

# 并行卡尔曼滤波及其心动阵列实现综述\*

张友民 戴冠中 张洪才

(西北工业大学自动控制系·西安, 710072)

**摘要:** 本文综述了基于心动(Systolic)阵列结构的并行卡尔曼滤波各种算法及其心动阵列实现,对各种阵列滤波算法在处理单元数,计算时间和效率等方面进行了比较,并对进一步的研究工作提出一些展望。

**关键词:** 并行处理;心动阵列处理器;卡尔曼滤波;平方根滤波;推广卡尔曼滤波

## 1 引言

自从卡尔曼滤波提出以来,它已成为信号处理、通信和控制等最基本、最重要的工具,并成功地应用到航空航天及工业控制等各个领域.但由于滤波本身计算量大,同时受目前计算机结构和速度的限制,形成所谓的计算“瓶颈”,严重地制约了其在线实时应用.尽管为了改善卡尔曼滤波的计算复杂性和数值稳定性,各种基于平方根滤波、序列滤波以及状态与参数解耦的卡尔曼滤波等方法得到不断发展,但这些滤波方法的计算复杂性仍是 $O(n^3)$ ,远不能满足越来越迫切的实时状态估计的需要,以英国计算机有限公司 INMOS 设计和生产的适于并行处理的晶片机(Transputer)为代表的新一代 VLSI 设计和实现理论,突破了传统的以 CPU 为中心的典型冯·诺依曼结构,采用以互连网络为中心的结构,为实现并行处理计算奠定了强有力的从硬件到软件的基础<sup>[1]</sup>.而以这种基于 VLSI 结构并行处理的卡尔曼滤波实现为解决卡尔曼滤波的实时处理和应用开辟了广阔的前景.

并行卡尔曼滤波及其心动阵列实现研究,自 Andrews<sup>[2]</sup>和 Jover-Kailath<sup>[3]</sup>开拓性的工作以来,迄今已提出许多新的不同结构形式的心动阵列卡尔曼滤波设计和实现方案<sup>[5~40]</sup>.本文综述了标准卡尔曼滤波,平方根滤波及推广卡尔曼滤波等各种心动阵列实现的最新发展,并对各种阵列滤波算法在所需处理单元(PE)数,迭代一次的时间步数及处理器的效率(利用率)等方面进行了比较,最后,对今后的研究工作提出一些展望和建议.

## 2 心动阵列的基本原理及有关矩阵计算的心动阵列算法

### 2.1 心动阵列的基本原理和特点

“Systolic”一词源于生理学,其原意是指心脏有节奏地做周期性跳动,并把血液以脉动方式送往身体各个部位.孔祥重教授等借用这一概念,首先提出了心动阵列结构<sup>[4]</sup>.心动阵列是一种典型的并行结构,它是由功能相同或相近的 PE 按照某种规则构成的一种阵列结构,它具有规整化、模块化、局部数据传输等特点,很适合于用 VLSI 技术来实现,并特别适于具有局部相关关系的线性递推计算,如卡尔曼滤波计算等.其基本原理是用多个

\* 国家自然科学基金与航空科学基金资助课题.

本文于1992年4月3日收到.1992年10月19日收到修改稿.

PE 构成的阵列取代原来单一的 PE,使得输入数据得以重复利用,整个阵列可获得更高的计算速度.心动阵列可解决一大类基本计算问题.其中包括大部分矩阵运算,数字与图象处理,卡尔曼滤波以及许多非数值计算问题.不同的算法有不同的阵列结构,同一算法也可有许多不同的阵列实现.从结构上,心动阵列可分为一维和二维线性阵列两大类,二维线性阵列又包括矩形、六角形和三角形等不同结构形式.

## 2.2 滤波计算常用的矩阵运算及其心动阵列实现

### 2.2.1 矩阵相乘和相加

对于矩阵  $A, B, C$  的相乘运算  $D=AB+C$ ,其递推运算可描述为  $d_{ij}=c_{ij}+\sum_{k=1}^n a_{ik}b_{kj}$ . 上述递推关系可由心动阵列实现(图 1(a)). 其中若  $C$  为零阵,则只需计算  $AB$ ,如果  $B$  为三角矩阵或向量形式,则仅需用三角形阵列或一维线性阵列来实现.

### 2.2.2 回代计算

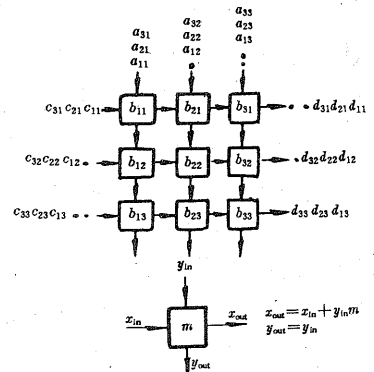
在卡尔曼滤波、特别是平方根滤波中,经常碰到三角阵运算及求解问题.当  $n$  阶线性方程组  $Ax=b$  经三角分解后,该方程的求解就变为一个回代计算问题,其递推关系可描述为  $x_i=(b_i-\sum_{j=i+1}^n a_{ij}x_j)/a_{ii}$ ,图 1(b)给出了上述递推关系的一种二维退化心动阵列实现.

### 2.2.3 正交分解

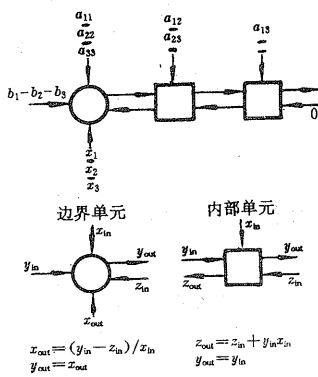
常用的正交分解有 Givens,修正的快速 Givens 及修正的 Gram-Schmidt 正交化法等,这些算法均可用如图 1(c)所示的三角形阵列结构来实现,只是相应的 PE 功能有所不同<sup>[18]</sup>.

### 2.2.4 Fadeev 算法及其心动阵列实现

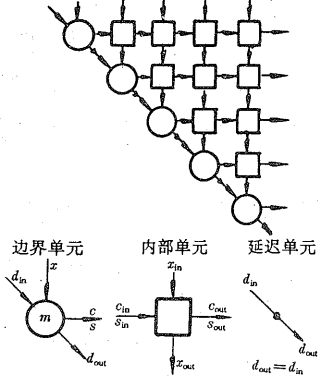
设已知  $A, B, C, D$  四个矩阵,其中  $A$  为非奇异阵,若由  $A, B, C, D$  构成如下矩阵  $F=\begin{bmatrix} A & B \\ -C & D \end{bmatrix}$ . 则 Fadeev 算法是对  $F$  阵进行如下变换,使  $A$  变为上三



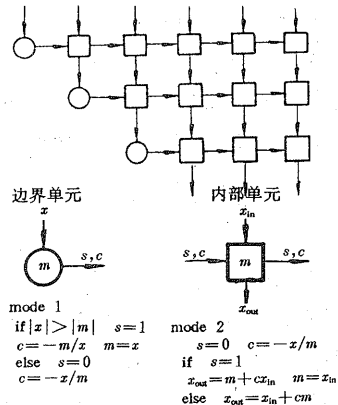
(a) 矩阵乘



(b) 回代计算



(c) 正交分解



(d) Fadeev 算法

图 1 几种矩阵运算的心动阵列实现

角阵,  $-C$  变为零阵, 即

$$\begin{bmatrix} A & B \\ -C & D \end{bmatrix} \rightarrow \begin{bmatrix} A & B \\ -C + WA & D + WB \end{bmatrix} \rightarrow \begin{bmatrix} A & B \\ 0 & D + CA^{-1}B \end{bmatrix}.$$

式中选  $W = CA^{-1}$ , 则  $-C + WA = 0$ ,  $D + WB = D + CA^{-1}B$ . 若首先通过正交变换使  $A$  三角化, 然后再通过变换使左下角阵元素为零即得到修正的 Fadeev 算法及心动阵列实现(图 1(d)). 为了改善算法的数值稳定性, 对  $A$  的正交分解常用邻主元素高斯消去法使之三角化, 而在零化  $-C$  时只能用简单高斯消去法, 因此, 边界单元对应有两种运算模式.

### 3 卡尔曼滤波及其心动阵列实现

自 Andrews<sup>[2]</sup>首次提出卡尔曼滤波的并行处理以后, Jover-Kailath<sup>[3]</sup>提出一种基于 U-D 分解实现的卡尔曼滤波方案, 类似的研究有 Hashimoto<sup>[5]</sup>和 Behera<sup>[6]</sup>的工作, 但上述研究均只解决了量测更新的并行处理, 未涉及时间更新, 并不是完整的卡尔曼滤波. Trassor<sup>[7]</sup>基于把卡尔曼滤波的时间更新和量测更新解耦的思想提出一种并行解耦滤波及其心动阵列结构, 本文作者为了改善滤波的数值稳定性和计算效率, 给出了更一般的并行解耦滤波算法及其 U-D 分解实现, 并在国内研制的 PD-100 型并行仿真计算机上进行了仿真和实际应用的计算<sup>[8]</sup>. 本文下面综述现有的各种并行滤波方法及其心动阵列实现.

#### 3.1 标准卡尔曼滤波及其心动阵列实现

##### 3.1.1 协方差卡尔曼滤波<sup>[9]</sup>

设考虑如下线性离散系统

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k), \quad (1)$$

$$z(k) = C(k)x(k) + v(k). \quad (2)$$

式中  $x(k) \in \mathbb{R}^n$  是状态向量,  $z(k) \in \mathbb{R}^m$  是量测向量,  $u(k) \in \mathbb{R}^p$  是已知输入向量,  $w(k) \in \mathbb{R}^n$  是系统噪声向量,  $v(k) \in \mathbb{R}^m$  是量测噪声向量, 假设系统噪声和量测噪声是互不相关的零均值高斯白噪声, 方差阵分别为  $Q(k)$ ,  $R(k)$ . 则协方差卡尔曼滤波方程表示为<sup>[9]</sup>

$$\hat{x}(k+1|k) = A(k)\hat{x}(k|k-1) + A(k)K(k)[z(k) - C(k)\hat{x}(k|k-1)], \quad (3)$$

$$P(k+1|k) = A(k)P(k|k-1)A^T(k) - A(k)K(k)C(k)P(k|k-1)A^T(k) + Q(k). \quad (4)$$

式中

$$K(k) = P(k|k-1)C^T(k)[C(k)P(k|k-1)C^T(k) + R(k)]^{-1}. \quad (5)$$

上述滤波可利用流水线心动阵列结构来实现<sup>[10]</sup>, 其中矩阵求逆可采用迭代计算近似, 但这可能会带来较大的估计误差.

##### 3.1.2 协方差/信息卡尔曼滤波<sup>[10~12]</sup>

Yeh 基于 Fadeev 算法利用图 1(d)的梯形阵列实现了一种联合协方差/信息滤波算法, 具体实现步骤如表 1 所示<sup>[10,12]</sup>.

如果上述 8 步运算用一个图 1(d)所示梯形阵列来实现, 则完成一次迭代计算需  $16n$  时间步(time-step), 处理器的利用率约为 80%<sup>[10]</sup>. 文[12]对[10]的梯形阵列用 Occam 语言进行了仿真, 并提出与[11]类似的改进措施, 若每步计算使用不同的阵列, 与仅使用一个阵列相比可加快速度 516%<sup>[12]</sup>.

表 1 协方差/信息卡尔曼滤波的 Fadeev 算法实现步骤

步骤	A	B	C	D	D+CA <sup>-1</sup> B(结果)
1	I	$\hat{x}(k-1 k-1)$	$-A(k-1)$	$B(k-1)u(k-1)$	$\hat{x}(k k-1)$
2	$P^{-1}(k-1 k-1)$	$A^T(k-1)$	$-A(k-1)$	$Q(k-1)$	$P(k k-1)$
3	$R(k)$	I	$C^T(k)$	0	$C^T(k)R^{-1}(k)$
4	$P(k k-1)$	I	$-I$	0	$P^{-1}(k k-1)$
5	I	$C(k)$	$-C^T(k)R^{-1}(k)$	$P^{-1}(k k-1)$	$P^{-1}(k k)$
6	$P^{-1}(k k)$	$C^T(k)R^{-1}(k)$	I	0	$K(k)$
7	I	$\hat{x}(k k-1)$	$C(k)$	$z(k)$	$z(k) - C(k)\hat{x}(k k-1) = \Delta z(k)$
8	I	$\Delta z(k)$	$-K(k)$	$\hat{x}(k k-1)$	$\hat{x}(k k)$

### 3.2 平方根卡尔曼滤波及其心动阵列实现

为改善滤波的数值稳定性,各种形式的平方根滤波得到广泛的研究和应用,并已有多种心动阵列结构实现.

#### 3.2.1 平方根协方差滤波(SRCF) I<sup>[14,15]</sup>

Sung-Hu<sup>[14]</sup>基于 Morf-Kailath<sup>[13]</sup>提出的平方根协方差滤波,利用每一迭代可用多个阵列实现的思路提出一种心动阵列结构.该算法包含有两个正交变换,分别完成时间更新和量测更新,其滤波方程如下:

$$\hat{x}(k+1|k) = A(k)\hat{x}(k|k) + B(k)u(k), \quad (6)$$

$$T_1(k) \begin{bmatrix} P^{\Gamma/2}(k|k)A^T(k) \\ Q^{\Gamma/2}(k|k) \end{bmatrix} = \begin{bmatrix} P^{\Gamma/2}(k+1|k) \\ 0 \end{bmatrix}, \quad (7)$$

$$R_o(k+1) = C(k+1)P(k+1|k)C^T(k+1) + R(k+1), \quad (8)$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + P(k+1|k)C^T(k+1)R_o^{-1}(k+1) \cdot [z(k+1) - C(k+1)\hat{x}(k+1|k)], \quad (9)$$

$$T_2(k) \begin{bmatrix} R^{\Gamma/2}(k+1) & 0 \\ P^{\Gamma/2}(k+1|k)C^T(k+1) & P^{\Gamma/2}(k+1|k) \end{bmatrix} = \begin{bmatrix} R_o^{\Gamma/2}(k+1) & R_o^{-1/2}(k+1)C(k+1)P(k+1|k) \\ 0 & P^{\Gamma/2}(k+1|k+1) \end{bmatrix}. \quad (10)$$

其阵列实现分成协方差更新和状态更新两大回路,考虑到方程(10)左端矩阵第二列仅包含一个上三角阵,正交化时,可将  $P^{\Gamma/2}(k+1|k)C^T(k+1)$ ,  $P^{\Gamma/2}(k+1|k)$  按照从最下面一行向上的顺序输入三角阵列,以保证三角化过程中  $P^{\Gamma/2}(k+1|k)$  始终保持上三角阵,因此可将三角阵列截成梯形阵列,从而减少了计算量和处理单元.但由于矩阵计算阵列不能重复使用,该阵列 PE 利用率较低.文<sup>[15]</sup>则给出了预测形式 SRCF 的心动阵列实现.

#### 3.2.2 平方根协方差滤波(SRCF) II<sup>[16~18]</sup>

Irwin-Gaston 根据 Morf-Kailath 的组合时间更新和量测更新的平方根协方差预测方程(SRCP),给出了另一种心动阵列实现<sup>[16,18]</sup>,滤波方程如下:

$$T(k) \begin{bmatrix} P^{\Gamma/2}(k|k-1)C^T(k) & P^{\Gamma/2}(k|k-1)A^T(k) \\ R^{\Gamma/2}(k) & 0 \\ 0 & Q^{\Gamma/2}(k) \end{bmatrix}$$

$$= \begin{bmatrix} R_o^{T/2}(k) & R_o^{-1/2}(k)C(k)P(k|k-1)A^T(k) \\ 0 & P^{T/2}(k+1|k) \\ 0 & 0 \end{bmatrix}, \quad (11)$$

$$\hat{x}(k+1|k) = A(k)\hat{x}(k|k-1) + B(k)u(k) + A(k)P(k|k-1)C^T(k)R_o^{-1}(k)[z(k) - C(k)\hat{x}(k|k-1)]. \quad (12)$$

其阵列实现如图 2 所示. 该阵列分成四步执行. 首先对(11)式进行正交分解使之三角化, 然后在阵列上面的  $m$  行进行 Fadeev 运算以获得  $\hat{x}(k+1|k)$ , 为了得到下次更新的输入, 第三步需把  $\hat{x}(k+1|k)$  和  $P^{T/2}(k+1|k)$  分别由  $A^T(k+1)$  和  $C^T(k+1)$  右乘, 最后进行相加运算.

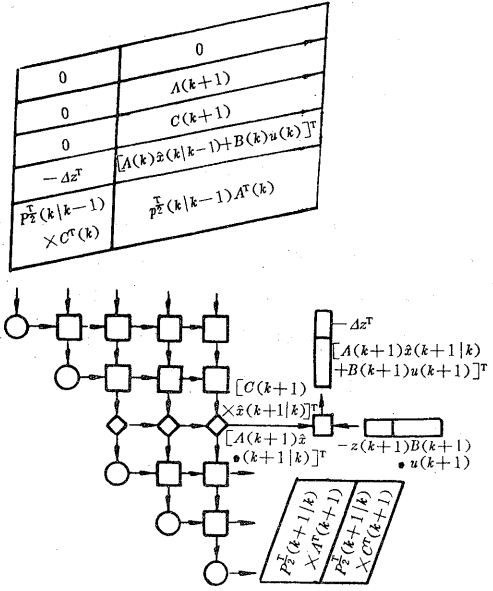


图 2 平方根协方差滤波 II 的心动阵列实现

Maguire-Irwin<sup>[17]</sup>又给出了 SRCF 的心动阵列实现, 其实现与图 2 相同, 只是输入输出的数据流有所不同. 该文还对该阵列利用 4 个 Transputer 进行了仿真计算及滤波性能的比较.

3.2.3 平方根信息滤波 (SRIF) I [22~27]

Paige-Saunders<sup>[21]</sup>基于把卡尔曼滤波转换为递推最小二乘估计的思想, 提出一种组合时间更新和量测更新的平方根信息滤波 (SRIF) 方案. Chen-Yao<sup>[22]</sup>根据上述算法提出了一种心动阵列实现. 其中包括滤波 (SRIF) 和预测 (SRIP) 两种形式. SRIP 方程如下<sup>[18]</sup>

$$T^T(k) \begin{bmatrix} P^{-T/2}(k|k-1) & 0 & P^{-T/2}(k|k-1)\hat{x}(k|k-1) \\ R^{-T/2}(k)C(k) & 0 & R^{-T/2}(k)z(k) \\ Q^{-T/2}(k)A(k) & -Q^{-T/2}(k) & -Q^{-T/2}(k)B(k)u(k) \end{bmatrix} = \begin{bmatrix} R_w^T & -R_w^{-1}A^T(k)Q^{-1}(k) & R_w^{-1}r \\ 0 & P^{-T/2}(k+1|k) & P^{-T/2}(k+1|k)\hat{x}(k+1|k) \\ 0 & 0 & * \end{bmatrix}. \quad (13)$$

式中

$$R_w R_w^T = P^{-1}(k|k-1) + C^T(k)R^{-1}(k)C(k) + A^T(k)Q^{-1}(k)A(k), \quad (14)$$

$$r = P^{-1}(k|k-1)\hat{x}(k|k-1) + C^T(k)z(k) - A^T(k)Q^{-1}(k)B(k)u(k). \quad (15)$$

文[22]给出了 SRIP 的一种梯形心动阵列实现. 该阵列在对  $P^{T/2}(k+1|k)$  更新时, 避免了要把下三角阵部分提取出来然后再重新装入上三角部分的麻烦, 但没有利用  $Q(k)$  的三角阵结构. Chen-Yao 在文[24, 25]又给出了 SRIF 的一种矩阵分解法的推导及心动阵列实现. Lincoln-Yao<sup>[26]</sup>利用相关图技术, 给出了 SRIF 的一种处理器利用率较高的三角形心动阵列实现. Gosling<sup>[27]</sup>为了克服 SRIF 各种阵列实现中状态估计不能直接求得的缺陷, 给出了一种直接求取状态向量的心动阵列结构, 该阵列比 Chen-Yao<sup>[22]</sup>及 Gaston-Irwin<sup>[19, 20]</sup>

的速度快,但处理器利用率不如 Gaston-Irwin 的高.

### 3.2.4 平方根信息滤波(SRIF) II [18~20]

Gaston-Irwin 利用上述 SRIF 和 SRIP 提出了三角形、梯形和矩形结构的心动阵列实现(图 3). 图 3(a)给出了 SRIP 的一种三角形阵列结构,注意到(13)式中左端矩阵中间列仅有一个上三角阵  $-Q^{-T/2}(k)$ ,如果按照修正的 Givens 旋转法进行正交分解,则可将图 3(a)的三角形结构“截”成图 3(b)所示的梯形结构[20]. 与图 3(a)比较,图 3(b)中  $P^{-T/2}(k+1|k)$  和  $P^{-T/2}(k+1|k)\bar{x}(k+1|k)$  没有存放在阵列的下部,因而更容易进入一次迭代计算. 但当(13)式左端阵的第一列进入处理时,阵列的中间部分并没有利用. 通过将阵列的中间部分“迭加”到上述三角阵列上的方法,可进一步提高计算效率,减少 PE 数. 从而得到图 3(c)所示矩形结构阵列. 图中带圆圈的方框对应阵列单元有两种运算方式. 最初作为三角化分解的边界单元,当  $Q^{-T/2}(k)A(k)$  的最后元素通过后,即作为内部单元进行运算. 可见,矩形阵列所需 PE 最少,处理器的效率最高. 文[20]也给出了 SRIF 的三角形、梯形和矩形阵列实现,它们分别与图 3(a)~(c)结构相同,只是输入输出数据流不同. 上述阵列实现的信号为单向传递,这限制了处理器利用率的进一步提高,另外,由于阵列实现中存在大量反馈回路,使得算法的硬件实现成本增加[32].

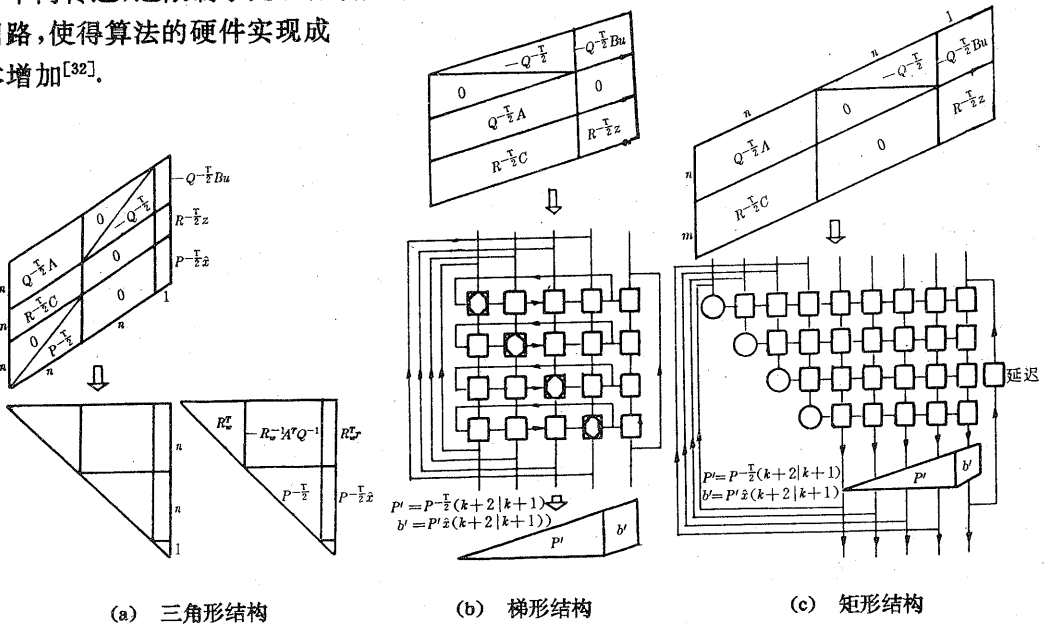


图 3 平方根信息滤波 II (SRIF II)的心动阵列实现

### 3.2.5 平方根信息滤波(SRIF) III [28~32]

Kung-Hwang 同样基于把卡尔曼滤波转化为最小二乘估计的思想[21],给出了在有色噪声和白噪声两种情况下均适用的平方根信息滤波的三角形心动阵列,实现如图 4 所示[28~32]. 该算法实现主要包括两大部分操作,首先利用 QR 三角化分解或 Givens 旋转通过  $S(k)$  零化  $\bar{C}(k)$  和  $\bar{A}(k)$ . 第二步是装入  $Q_w(k)$ , 经过  $3n+m$  时间步,  $\bar{A}(k)$  零化结束后,  $Q_w(k)$  应立即装入上三角阵列. 在  $t=3n+m$  时刻,需对阵列进行某些控制,使阵列的边界单元完成内部单元的功能,并把信息通道转换成由下向上的方向[32].

### 3.2.6 推广卡尔曼滤波方法及应用<sup>[33~41]</sup>

Baheti 等人近年来针对目标跟踪问题, 利用 Warp 可编程线性心动阵列计算机, 对推广卡尔曼滤波(EKF)及其在 Warp 机上的并行实现进行了探索. Baheti 等首先针对 9 状态 3 量测的目标跟踪问题, 把具有量测序列处理的 EKF 映射到  $n+2$  个 PE 组成的线性阵列上, 并在 Warp 机上予以实现<sup>[33,34]</sup>. 为了提高 EKF 的数值稳定性, Baheti 等利用 LDU<sup>T</sup> 矩阵因子分解滤波的思想<sup>[3]</sup>, 利用 Givens 旋转对组合时间更新和量测更新的滤波方程进行三角化运算. 在 Warp 机上利用  $n+m+1$  处理单元实现了因子分解的卡尔曼滤波(FKF)及其在目标跟踪中的应用<sup>[36~38]</sup>. 类似的研究有 Atherton 等<sup>[39~41]</sup>, Atherton 利用多处理器进行多目标跟踪的并行处理, 给出协方差卡尔曼滤波及其在 Transputer 计算机网络上的仿真计算<sup>[41]</sup>.

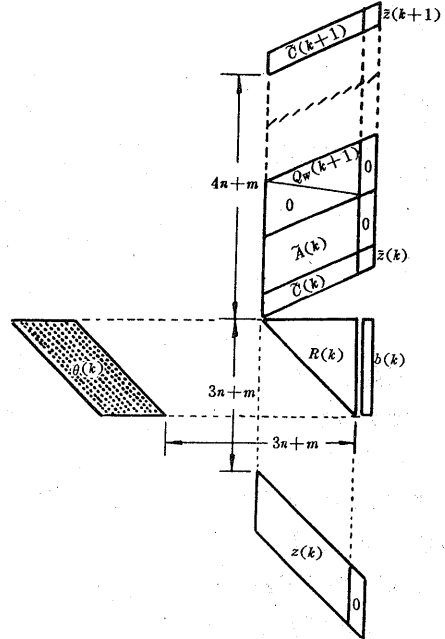


图 4 平方根信息滤波Ⅲ的心动阵列实现

## 4 各种并行卡尔曼滤波算法的比较

### 4.1 基于 VLSI 的并行卡尔曼滤波性能评估指标

基于 VLSI 实现的心动阵列结构, 使用了大量的具有一定互连方式和相同逻辑功能的 PE, 其计算时间的减少是以增加大量硬件为代价的. 因此, 计算时间不再是评估并行卡尔曼滤波的唯一主要指标, 而要同时考虑算法的计算时间和所需的 PE. 其常用的主要性能指标有加速比和效率<sup>[17]</sup>. 加速比  $S_p$  定义为串行算法在单处理器(机)上的运行时间  $T_1$  与并行算法在  $P$  个处理器(机)上的运行时间  $T_p$  之比, 即:  $S_p = T_1/T_p$ ; 效率  $E_p$  定义为加速比与所用处理器个数的百分比. 用来衡量处理器的利用率, 即:  $E_p = (S_p/P) \times 100\%$ .

### 4.2 各种并行心动阵列滤波性能比较

为了对各种并行心动阵列滤波的性能有一全面了解, 表 2 给出了各种滤波算法在阵列实现中所用的 PE 数, 迭代一次所需时间步数以及阵列的效率.

表 2 各种并行心动阵列滤波的性能比较

Systolic 卡尔曼滤波	处理单元数 (阵列大小)	一次迭代的时间步 (计算时间)	效率
• 协方差滤波 <sup>[9]</sup>	$n^2$	$9n+22m$	22~52%
• 协方差/信息滤波 <sup>[10,11]</sup>	$n(3n+1)/2$	$9n+4m+3$	81~100%
• 平方根协方差滤波			
I Sung-Hu <sup>[14,15]</sup>	$(5n^2+m^2+6m+3n+3m+2)/2$	$4n+3m-3$	7~26%
II Irwin-Gaston <sup>[16,18]</sup>	$[(n+m)(n+m+1)+2(n+1)]/2$	$2n+2m+1$	38~100%

(续上表)

Systolic 卡尔曼滤波	处理单元数 (阵列大小)	一次迭代的时间步 (计算时间)	效率
· 平方根信息滤波			
I Chen-Yao <sup>[22,23]</sup>	$n(3n+5)/2$	$4n+m$	21~31%
Lincoln-Yao <sup>[26]</sup>	$n(5n+13)/8$	$3n+m$	53~60%
Gosling <sup>[27]</sup>	$n(2n+3)$	$2n+m$	37~47%
II Gaston-Irwin <sup>[18~20]</sup>			
三角形结构	$(2n+1)(n+1)$	$4n+m+1$	17~30%
梯形结构	$3n(n+1)/2$	$3n+m$	22~42%
矩形结构	$n(n+1)$	$3n+m$	33~63%
III Kung-Hwang <sup>[28~32]</sup>	$n(n+3)/2$	$4n+m$	50~60%
· 推广卡尔曼滤波			
I 量测序列滤波 <sup>[33,37]</sup>	$n+2$	0.125ms( $S_F=7$ )	87%
II 因子分解滤波 <sup>[35~38]</sup>	$n+m+1$	0.8ms( $S_F=2$ )	20%

## 5 结语及展望

随着 VLSI 技术的迅猛发展,并行卡尔曼滤波近年来在国际上受到人们的极大重视和广泛研究.短短几年,已对上述各种滤波方法进行了广泛研究,并应用于目标跟踪、信号处理等问题中,成为估计理论的最新并具有挑战性的研究课题之一.但目前工作基本上处于算法的结构研究阶段,还未形成一套完整的理论、算法、结构和实现体系,比较成熟、实用的芯片还很少,成功的应用实例也不多;算法结构(特别是平方根滤波)还有待进一步改进,以减少处理单元,提高效率;新的基于心动阵列,波前(Wavefront)阵列的算法结构尚需进一步探讨和研究,参数估计的心动阵列结构也需大力研究.目前,国内已研制出由 INMOS-T800 晶片机构成的 PD-100 并行计算机以及用 TMS-320C25 组成的并行计算机等,这为并行滤波的仿真和实际计算打下良好的硬件基础,不失时机地开展并行卡尔曼滤波及其包括在并行仿真、目标跟踪、飞行状态与参数估计以及控制等各个方面的应用研究是国内亟待解决的课题,具有广阔而且诱人的前景.

## 参 考 文 献

- [1] 陈国良,陈峻.VLSI 计算理论与并行算法.合肥:中国科学技术大学出版社,1991
- [2] Andrews, A.. Parallel Processing of Kalman Filter. Proc. Int. Conf. on Parallel Processing, 1981, 216-220
- [3] Jover, J. and Kailath, T.. A Parallel Architecture for Kalman Filter Measurement Update and Parameter Estimation. Automatica, 1986, 22(1):43-57
- [4] Kung, H. T.. Why Systolic Architectures? IEEE Computer, 1982, 15(1):37-46
- [5] Hashimoto, K. and Kimura, H.. A Parallel Architecture for Recursive Least Square Identification. IEEE Proc. ICASSP, 1986, 1185-1188
- [6] Behera, P. K.. Implementation of Factorized Filter in Systolic Arrays. IEEE Proc. ICASSP, 1988, 2065-2068
- [7] Kung, S. Y., Whitehouse, H. J. and Kailath, T.. VLSI and Modern Signal Processing. Prentice-Hall, Inc., Englewood



- Cliffs, 1985;375—388
- [8] Zhang Youmin, Zhang Hongcai and Dai Guangzhong. A Parallel Decoupled Kalman Filtering Algorithm and Architecture. Proc. Int. Conf. Syst. Simu. and Scient. Comput., Beijing, 1992, 232—236
- [9] Papadourakis, G. M. and Taylor, F. J.. Implementation of Kalman Filter Using Systolic Arrays. IEEE Proc. ICASSP, 1987, 783—786
- [10] Yeh, H. G.. Kalman Filtering and Systolic Processors. IEEE Proc. ICASSP, 1986, 2139—2142
- [11] Yeh, H. G.. Systolic Implementation on Kalman Filters. IEEE Trans. Acoustics, Speech and Signal Processing, 1988, 36(9);1514—1517
- [12] Irwin, G. W. and Gaston, F. M. F.. Occam Simulation of a Systolic Architecture for Parallel Kalman Filtering. Parallel Processing in Control, Fleming, P. I. Ed., 1988, 215—231
- [13] Morf, M. and Kailath, T.. Square Root Algorithms for Least Squares Estimation. IEEE Trans. Automat. Contr., 1975, AC-20(4);487—497
- [14] Sung, T. Y. and Hu, Y. H.. VLSI Implementation of Real-Time Kalman Filter. Proc. ICASSP, 1986, 2223—2226
- [15] Sung, T. Y. and Hu, Y. H.. Parallel VLSI Implementation of the Kalman Filter. IEEE Trans. Aerospace and Electronic Systems, 1987, 23(2);215—224
- [16] Irwin, G. W. and Gaston, F. M. F.. A Systolic Architecture for Square Root Covariance Kalman Filtering. Int. Conf. Systolic Arrays, 1989, 255—263
- [17] Maguire, L. P. and Irwin, G. W.. Transputer Implementation of Kalman Filters. IEE Proceedings, Pt. D, 1991, 138(4);355—362
- [18] Gaston, F. M. F. and Irwin, G. W.. Systolic Kalman Filtering; an Overview. IEE Proceedings, Pt. D, 1990, 137(4);235—244
- [19] Gaston, F. M. F. and Irwin, G. W.. A Systolic Square Root Information Kalman Filter. IEEE Int. Conf. Systolic Arrays, 1988, 643—652
- [20] Gaston, F. M. F. and Irwin, G. W.. Systolic Approach to Square Root Information Kalman Filter. Int. J. Control, 1989, 50(1);225—248
- [21] Paige, C. C. and Saunders, M. A.. Least Squares Estimation of Discrete Linear Dynamic Systems Using Orthogonal Transformations. SIAM J. of Numerical Analysis, 1977, 14(1);180—193
- [22] Chen, M. J.. On Realizations of Least-Squares Estimation and Kalman Filtering. Proc. 1st Int. Workshop on Systolic Arrays, 1986, 161—170
- [23] Chen, M. J.. On Realization and Implementation and Kalman Filtering by Systolic Array. Proc. the 21th Conf. on Information Science and Systems, 1987, 375—380
- [24] Chen, M. J. and Yao, K.. Recursive Least Squares Estimation and Kalman Filtering by Systolic Arrays. Proc. 1st Int. Conf. on Advances in Communication Control Systems, 1988, 26—37
- [25] Chen, M. J. and Yao, K.. Systolic Kalman Filtering Based on QR Decomposition. Proc. of SPIE, 1987, 826;25—32
- [26] Lincoln, R. A. and Yao, K.. Efficient Systolic Kalman Filtering Design by Dependence Graph Mapping. Proc. VLSI Signal Processing III, 1988, 396—407
- [27] Gosling, P. et al. Direct Extraction of The State Vector From Systolic Implementation of the Square Root Kalman Filter. Int. Conf. on Systolic Arrays, 1989, 42—51
- [28] Kung, S. Y. and Hwang, J. N.. Systolic Architectures for Kalman Filtering. 21th Conf. on Signals, System and Computer, 1987, 746—751
- [29] Kung, S. Y. and Hwang, J. N.. Systolic Designs for State Models; Kalman Filtering and Neural Network. Proc. the 26th Conf. on Decision and Control, 1987, 1461—1467
- [30] Kung, S. Y.. VLSI Array Processors. Prentice-Hall, Englewood Cliffs, 1988
- [31] Kung, S. Y. and Hwang, J. N.. An Efficient Triarray Systolic Design for Real-Time Kalman Filtering. IEEE Proc. ICASSP, 1988, 2041—2044
- [32] Kung, S. Y. and Hwang, J. N.. Systolic Arrays Designs for Kalman Filtering. IEEE Trans. on Signal Processing,

- 1991, 39(1):171—182
- [33] Baheti, R. S. and O'Hallaron, D. R. . Efficient Parallel Implementation of the Tracking Kalman Filter. Proc. of the Conf. on Decision and Control, Austin, Texas, 1988, 376—381
- [34] O'Hallaron, D. R. and Baheti, R. S. . Fast Mapping of a Kalman Filter on Warp. Proc. of SPIE, 1988, 977;196—205
- [35] O'Hallaron, D. R. and Baheti, R. S. . Parallel Implementation of a Kalman Filter on the Warp Computer. Proc. Int. Conf. on Parallel Processing, 1988, 108—111
- [36] Itzkowitz, H. R. and Baheti, R. S. . Demonstration of Square Root Kalman Filter on Warp Parallel Computer. Proc. American Control Conf. , 1989, 1754—1762
- [37] Baheti, R. S. , O'Hallaron, D. R. and Itzkowitz, H. R. . Mapping Extended Kalman Filter onto Linear Arrays. IEEE Trans. on Automat. Contr. , 1990, AC-35(12);1310—1319
- [38] Baheti, R. S. and Itzkowitz, H. R. . Factorized Kalman Filter Mapping on Linear Array Processor. IFAC 10th Congress on Automatic Control, 1990, 70—75
- [39] Phillips, T. C. . Systolic Architecture for Extended Kalman Filtering. Proc. of SPIE, 1987, 826;33—40
- [40] Lee, E. K. B. and Haykin, S. . Parallel Implementation of the Tracking Kalman Filter. IEEE Proc. ICASSP, 1988, 2092—2095
- [41] Atherton, D. P. et al. . Tracking Multiple Targets Using Parallel Processing. IEE Proc. Pt. D, 1990, 137(4);225—234
- [42] Hashemipour, H. R. , Roy, S. and Laub, A. J. . Decentralized Structures for Parallel Kalman Filtering. IEEE Trans. Automatic Control, 1988, 33(1);88—96

## A Survey for Systolic Implementation of Parallel Kalman Filtering

ZHANG Youmin, DAI Guanzhong and ZHANG Hongcai

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

**Abstract:** Focused on the principle of parallel processing for Kalman filtering, a variety of published systolic algorithms for standard, square-root and extended Kalman filtering are reviewed and summarized in this paper. For different systolic array implementation of parallel Kalman filtering, the number of processors required, the number of time-step for each iteration and the processor utilization or efficiency are listed for comparison with each other. Finally, the suggestions for future development of parallel Kalman filtering are presented.

**Key words:** parallel processing; systolic array processor; Kalman filtering; square-root Kalman filtering; extended Kalman filtering

### 本文作者简介

**张友民** 1963年生. 副教授. 1983年毕业于西北工业大学自动控制系, 1986年获该系工学硕士学位, 并留校任教. 1992年晋升为副教授. 主要研究方向为滤波、辨识与随机控制, 控制系统中的并行处理, 故障诊断与容错控制研究等.

**戴冠中** 1937年生. 教授. 自动控制理论及应用学科博士生导师, 西北工业大学校长, 航空航天工业部科学技术研究院西北工业大学分院副院长. 目前研究领域: 大系统理论, 估计理论在石油勘探、飞行器中的应用, 飞行器的并行仿真, 神经网络与神经网络计算机, 工业智能控制系统研究, 复杂系统仿真技术研究, 容错控制系统研究等.

**张洪才** 1938年生. 教授. 1961年毕业于西北工业大学自动控制系, 1964年西北工业大学研究生毕业. 1981年至1983年去美国威斯康星-麦迪逊大学电气及计算机工程系作访问学者, 现为西北工业大学自动控制系教授. 主要研究方向为估计理论, 系统辨识, 随机控制等.