第21卷 第1期
1995 年 1 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 21, No. 1
Jan., 1995

短 文

基于并行结构实现修正的扩展 Kalman 滤波计算

慕 德 俊　戴 冠 中
(西北工业大学自动控制系　西安　710072)

摘　要

该文基于平方根算法提出了一种新的脉动阵列结构实现修正的扩展 Kalman 滤波计算，使得计算的数值稳定性得到了提高，同现有文献相比，该文使用的算法和结构在计算的实时性和处理器的利用率都得到了较大的提高.

关键词: 并行算法，脉动阵列，滤波计算.

1 引言

Kalman 滤波是一种最优估计算法、在信号处理、目标跟踪、导航系统等方面得到了广泛的应用，可是很多实际系统往往不能用线性方程来描述，而要用更一般的非线性方程描述. 对于非线性系统，最常用的实时滤波算法是扩展的 Kalman 滤波法 (EKF)，Baheti[1] 将其算法映射到一组线性的阵列上来提高其实时性，而这种算法往往得不到满意的滤波效果，为此 Chui[2] 提出了一种修正的扩展 Kalman 滤波法(MEKF)，Lu[3] 给出了实现此算法的并行结构，但是这种结构的硬件需求量大，处理器的利用率很低且迭代周期长. 本文基于平方根算法提出了一种易于并行计算的脉动 (systolic) 结构来实现修正的扩展 Kalman 滤波计算，[3,4]使得硬件要求量减少，处理器的利用率及计算的实时性都得到了提高.

2 MEKF 的平方根算法

对于离散时间的非线性系统中，若可表示成两个子系统，则整个系统可描述如下[2]:

$$\begin{cases} \begin{bmatrix} \boldsymbol{x}(k+1) \\ \boldsymbol{y}(k+1) \end{bmatrix} = \begin{bmatrix} f[\boldsymbol{y}(k)]\boldsymbol{x}(k) \\ h[\boldsymbol{x}(k)], \boldsymbol{y}(k) \end{bmatrix} + \begin{bmatrix} \Gamma_1[\boldsymbol{x}(k), \boldsymbol{y}(k)] & 0 \\ \Gamma_2[\boldsymbol{x}(k), \boldsymbol{y}(k)] & \Gamma_3[\boldsymbol{x}(k), \boldsymbol{y}(k)] \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1(k) \\ \boldsymbol{w}_2(k) \end{bmatrix} \\ \boldsymbol{z}(k) = [c[\boldsymbol{x}(k), \boldsymbol{y}(k)], 0] \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{y}(k) \end{bmatrix} + \boldsymbol{v}(k) \end{cases} \tag{1}$$

本文于 1992 年 12 月 1 日收到

式中 $\boldsymbol{x}$，$\boldsymbol{y}$ 分别为 n、m 维状态矢量，$f(\cdot)$ 为 $n \times n$ 阶矩阵函数，$h(\cdot)$ 为 m 维非线性向量函数，$\Gamma_1(\cdot)$、$\Gamma_2(\cdot)$、$\Gamma_3(\cdot)$ 分别为 $n \times P_1$、$m \times P_1$、$m \times P_2$ 阶非线性的噪声驱动阵，$\boldsymbol{z}$ 为 r 维测量向量，$c(\cdot)$ 为 $r \times n$ 阶非线性向量函数，$\boldsymbol{w}_1, \boldsymbol{w}_2$ 分别为 P_1、P_2 维系统噪声向量，$\boldsymbol{v}$ 为 r 维测量噪声向量且假设系统噪声 $\{w_1(k)\}$、$\{w_2(k)\}$ 与测量噪声 $\{v(k)\}$ 是互不相关的零均值白噪声序列，并有 $Q(k) = \mathrm{Var}\begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}$，$R(k) = \mathrm{Var}[v(k)]$。

MEKF 方法不需要围绕预测估计进行 Taylor 展开，为了改善滤波性能，对 $x(k)$ 的估计可通过对(2)式进行标准 Kalman 滤波来得到。

$$\begin{cases} x(k+1) = f[\bar{y}(k)]x(k) + \Gamma_1[\bar{x}(k), \bar{y}(k)]w_1(k) \\ z(k) = c[\bar{x}(k), \bar{y}(k)]x(k) + v(k) \end{cases} \tag{2}$$

式中 $\bar{x}(k), \bar{y}(k)$ 是通过对非线性模型(1)的滤波计算得到的。上述对模型(1)和(2)的滤波计算可用并行的方法来实现，初值 $\bar{x}(0) = E[x(0)] = \hat{x}(0)$，$\bar{y}(0) = E[y(0)]$，$Q_1(k) = \mathrm{Var}[w_1(k)]$，$R(k) = \mathrm{Var}[v(k)]$。并行计算框图见图 1，若采用平方根算法将具有更好的滤波稳定性。

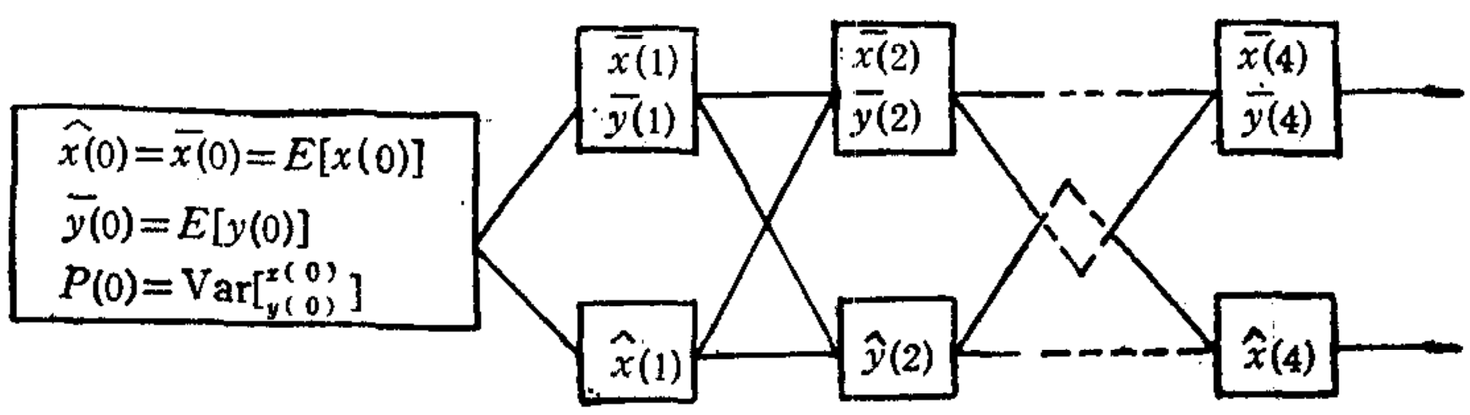


图 1　并行计算框图

定义　$Q(k) = Q^{\frac{1}{2}}(k)Q^{\frac{T}{2}}(k)$，　$Q_1(k) = Q_1^{\frac{1}{2}}(k)Q_1^{\frac{T}{2}}(k)$，

$R(k) = R^{\frac{1}{2}}(k)R^{\frac{T}{2}}(k)$，　$P(k) = P^{\frac{1}{2}}(k)P^{\frac{T}{2}}(k)$，

其中　$Q^{\frac{T}{2}}(k)$，$Q_1^{\frac{T}{2}}(k)$，$R^{\frac{T}{2}}(k)$，$P^{\frac{T}{2}}(k)$ 为上三角阵。对于模型(1)的平方根算法可表示如下：

$$T(k)\begin{bmatrix} P^{\frac{T}{2}}(k-1)F^T(\cdot)H^T(\cdot) & P^{\frac{T}{2}}(k-1)F^T(\cdot) \\ Q^{\frac{T}{2}}(k-1)\Gamma^T(\cdot)H^T(\cdot) & Q^{\frac{T}{2}}(k-1)\Gamma^T(\cdot) \\ R^{\frac{T}{2}}(k) & 0 \end{bmatrix}$$

$$= \begin{bmatrix} v_e^{\frac{T}{2}}(\cdot) & v_e^{-\frac{1}{2}}(\cdot)H(\cdot)P(k|k-1) \\ 0 & P^{\frac{T}{2}}(k) \\ 0 & 0 \end{bmatrix} \tag{3}$$

$$[\bar{x}^T(k), \bar{y}^T(k)] = [\hat{x}^T(k-1)f^T(\cdot), h^T(\cdot)] + [z(k) - c(\cdot)f(\cdot)\hat{x}(k-1)]^T[v_e^{\frac{T}{2}}(\cdot)]^{-1}[v_e^{-\frac{1}{2}}(\cdot)H(\cdot)P(k|k-1)] \tag{4}$$

式中 $T(k)$ 是使初始阵三角化的正交矩阵，

$$F(\cdot)=\frac{\partial}{\partial\begin{bmatrix}x(k-1)\\y(k-1)\end{bmatrix}}\begin{bmatrix}f[y(k-1)]x(k-1)\\h[x(k-1),y(k-1)]\end{bmatrix}\Bigg|_{\substack{x(k-1)=\hat{x}(k-1)\\y(k-1)=\bar{y}(k-1)}}$$

$$H(\cdot)=\frac{\partial}{\partial\begin{bmatrix}x(k)\\y(k)\end{bmatrix}}[c[x(k),y(k)]x(k)]\Bigg|_{\substack{x(k)=\bar{x}(k|k-1)\\y(k)=\bar{y}(k|k-1)}}$$

$$\Gamma(\cdot)=\begin{bmatrix}\Gamma_1[\bar{x}(k-1),\bar{y}(k-1)] & 0\\ \Gamma_2[\bar{x}(k-1),\bar{y}(k-1)] & \Gamma_3[\bar{x}(k-1),\bar{y}(k-1)]\end{bmatrix},$$

$$c(\cdot)=c[\bar{y}(k|k-1),\bar{x}(k|k-1)]$$

$$\bar{x}(k|k-1)=f[\bar{y}(k-1)]\hat{x}(k-1),\quad \bar{y}(k|k-1)=h[\bar{y}(k-1),\hat{x}(k-1)]$$

$$v_c^{\frac{1}{2}}(\cdot)v_c^{\frac{T}{2}}(\cdot)=H(\cdot)F(\cdot)P(k-1)F^T(\cdot)H^T(\cdot)+H(\cdot)\Gamma(\cdot)Q(k-1)\Gamma^T(\cdot)H^T(\cdot)+R(k)$$

对于(4)式的计算可通过对右边四个[ ]里项进行并行的 Faddeev 运算.

模型(2)的平方根算法如下:

$$T_1(k)\begin{bmatrix}P^{\frac{T}{2}}(k-1)f^T(\cdot)c^T(\cdot\cdot) & P^{\frac{T}{2}}(k-1)f^T(\cdot)\\ Q_1^{\frac{T}{2}}(k-1)\Gamma_1^T(\cdot)c^T(\cdot\cdot) & Q_1^{\frac{T}{2}}(k-1)\Gamma_1^T(\cdot)\\ R^{\frac{T}{2}}(k) & 0\end{bmatrix}$$

$$=\begin{bmatrix}v_{e_1}^{\frac{T}{2}}(\cdot) & v_{e_1}^{-\frac{1}{2}}(\cdot)c(\cdot\cdot)P_1(k|k-1)\\ 0 & P_1^{\frac{T}{2}}(k)\\ 0 & 0\end{bmatrix}\tag{5}$$

$$\hat{x}^T(k)=[f(\cdot)\hat{x}(k-1)]^T+[z(k)-c(\cdot\cdot)f(\cdot)\hat{x}(k-1)]^T[v_{e_1}^{\frac{T}{2}}(\cdot)]^{-1}\times[v_{e_1}^{-\frac{1}{2}}(\cdot)c(\cdot\cdot)P_1(k|k-1)]\tag{6}$$

式中 $T_1(k)$ 是使 (5) 式初始矩阵三角化的正交阵，$c(\cdot\cdot)=c[\bar{x}(k-1),\bar{y}(k-1)]$，$f(\cdot)=f[\bar{y}(k-1)]$，(6) 式用 Faddeev 算法并行计算.

$$v_{e_1}^{\frac{1}{2}}(\cdot)\ v_{e_1}^{\frac{T}{2}}(\cdot)=c(\cdot\cdot)f(\cdot)P_1(k-1)f^T(\cdot)c^T(\cdot\cdot)+c(\cdot\cdot)\Gamma_1(\cdot)Q_1(k-1)\Gamma_1^T(\cdot)c^T(\cdot\cdot)+R(k)$$

3 脉动阵列实现 MEKF

上述平方根算法很容易映射到一个$(n+r)\times(n+r)$和一个$(m+n+r)\times(m+n+r)$的三角阵列上，其结构和数据流见图 2. 整个执行过程可分三个部分进行：首先将数据流的前两行分别输入相应的三角阵列进行 QR 分解完成(3)式和(5)式的运算.第二部分是将数据流的第三行输入相应的前 r 行梯形阵列执行 Faddeev 计算，完成(4)式和 (6) 式的运算，左边阵列输出的 $[\bar{x}^T(k),\bar{y}^T(k)]$ 用来计算下次迭代所需的 $F(\cdot)$、

$H(\cdot)$、$\Gamma(\cdot)$、$f(\cdot)$、$c(\cdot)$、$c(\cdot\cdot)$. 右边阵列得到 $\hat{x}(k)$ 存贮在梯形阵列下部的菱形阵列里. 第三部分是矩阵相乘运算. 为了下次迭代计算，需将数据流的第四、五两行分别与下部三角阵列中的 $P^{\frac{T}{2}}(k)$ 相乘，相乘的结果输出后反馈到输入端，在右边的阵列中还需将四、五、六三行分别与菱形阵列中的 $\hat{x}(k)$相乘，然后输出与 $z(k+1)$ 相减得到 Δz 和 Δz_1 作为下次迭代的输入. $\hat{x}(k)$ 还要用来计算 $F(\cdot)$，$\bar{x}(k+1|k)$,$\bar{y}(k+1|k)$,

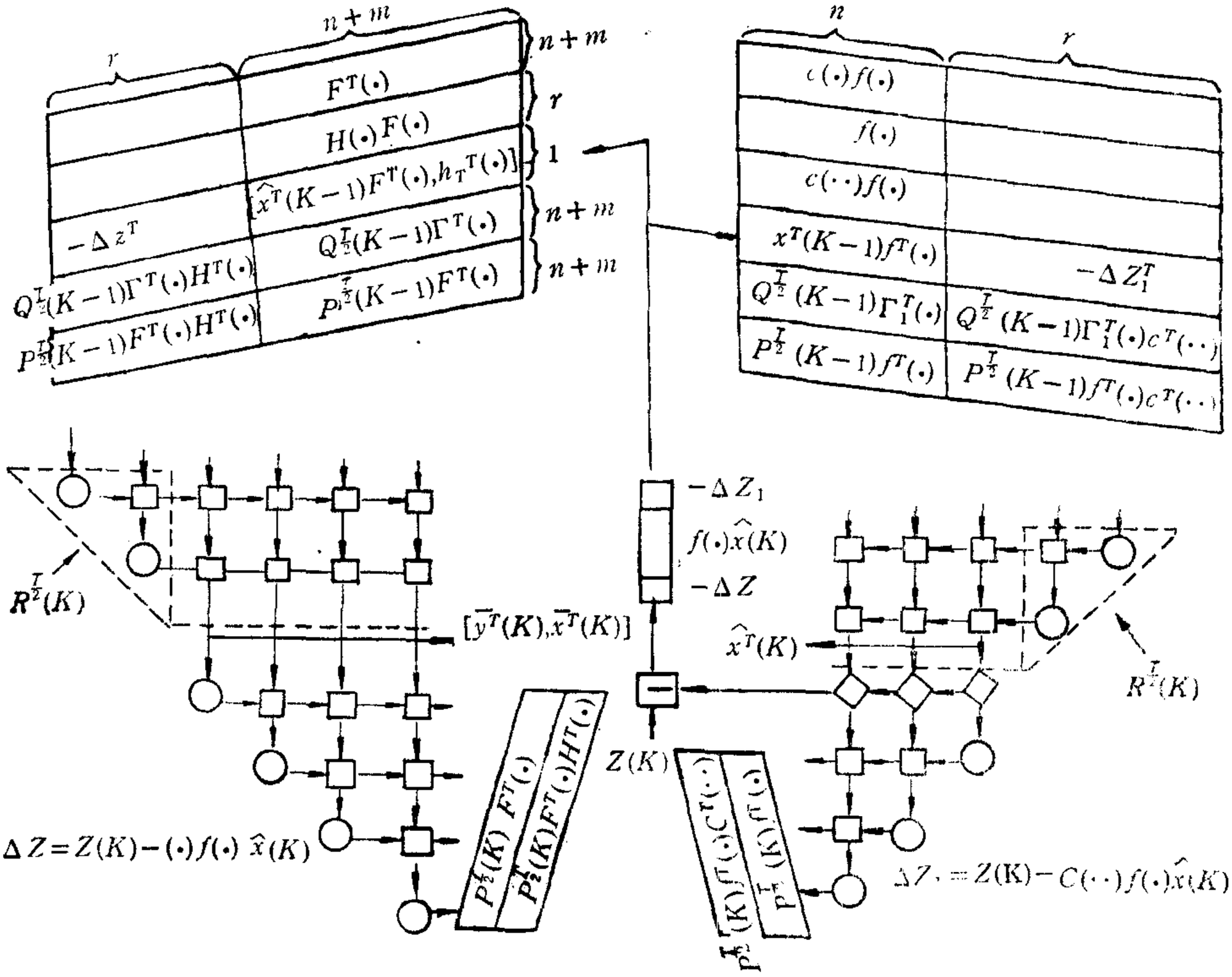
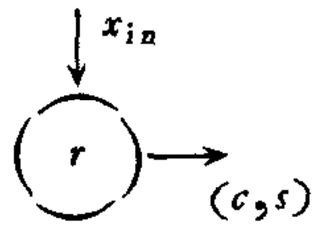
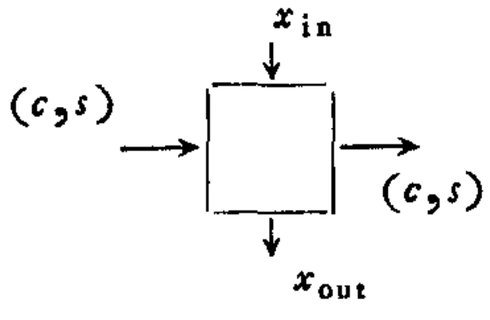


图 2　实现 MEKF 的阵列结构及数据流

表 1　各单元的运算功能

单元 \ 算法	QR 分解	Faddeev 运算	矩阵相乘
	If $x_{in}=0$ then $c=1$; $s=0$ else $r'=(r^2+x_{in}^2)^{\frac{1}{2}}$ $c=r/r'$; $s=x_{in}/r'$ $r=r'$ end	$s=x_{in}/r$	$s=x_{in}\cdot r$
	$x_{out}=c\cdot x_{in}-s\cdot r$ $r=s\cdot x_{in}+c\cdot r$	$x_{out}=x_{in}-s\cdot r$	$s_{out}=s_{in}+x_{in}\cdot r$

$c(\cdot)$. 每部分计算时各处理单元的运算功能见表 1. 上述计算中假设 $H(\cdot)$, $F(\cdot)$, $\Gamma(\cdot)$, $c(\cdot)$, $c(\cdot\cdot)$, $Q^{\frac{T}{2}}(k)\Gamma^T(\cdot)$ 以及 $Q_{\frac{T}{2}}(\cdot)\Gamma^T(\cdot)H^T(\cdot)$ 之预先得到.

整个阵列结构共有 $n+1+[(n+m+r)(n+m+r+1)+(n+r)(n+r+1)]/2$ 个处理单元,在合理地组织数据流的情况下,完成一次迭代计算需 $3n+3m+2r$ 个计算步 (定义边界单元执行一次运算为一个计算步). 左边的阵列在执行 QR 分解、Faddeev 计算和矩阵相乘时所使用的处理单元分别为 $(n+m+r)(n+m+r+1)/2$、$r(2n+2m+r+1)/2$、$(m+n)(m+n+1)/2$, 每个单元所需的计算步分别为 $2n+2m$、1、$n+m+r$. 对于右边阵列进行上述三种运算所需的处理单元分别为$(n+r)(n+r+1)/2$、$(2n+r+1)/2$ 和 $n(n+1)/2$, 所使用的单元进行的计算步分别为 $2n$、1、$2r+n$, 另外菱形部分共 n 个单元, 每个单元进行了 $n+2r$ 个计算步. 通过上述分析整理可得整个处理器的利用率约在 60% 左右.

4 结论

MEKF 方法可改善滤波效果,平方根 MEKF 方法可提高计算的数值稳定性,本文所使用的平方根算法很容易映射到阵列结构上, 通过使用 $O[(n+m)^2]$ 个处理单元可使滤波计算的复杂性由 $O[(n+m)^3]$ 变为 $O(m+n)$, 同文献 [3] 中所提出的算法和结构相比,本文的算法更简单,处理器的利用率更高,硬件的需求量少,迭代周期短, 因此是一种很有效的实时 MEKF 方法.

参 考 文 献

[1] Baheti R S. Mapping extended Kalman fihers onto linear array. *IEEE Trans* 1990, **AC-35** (12): 1310—1319.

[2] Chui C K. Modified extended Kalman filtering and a real-time parallel algorithm for System parameter identification. *IEEE Trans* 1990, **AC-35** (1): 100—104.

[3] LU M. A Parallel square-root algorithm for modified extenede Kalman filtr. *IEEE Trans* 1992, **AES-28** (1): 153—162.

[4] Mccanny J. Systolic array processors. Prentice Hall 1989.

A PARALLEL ARCHITECTURE FOR MODIFIED EXTENDED KALMAN FILTER

MU DEJUN　DAI GUANZHONG

(Department of Automatic Control, Northwestern Polytechnical University Xi'an 710072)

ABSTRACT

A new systolic architecture based on the square-root algorithm for parallel Modified Extended Kalman Filter (MEKF) is proposed in this paper. This algorithm is more numerical stable. Comparing with other systolic arrays implementing square root MEKF existing in literatures, this systolic architecture with algorithm in this paper has higher efficiency and uses fewer time-steps for a complete iteration at each instant.

Key words: Parallel algorithm, Systolic array, Kalman filter.