

用于 MIMO-OFDM 系统 QR 分解的分布式脉动阵列处理算法

朱勇旭* 吴斌 周玉梅 蔡菁菁 夏凯锋

(中国科学院微电子研究所 北京 100029)

摘要: 针对多载波系统中信道矩阵QR(正交三角矩阵)分解的延时问题, 该文提出适用于MIMO-OFDM系统QR分解的分布式脉动阵列处理(Distributed Systolic Array Processing, DSAP)算法。该算法包含两种处理机制, 一是交织预处理, 对不同子载波信道矩阵行矢量进行分组交织处理, 按照延时递增规律将每列信道矩阵元素读出并输入到脉动阵列; 二是分布式脉动阵列计算, 通过脉动阵列边界单元和内部单元中流水线CORDIC计算和子载波同步处理实现信道矩阵QR分解分布式处理, 实现不同子载波QR分解分布于脉动阵列边界单元和内部单元中CORDIC不同级。与串行脉动阵列处理(Serial Systolic Array Processing, SSAP)算法比, DSAP算法充分利用时钟周期, 分解延时约为SSAP算法的8%, 有效减少数据处理延时, 而复杂度几乎没有增加。

关键词: 无线通信; QR(正交三角矩阵)分解; 分布式脉动阵列处理; MIMO-OFDM; 低延时

中图分类号: TN92

文献标识码: A

文章编号: 1009-5896(2012)08-1968-06

DOI: 10.3724/SP.J.1146.2011.00768

A Distributed Systolic Array Processing Algorithm for QR-decomposition in MIMO-OFDM System

Zhu Yong-xu Wu Bin Zhou Yu-mei Cai Jing-jing Xia Kai-feng

(Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China)

Abstract: To reduce the delay of Quadrature tRiangle (QR)-decomposition in the multi-carrier systems, a Distributed Systolic Array Processing (DSAP) algorithm for MIMO-OFDM system is proposed. The algorithm includes two processing mechanism: the first is the interleaved pre-processing, which groups and interleaves the row vectors of different channel matrix, and then the matrix elements are sent into the systolic array in columns according to the delay increasing pattern; the other is the signal processing of the distributed systolic array, which uses the CORDIC computation in the boundary and internal cells of systolic array and the synchronization operation, can distribute the QR-decomposition of different sub-carriers into the different stages of the pipelining operation of CORDIC in systolic array. Compared to Serial Systolic Array Processing (SSAP) algorithm, the clock periods can be put to great use in the DSAP algorithm, the delay is reduced by ninety-two percent with the same complexity, and it reduces the delay of subsequent data processing effectively.

Key words: Wireless communication; Quadrature tRiangle (QR)-decomposition; Distributed Systolic Array Processing (DSAP); MIMO-OFDM; Low-delay

1 引言

多输入多输出(Multiple Input Multiple Output, MIMO)和正交频分复用(Orthogonal Frequency Division Multiplexing, OFDM)被认为是提高吞吐率和增强传输可靠性的有效技术, 并应用在诸多通信标准中, 如IEEE 802.11n/ac/ad, 802.16e/m, LTE等。MIMO检测器作为MIMO通信系统核心模块, 其检测性能及复杂度很大程度影响了MIMO通信链路性能及实现复杂度。通过QR(正交三角矩阵)

分解可将信道矩阵 H 分解为正交酉矩阵 Q 和上三角矩阵 R 的乘积, 对接收信号矢量 Y 左乘 Q^H 后, 可将频域等效模型转换为以 R 矩阵为基础自下而上的上三角频域模型, 减少了MIMO检测器信号检测的复杂度。如在QR-ZF, QR-MMSE, QR-SIC和QR-OSIC等线性和连续干扰消除检测算法中^[1,2], 通过QR分解预处理避免了复杂的伪逆计算。基于搜索策略的球形检测和K-Best检测算法等利用QR分解将检测问题转换为树形结构的星座点搜索, 减少了搜索空间大小, 取得性能和复杂度的平衡^[3,4]。

QR分解主要有3种实现方法: 第一是采用修正格拉姆-施密特(Modified Gram-Schmidt, MGS)正交化方法^[5], 其中包含平方、开方、除法等复杂运

2011-07-28收到, 2012-05-31改回

国家自然科学基金(60976022), 国家重大科技专项(2010ZX03005-001)和北京科技新星(392010B060)资助课题

*通信作者: 朱勇旭 zhuyongxu1983@163.com

算，存在扩展性不强，不易于多载波系统信道系数并行处理；第二是采用 Householder 变换算法，复杂度过高，一般不做为实现考虑对象；第三是采用 Givens 旋转实现 QR 分解^[6]，基于协调旋转数字计算 (COordinate Rotation DIgital Computer, CORDIC)运算实现 Givens 旋转的脉动阵列是 QR 分解常用实现结构，无需复杂运算，具有结构规则和扩展性强等特点，但脉动阵列应用于多载波信道系数分解时^[7]，传统串行输入方式按照子载波先后顺序进行 QR 分解处理，这样会因等待造成很大延时，延时随子载波数线性增加，不适合高速通信系统设计需求。

因此，本文提出了一种分布式脉动阵列处理 (Distributed Systolic Array Processing, DSAP)算法，可实现多载波信道矩阵 QR 分解在脉动阵列中分布式处理，很大程度减少处理延时，而复杂度几乎没有增加。

2 基于 Givens 旋转的 QR 分解原理

MIMO-OFDM 系统频域复数模型可表示为

$$\mathbf{r}(k) = \mathbf{h}(k)\mathbf{s}(k) + \mathbf{n}(k) \quad (1)$$

其中 $\mathbf{s}(k)$ 为发送复数信号矢量， $\mathbf{r}(k)$ 为接收复数信号矢量， $\mathbf{h}(k)$ 是复数信道矩阵， $\mathbf{n}(k)$ 是复数白噪声矢量， k 为子载波序号。

可进一步将式(1)表示为实数域形式(2)。

$$\begin{bmatrix} \Re(\mathbf{r}(k)) \\ \Im(\mathbf{r}(k)) \end{bmatrix} = \begin{bmatrix} \Re(\mathbf{h}(k)) & -\Im(\mathbf{h}(k)) \\ \Im(\mathbf{h}(k)) & \Re(\mathbf{h}(k)) \end{bmatrix} \cdot \begin{bmatrix} \Re(\mathbf{s}(k)) \\ \Im(\mathbf{s}(k)) \end{bmatrix} + \begin{bmatrix} \Re(\mathbf{n}(k)) \\ \Im(\mathbf{n}(k)) \end{bmatrix} \quad (2)$$

等效写为

$$\mathbf{Y}(k) = \mathbf{H}(k)\mathbf{X}(k) + \mathbf{N}(k) \quad (3)$$

其中 \Re 和 \Im 指取实部和虚部， $\mathbf{H}(k)$ ， $\mathbf{X}(k)$ ， $\mathbf{Y}(k)$ ， $\mathbf{N}(k)$ 分别是实数域的信道矩阵、发送矢量、接收矢量、白噪声。

下面以具体例子说明基于 Givens 旋转的 QR 分解处理流程，以 3×3 实数信道矩阵为例，忽略噪声

和子载波号，发射信号矢量和接收信号矢量之间关系如式(4)，并展开为式(5)。

$$\mathbf{Y} = \mathbf{H}\mathbf{X} \quad (4)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (5)$$

数学上对矩阵 \mathbf{H} 进行 QR 分解得到正交酉矩阵 \mathbf{Q} 和上三角矩阵 \mathbf{R} ， \mathbf{Q} 满足 $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$ ， \mathbf{I} 为单位阵，其相互关系和表达式如下：

$$\mathbf{H} = \mathbf{Q}\mathbf{R} \quad (6)$$

$$\mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (7)$$

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} \quad (8)$$

如图 1 所示，是基于 Givens 旋转的 QR 处理流程，通过 3 次对矩阵正交旋转操作来完成矩阵元素清零，实现矩阵 QR 分解。第 1 次选择旋转矩阵 \mathbf{G}_1 对 h_{31} 清零，通过类似的 \mathbf{G}_2 矩阵对 $h_{21}^{(1)}$ 清零和 \mathbf{G}_3 矩阵对 $h_{32}^{(2)}$ 清零，得到上三角矩阵 $\mathbf{G}_3\mathbf{G}_2\mathbf{G}_1\mathbf{H}$ 即是 \mathbf{R} 矩阵

$$\mathbf{R} = \mathbf{G}_3\mathbf{G}_2\mathbf{G}_1\mathbf{H} \quad (9)$$

而 \mathbf{G}_1 ， \mathbf{G}_2 和 \mathbf{G}_3 是正交酉矩阵，其逆矩阵等于其转置矩阵，可有

$$\mathbf{H} = (\mathbf{G}_3\mathbf{G}_2\mathbf{G}_1)^H \mathbf{R} \quad (10)$$

$$\mathbf{Q} = (\mathbf{G}_3\mathbf{G}_2\mathbf{G}_1)^H \quad (11)$$

3 脉动阵列处理

采用 CORDIC 运算实现 Givens 旋转的脉动阵列是实现 QR 分解的常用结构，图 2 是 3×3 信道矩阵 QR 分解的脉动阵列，其中圆形是边界单元，正方形是内部单元，边界单元工作于 CORDIC 运算向量模式，将信道元素组成的向量旋转至 x 轴，实现元素清零并向右输出旋转角度，内部单元工作于

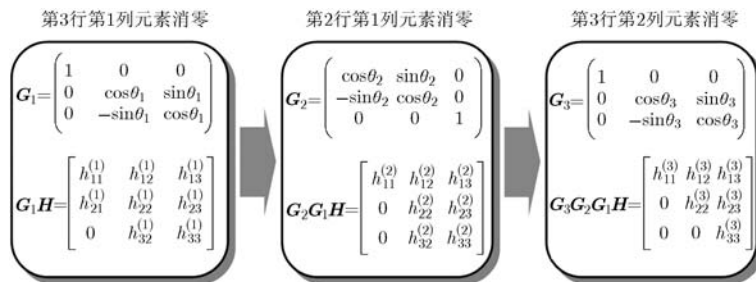


图 1 基于 Givens 旋转的 QR 分解处理流程

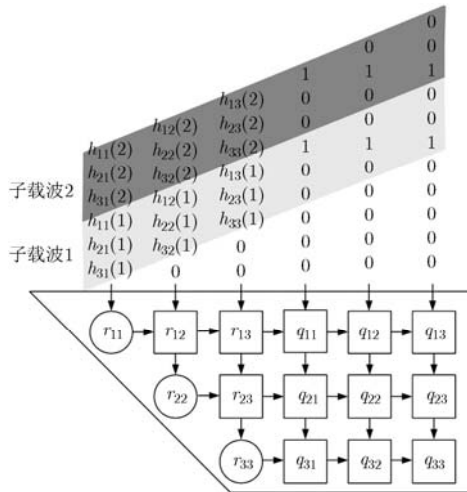


图 2 3×3 信道矩阵 QR 分解的脉动阵列

CORDIC 运算旋转模式，根据输入角度对向量进行旋转，并向下输出旋转后向量虚部和向右传递输入角度。边界单元和内部单元采用同级数的 CORDIC 运算，为实现分解的同步处理，后列输入相对前列输入需要延时 CORDIC 级数个周期，左边的 3×3 上三角阵列用于计算 R 矩阵，其输入是列延时后的信道矩阵，右边的 3×3 方阵用于计算 Q 矩阵，其输入是列延时后的单位阵。

脉动阵列应用到多载波 OFDM 系统时，传统的按子载波先后次序串行输入方式会造成很大的分解延时。文献[7]中采用了串行脉动阵列处理 (Serial Systolic Array Processing, SSAP) 算法，下一子载波必须等到前一子载波分解完毕才能输入，造成很大延时，其延时随子载波数线性增加，在 80 MHz 工作频率下对 52 个子载波信道系数进行 QR 分解的延时是 57.2 μs ，即 4576 个时钟周期，如此大的延时将影响系统延时和数据吞吐率。

4 分布式脉动阵列处理算法

从上述讨论可知，脉动阵列应用到多载波系统的关键问题是如何在不增加复杂度的情况下有效减少处理延时。对此，本文提出了一种分布式脉动阵列处理 (Distributed Systolic Array Processing, DSAP) 算法，其包含两种处理机制：交织预处理和分布式脉动阵列计算，其中交织预处理为后者提供分布式计算所需的信道系数输入形式。

设实数信道矩阵维数是 $M \times M$ ，那么对应脉动阵列中左边 M 列是 $M \times M$ 的上三角矩阵，用来计算 R 矩阵，右边 M 列是 $M \times M$ 的矩阵，用来计算 Q 矩阵，子载波数是 S ，脉动阵列中边界单元和内部单元中的 CORDIC 运算的流水线级数是 T 。

4.1 交织预处理

(1) 以 T 为基数对 S 个子载波信道矩阵构成的序列 $\{\mathbf{H}(1), \mathbf{H}(2), \dots, \mathbf{H}(S)\}$ 按子载波号从小到大进行分组 (为方便，设 S 是 T 的倍数)，则可以划分为 $W = \lceil S/T \rceil$ 组 ($\lceil \cdot \rceil$ 表示上取整)，每组包含 T 个子载波，每组信道矩阵序列表示为 $\{\mathbf{H}_w^1, \mathbf{H}_w^2, \dots, \mathbf{H}_w^T\}$ ，其中 \mathbf{H}_w^t 表示第 w 组第 t 个子载波的信道矩阵 ($t \in \{1, 2, \dots, T\}$) 且 $\mathbf{H}_w^t = \mathbf{H}((w-1)T+t)$ ，用 $\mathbf{H}_w^t(i, j)$ 表示 \mathbf{H}_w^t 中第 i 行 j 列元素，将每组信道矩阵 $\{\mathbf{H}_w^1, \mathbf{H}_w^2, \dots, \mathbf{H}_w^T\}$ 按式 (12) 交织方式填入维数为 $MT \times M$ 的矩阵 \mathbf{P}_w 中，矩阵 \mathbf{P}_w 中第 i 行 j 列元素 $\mathbf{P}_w(i, j)$ 可表示为如下，事实上 $\mathbf{P}_w(i, j)$ 对应的是子载波号为 $(w-1)T + \text{mod}(i-1, T) + 1$ 的信道矩阵 $\mathbf{H}((w-1)T + \text{mod}(i-1, T) + 1)$ 中第 $(\lfloor (i-1, T) \rfloor + 1)$ 行 j 列元素，

$$\mathbf{P}_w(i, j) = \mathbf{H}_w^{\text{mod}(i-1, T)+1}(\lfloor (i-1, T) \rfloor + 1, j) \quad (12)$$

其中， $\text{mod}(i-1, T)$ 表示 $i-1$ 对 T 取模， $\lfloor (i-1, T) \rfloor$ 表示 $i-1$ 对 T 下取整。

按式 (12) 所示的交织方式将每组信道系数 $\{\mathbf{H}_w^1, \mathbf{H}_w^2, \dots, \mathbf{H}_w^T\}$ 映射到 $MT \times M$ 维的矩阵 \mathbf{P}_w 中。

(2) 取 $M \times M$ 维的单位阵 \mathbf{I}_w^t 时形成单位矩阵序列 $\{\mathbf{I}_w^1, \mathbf{I}_w^2, \dots, \mathbf{I}_w^T\}$ ，并采用与 (1) 相同交织处理形成 $MT \times M$ 维的矩阵 \mathbf{E}_w ($w \in \{1, 2, \dots, W\}$)。

(3) 将 \mathbf{P}_w 和 \mathbf{E}_w 按照行组成 $MT \times 2M$ 维矩阵 $[\mathbf{P}_w, \mathbf{E}_w]$ 。将 W 组 $MT \times 2M$ 维的矩阵 $[\mathbf{P}_w, \mathbf{E}_w]$ 按列组合在一起，形成 $WMT \times 2M$ 维的矩阵 $[\mathbf{P}, \mathbf{E}]$ ，其中 $\mathbf{P}(i, j)$ 和 $\mathbf{E}(i, j)$ 是 \mathbf{P} 和 \mathbf{E} 矩阵的第 i 行 j 列元素， $WMT \times M$ 维 \mathbf{P} 矩阵包含了所有 S 个子载波信道系数元素， $\mathbf{P}(i, j)$ 的值是子载波号为 $(\lfloor (i-1, MT) \rfloor)T + \text{mod}(i-1, T) + 1$ 的 $M \times M$ 维信道矩阵的第 $\text{mod}(\lfloor (i-1, T) \rfloor, M) + 1$ 行 j 列元素， $\mathbf{E}(i, j)$ 的值是 $M \times M$ 维单位矩阵的第 $\text{mod}(\lfloor (i-1, T) \rfloor, M) + 1$ 行 j 列元素。

(4) 将信道矩阵交织处理后得到的矩阵 $[\mathbf{P}, \mathbf{E}]$ 按照每列延时 T 流水线输入脉动阵列中，如图 3 所示。交织预处理实现了不同子载波信道矩阵行矢量分组交织处理，按照延时递增规律将每列信道矩阵元素读出并输入到脉动阵列。

4.2 分布式脉动阵列计算

分布式脉动阵列整体结构图与图 2 类似，由 R 阵列和 Q 阵列构成，分别用来计算 QR 分解的 R 矩阵和 Q 矩阵。交织预处理后的信道系数输入脉动阵列中，通过脉动阵列中边界单元和内部单元 CORDIC 流水线运算和子载波同步处理来实现信道系数 QR 分解的分布式计算。

(1) 边界单元 CORDIC 流水线计算：脉动阵列

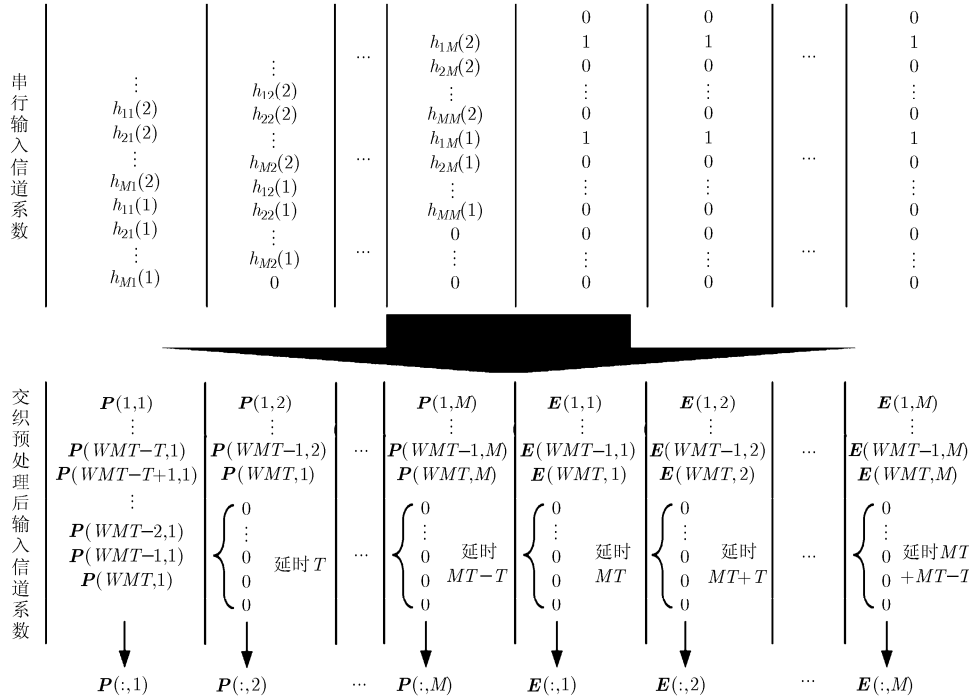


图 3 交织预处理后输入信道系数

中圆形单元，工作于 CORDIC 向量模式，采用流水线计算，将输入 y_{in} 与边界单元内部存储值 x_{in} 组成向量 (x_{in}, y_{in}) 旋转到 x 轴，向右输出旋转角度 $\theta = \arctan(y_{in}/x_{in})$ ，并用 $\sqrt{x_{in}^2 + y_{in}^2}$ 更新内部存储器，运算过程采用式(13)所示的 T 级流水线迭代逼近处理，初始 $\theta_1 = 0^\circ$ ， $x_1 = x_{in}$ ， $y_1 = y_{in}$ ， T 级流水线处理后，得到的 $\theta_{out} = \theta_{T+1} \approx \arctan(y_{in}/x_{in})$ ， $y_{T+1} \approx 0$ ， $x_{T+1} \approx \sqrt{x_{in}^2 + y_{in}^2}$ ，输出角度 $\theta_{out} = \theta_{T+1}$ ，并用 x_{T+1} 更新内部存储值。

$$\left. \begin{aligned} x_{i+1} &= x_i - y_i \cdot d_i \cdot 2^{-i} \\ y_{i+1} &= y_i + x_i \cdot d_i \cdot 2^{-i} \\ \theta_{i+1} &= \theta_i - d_i \cdot \tan^{-1}(2^{-i}) \end{aligned} \right\}, \quad d_i = \begin{cases} 1, & y_i < 0 \\ -1, & \text{其它} \end{cases} \quad (13)$$

(2)内部单元 CORDIC 流水线计算：脉动阵列中正方形单元，工作于 CORDIC 旋转模式，采用流水线计算，将输入的 y_{in} 与内部存储值 x_{in} 组成向量 (x_{in}, y_{in}) 旋转输入角度 θ ，得到旋转后向量为 $(x_{in} \cos \theta + y_{in} \sin \theta, -x_{in} \sin \theta + y_{in} \cos \theta)$ ，同时向下输出 $-x_{in} \sin \theta + y_{in} \cos \theta$ 和向右输出输入角度 θ ，并用 $x_{in} \cos \theta + y_{in} \sin \theta$ 更新内部存储值，实际运算过程采用式(14)所示的 T 级流水线迭代逼近处理，初始 $\theta_1 = \theta$ ， $x_1 = x_{in}$ ， $y_1 = y_{in}$ ， T 级流水线处理后，得到 $\theta_{T+1} \approx 0$ ， $y_{T+1} \approx -x_{in} \sin \theta + y_{in} \cos \theta$ ， $x_{T+1} \approx x_{in} \cos \theta + y_{in} \sin \theta$ ，向下输出 y_{T+1} 和向右输出输入角度 θ ，并用 x_{T+1} 更新内部存储值。

$$\left. \begin{aligned} x_{i+1} &= x_i - y_i \cdot d_i \cdot 2^{-i} \\ y_{i+1} &= y_i + x_i \cdot d_i \cdot 2^{-i} \\ \theta_{i+1} &= \theta_i - d_i \cdot \tan^{-1}(2^{-i}) \end{aligned} \right\}, \quad d_i = \begin{cases} -1, & \theta_i < 0 \\ 1, & \text{其它} \end{cases} \quad (14)$$

(3)子载波同步处理：当 $P(:,1)$ 中采用流水线输入脉动阵列的第 1 列， $\{P(WMT,1), P(WMT-1,1), \dots, P(WMT-T+1,1)\}$ 是前 T 周期输入的值，分别是第 W 组信道矩阵序列(子载波号是 S 到 $S-T+1$)中每个信道矩阵的第 M 行 1 列元素，由于脉动阵列中边界单元和内部单元均是采用了 T 级 CORDIC 流水线运算，第 1 个元素 $P(WMT,1)$ 输入后，在边界单元 r_{11} 中需经过 T 周期 CORDIC 流水线处理才能得到向量 $(0, P(WMT,1))$ (内部存储器值初始时为 0)旋转到 x 轴的角度 θ ，这样连续 T 周期输入 $\{P(WMT,1), P(WMT-1,1), \dots, P(WMT-T+1,1)\}$ 将分布于边界单元 r_{11} 流水线的不同级，并分别流水线输出消零处理后的旋转角度，此时 $P(:,2)$ 也采用流水线输入， $P(:,2)$ 前 T 周期输入值 $\{P(WMT,2), P(WMT-1,2), \dots, P(WMT-T+1,2)\}$ 分别是第 W 组信道矩阵序列(子载波号是 S 到 $S-T+1$)中每个信道矩阵的第 M 行 2 列元素， $P(:,2)$ 延时 T 输入时，正好与 r_{11} 中进行消零处理后输出旋转角度的子载波号对应上，保证了子载波信道系数 QR 分解运算同步性。脉动阵列中其余的边界单元和内部单元均采用类似的处理方式，直到 $[P, E]$ 矩阵中 $E(1, M)$ 元素输入并经过内部单元 q_{1M}

旋转处理后,所有子载波 QR 分解运算处理完毕。不同子载波在脉动阵列中进行信道系数 QR 分解实现了完全同步,不同子载波的处理分布在脉动阵列边界单元和内部单元 CORDIC 运算的不同级,充分利用了时间周期。

5 结果与分析

(1)DSAP 算法的时间效率分析: DSAP 算法通过对子载波信道系数交织预处理和分布式脉动阵列计算,利用脉动阵列中 CORDIC 的流水线特征,将不同子载波信道系数 QR 分解分布于 CORDIC 运算的不同级,保证了每个子载波信道系数进行 QR 分解同步处理,充分利用了时钟周期,时间效率高。

如图 3 下部分所示,交织预处理后的信道系数流水线输入脉动阵列,当最晚输入脉动阵列元素 $E(1, M)$ 在内部单元 q_{1M} 中经过 CORDIC 流水线计算后,即完成所有子载波信道系数 QR 分解计算,通过前面分析可得 S 个 $M \times M$ 维信道矩阵 QR 分解的延时为 $2MT + WMT$, 其中 W 是将 S 子载波以 CORDIC 流水线级数 T 为基数进行分组得到的组数, $W = \lceil S/T \rceil$, 延时可表示为 $MT(2 + \lceil S/T \rceil)$ 。而 SSAP 算法按照子载波先后次序将信道系数输入脉动阵列,如图 3 上部分所示,下一子载波的 QR 分解必须等待当前子载波处理完成,所以 S 个 $M \times M$ 维信道矩阵 QR 分解的延时约为 MTS , 其延时随子载波数线性增加。两者的延时之比是 $(2 + \lceil S/T \rceil)/S$ 。在实现中,一般 CORDIC 运算迭代次数为 12~14 即可逼近理想值。图 4 曲线分别对应了迭代次数为 12~14 时两种算法的延时比关系,当子载波数较多时,延时比趋近 $1/T$, 约为 8% ($T = 12$), 同时与 CORDIC 级数成反比。

随着子载波数增加,子载波等待造成 SSAP 算法的延时随子载波数线性增加,而 DSAP 算法将不同子载波 QR 分解分布于 CORDIC 流水线不同级,无需等待,在子载波数较大时两种算法的延时比约

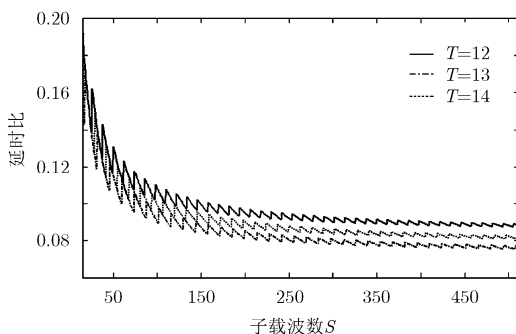


图 4 不同 CORDIC 级数下延时比随子载波数变化图

为 8% ($T = 12$), 可见,采用 DSAP 算法进行多载波信道矩阵 QR 分解的延时可得到有效减小,在延时上的优势避免或减小了对数据进行缓存等待处理,可减小检测延时及缓存器开销。

(2)DSAP 算法的复杂度分析:以 4×4 实数矩阵 QR 分解为例 ($N = 4$), 表 1 是各种分解算法计算复杂度^[5]。相比 Gram-Schmidt, Householder 和传统 Givens Rotation 算法,采用 CORDIC 运算实现 Givens 旋转的脉动阵列的计算逻辑是 CORDIC 单元(由加减法和移位运算组成),避免了大量的乘法、开方和除法运算,具有低的计算复杂度,同时脉动阵列结构规则,易于天线扩展。SSAP-GR 和 DSAP-GR 算法在 CORDIC 使用个数上是一样的,区别在于 DSAP-GR 需要增加少量控制,用于子载波在脉动阵列中分布计算的同步处理。

(3)与相关文献结果的比较:表 2 是本文提出的 DSAP 算法和文献中 QR 分解算法的延时大小比较。对 DSAP 算法,设发射天线和接收天线是 4, CORDIC 级数取 13,子载波数 52 时,对应参数 $S = 52$, $M = 8$, $T = 13$, 则进行 52 个子载波处理延时是 $MT(2 + \lceil S/T \rceil) = 624$ 时钟周期,而文献[7]中采用了 SSAP 算法进行 52 个子载波信道矩阵 QR 分解,其延时是 $57.2 \mu s \times 80 \text{ MHz} = 4576$ 时钟周期。为增强对比性,采用平均延时来衡量各种算法,平均延时=延时/QR 分解个数,在平均延时上,DSAP 算法充分利用了每一个时钟周期,脉动阵列中 CORDIC 运算的使用效率达到了 100%,相比文献中的平均延时大小,DSAP 可获得较好的平均处理延时。

6 结束语

在 MIMO 检测器中,对信道矩阵进行 QR 矩阵分解是常用的减小运算复杂度或简化搜索空间的有效途径。基于 Givens 旋转的脉动阵列是实现 QR 分解的一种常用结构,具有复杂度低、结构规则和易于扩展等优点。但是,在多载波信道系数 QR 分解时,延时成为传统的串行脉动阵列处理 (Serial Systolic Array Processing, SSAP) 算法的一个瓶颈。本文提出了分布式脉动阵列处理 (Distributed Systolic Array Processing, DSAP) 算法,通过对多载波信道矩阵进行交织预处理和分布式脉动阵列计算,能将不同子载波的 QR 分解分布于脉动阵列中 CORDIC 流水线运算的不同级,充分利用了时钟周期,减少了多载波 QR 分解延时。与 SSAP 算法相比,DSAP 算法延时约为其 8% ($T = 12$), 而复杂度几乎没有增加。

表 1 各种 QR 分解复杂度分析

算法	乘法	加/减	开方	除法	CORDIC
Gram-Schmidt ^[5]	$4N^2$	$4N^2 - 3N$	N	N^2	0
Householder ^[5]	$O(N^4)$	$O(N^4)$	N	$O(N^3)$	0
传统 Givens Rotation ^[5]	$2N^2 + 6N - 8$	$N^2 + 3N - 4$	$N - 1$	$2N - 2$	$N^2/2 + 3N/2 - 2$
SSAP-GR	0	0	0	0	$3N^2/2 + N/2$
DSAP-GR	0	0	0	0	$3N^2/2 + N/2$

表 2 本文算法与不同文献中算法的比较

	文献[7]	文献[8]	文献[9]	文献[10]	文献[11]	本文算法
算法	SSAP-GR	MGS	GR	GR	GR	DSAP
天线数	4×4	4×4	2×2	4×4	4×4	4×4
QR 分解个数	52	1	1	52	52	52
CORDIC 级数	13	-	18	8	18	13
延时(周期)	4576	35	54	704	702	624
平均延时(周期)	88	35	54	14	14	12

参 考 文 献

- [1] Chiu P L, Huang L Z, Chai L W, *et al.* A 684 Mbps 57 mW joint QR decomposition and MIMO processor for 4×4 MIMO-OFDM systems[C]. IEEE Asian Solid State Circuits Conference(ASSCC), Jeju, Korea, Nov. 14-16, 2011: 309-312.
- [2] Ramakrishnan V, Veerkamp T, Ascheid G, *et al.* Implementations of sorted-QR decomposition for MIMO receivers: complexity, reusability and efficiency analysis[J]. *Journal of Signal Processing Systems*, 2011, 69(3): 1-13.
- [3] Kim T H and Park I C. Small-area and low-energy K-Best MIMO detector using relaxed tree expansion and early forwarding[J]. *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 2010, 57(10): 2753-2761.
- [4] Shen C A and Eltawil A M. A radius adaptive K-Best decoder with early termination: algorithm and VLSI architecture [J]. *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 2010, 57(9): 2476-2486.
- [5] Huang Z Y and Tsai P Y. Efficient implementation of QR decomposition for gigabit MIMO-OFDM systems[J]. *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 2011, 58(10): 2531-2542.
- [6] Ma L, Dickson K, and McAllister J. QR decomposition-based matrix inversion for high performance embedded MIMO receivers[J]. *IEEE Transactions on Signal Processing*, 2011, 59(4): 1858-1867.
- [7] Haene S, Perels D, and Burg A. A real-time 4-stream MIMO-OFDM transceiver: system design, FPGA implementation, and characterization[J]. *IEEE Journal on Selected Areas in Communications*, 2008, 26(6): 877-889.
- [8] Chang R, Lin C, Lin K, *et al.* Iterative QR decomposition architecture using the modified Gram-Schmidt algorithm for MIMO systems[J]. *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 2010, 57(5): 1095-1102.
- [9] Chen D D and Sima M. Fixed-point CORDIC-based QR decomposition by Givens rotations on FPGA[C]. International Conference on Reconfigurable Computing and FPGAs (ReConFig), Cancun, Mexico, 2011: 327-332.
- [10] Hwang Y T and Chen W D. Design and implementation of a high-throughput fully parallel complex-valued QR factorisation chips[J]. *IET Circuits Devices Systems*, 2011, 5(5): 424-432.
- [11] Josep S G, Milford D, Sandell M, *et al.* Implementation and evaluation of a high-performance MIMO detector for wireless LAN systems[J]. *IEEE Transactions on Consumer Electronics*, 2011, 57(4): 1519-1527.
- 朱勇旭：男，1983年生，助理研究员，博士，研究方向为 MIMO-OFDM 的宽带无线通信系统物理层算法。
- 吴 斌：男，1976年生，副研究员，博士，研究方向为宽带通 SOC VLSI 体系架构设计与实现。
- 周玉梅：女，1962年生，研究员，博士生导师，研究方向为集成电路设计技术。