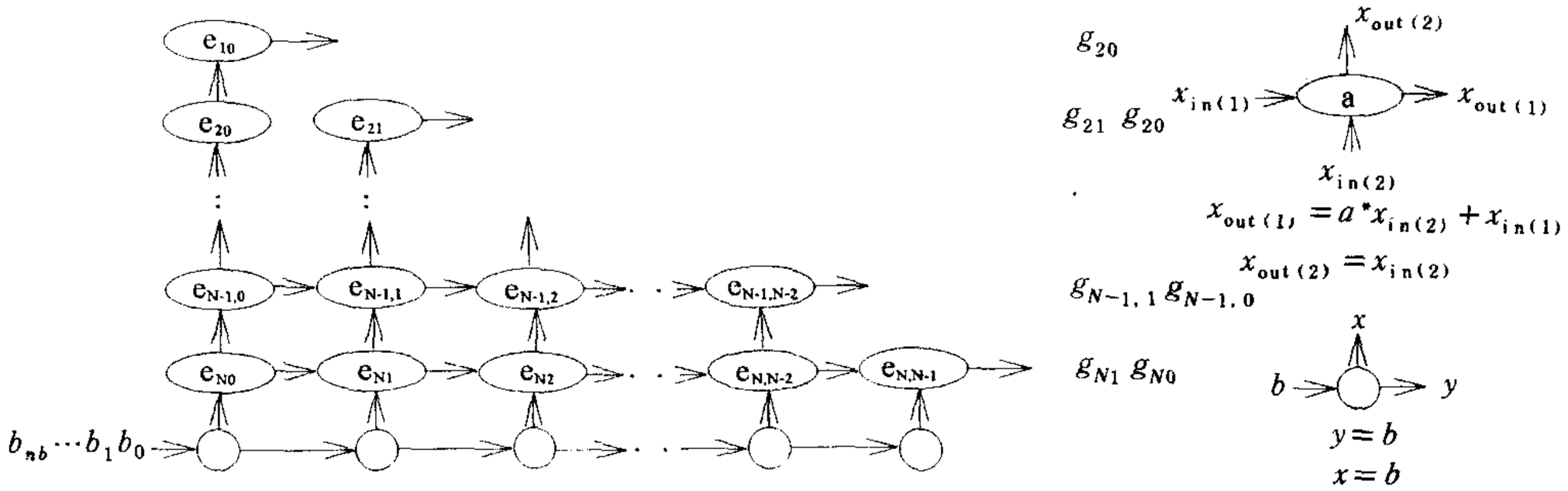

图 3　计算 $\tilde{G}$ 和 $\tilde{g}^{\mathrm{T}}$ 的心动实现

$$\tilde{G}=\begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_{N-NU} \end{bmatrix}_{N\times NU},\quad \tilde{H}=\begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n_b} \\ g_{22} & g_{23} & \cdots & g_{2(n_b+1)} \\ \vdots & \vdots & \ddots & \vdots \\ g_{NN} & g_{N,N+1} & \cdots & g_{N(N+n_b-1)} \end{bmatrix}_{N\times n_b}=\begin{bmatrix} \tilde{h}_1 \\ \tilde{h}_2 \\ \vdots \\ \tilde{h}_N \end{bmatrix}. \tag{2-11}$$

其中

$$g_i = g_{j,i}\quad (i<j;\ j=1,\cdots,N-1),$$

$$\tilde{P}=\begin{bmatrix} f_{10} & f_{11} & \cdots & f_{1n} \\ f_{20} & f_{21} & \cdots & f_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N0} & f_{N1} & \cdots & f_{Nn} \end{bmatrix}_{N\times(n+1)}=\begin{bmatrix} \tilde{P}_1 \\ \tilde{P}_2 \\ \vdots \\ \tilde{P}_N \end{bmatrix}. \tag{2-12}$$

第三步. 求 $\tilde{g}^{\mathrm{T}}=[g_0', g_1', \cdots, g_{N-1}']$.

$$\tilde{g}^{\mathrm{T}}=(1\quad 0\quad \cdots\quad 0)(\tilde{G}^{\mathrm{T}}\tilde{G}+\lambda\tilde{I})^{-1}\tilde{G}^{\mathrm{T}}. \tag{2-13}$$

令

$$\tilde{G}=(\alpha_1,\alpha_2,\cdots,\alpha_{NU})=(\beta_1,\beta_2,\cdots,\beta_N)^{\mathrm{T}},$$

$$\tilde{G}^{\mathrm{T}}\tilde{G}+\lambda\tilde{I}=\tilde{D}=(d_{ij})_{NU\times NU},\ (\tilde{G}^{\mathrm{T}}\tilde{G}+\lambda\tilde{I})^{-1}\tilde{G}^{\mathrm{T}}=\tilde{X}=(\tilde{x}_1,\tilde{x}_2,\cdots,\tilde{x}_N).$$

其中　　$\tilde{x}_i=(x_{1i},x_{2i},\cdots,x_{NU,i})^{\mathrm{T}}$,

则 $\tilde{g}^{\mathrm{T}}$ 的计算可分解为

1) 计算 $\tilde{D}$ 矩阵

$$d_{ij}=\begin{cases}\lambda+\alpha_i^{\mathrm{T}}\cdot\alpha_i & (i=j)\\ \alpha_i^{\mathrm{T}}\cdot\alpha_j & (i\neq j)\end{cases}\quad (i,j=1,\cdots,NU), \tag{2-14}$$

以上 NU^2 组向量的内积 ($O(\log_2 NU)$), 可采用图 1 结构计算.

2) 解方程组求 $\tilde{g}^{\mathrm{T}}$.

由 $\tilde{g}^{\mathrm{T}}=(1\quad 0\quad \cdots\quad 0)\cdot(\tilde{x}_1,\tilde{x}_2,\cdots,\tilde{x}_N)=(x_{11},x_{12},\cdots,x_{1N})$

表明计算 $\tilde{g}^{\mathrm{T}}$ 可转化为求解 $\tilde{D}\tilde{X}=\tilde{G}^{\mathrm{T}}\tilde{x}_i$ 的首元.

首先将 $\tilde{D}$ 阵进行 UL 分解 (这里省去选主元的叙述, 假定这种分解可以进行), 用递推或 Guass 消元并行获得; 继而求解上三角方程 $\tilde{U}\tilde{Y}=\tilde{G}^{\mathrm{T}}$ 得到 $\tilde{Y}$, 取其首行向量即为 $\tilde{g}^{\mathrm{T}}$, 求解采用向量回代法或扫描法[2]. 这样较 LU 分解回代计算量减半.

3 并行GPC算法的性能分析与评价

并行计算时间的减少，以硬件增加为代价．因此对其评价要考虑所用处理器个数这一因素．下面给出性能指标[3]

$$\text{加速比 } S_P = T_1/T_P\text{，效率 } E_P = S_P/P \times 100\%, \tag{3-1}$$

其中，T_P 为并行耗时，T_1 为串行耗时，P 为处理器个数．

对上述并行GPC算法的性能评价可见表1．

表1 并行GPC算法评价

并行GPC算法	处理单元数	计算时间 T_P	串行耗时 T_1	效率
模型辨识 A,B	m^2+2m	$4[\log_2 m]+9$	$m^2+12m-4$	0 — 30%
计算 E,F	$2n+1$	$N+1$	$2Nn$	60% — 1
计算 $\tilde{G}$	$\frac{N}{2}(N+3)$	$N+n_b+1$	$N[(N+1)(2n_b+1)-n_b(n_b+1)]$	4 — 25%
计算 g^{T}	$NU^2(N+2)-NU$	$2NU+[\log_2 N]$	$NU^2(NU+4N)-NU(N-NU)$	16 — 45%
计算 $\Delta u(k)$	$N(n_b+1)$	$2[\log_2 n_b]+[\log_2 N]+4$	$4Nn_b+2N-1$	20 — 30%

该文提出的并行GPC算法，在减少在线计算时间的同时，也消除了原有算法的局限而使控制的灵活性增加，并改善了系统的动态特性．

参 考 文 献

[1] Clarke D W, et al. Generalized Predictive Control. *Automatica*, 1987, **32**(2): 137— 162.

[2] 王嘉谟等. 并行计算方法. 国防工业出版社, 1985.

[3] Maguire L P, et al. Transputer Implementation. *IEE Proceedings*, *Pt. D*, 1991, **138**(4): 355 — 362.

PARALLEL ALGORITHMS IN GENERALIZED PREDICTIVE CONTROL

WANG YI XI YUGENG

(*Institute of Automation, Shanghai Jiao Tong University*, 200030)

ABSTRACT

This paper studies the parallel algorithms in Generalized Predictive Control. Generally, GPE designing faces the contradiction of time-expense in on-line matrix-inversion and control-adaptivity. Based on the existed GPC algorithm, we give some ideas of the on-line parallel algorithms and their realization. As the on-line control were completed the controller's adaptation to the complicated system has also improved.

Key words: GPC, Parallel algorithm, On-line realization.