

固定样本数目的 QR 分解递推算法

倪淑燕¹, 程乃平¹, 倪正中²

(1. 装备指挥技术学院, 北京 101416; 2. 遥感信息研究所, 北京 100192)

摘 要: 自适应波束形成中基于 QR 分解的递推算法大都在样本不断累积下推出, 即每次快拍增加一个样本。实际中可利用的样本不可能无限多, 在到达一定样本数目后, 每增加一个新样本的同时需要剔除一个旧样本。针对这种样本数目固定的数据更新方式, 利用双曲 Householder 变换, 提出了一种更实用的 QR 分解递推算法, 并对其进行了简化, 大大减小了运算量; 之后利用逆 QR 分解的思想对其进行了进一步改进, 使算法更利于系统的实时实现; 在此基础上研究了更为稳健的对角加载逆 QR 分解的递推实现方式。计算机仿真证明, 在有限样本情况下, 本算法比常规 QR 分解算法具有更高的阵增益和更好的波束性能。

关键词: 阵列信号处理; 波束形成; QR 分解; 有限快拍

中图分类号: TN911.7

文献标识码: A

文章编号: 1000-436X(2010)8A-0195-06

Recursive algorithm based on QR decomposition for fixed sample size

NI Shu-yan¹, CHENG Nai-ping¹, NI Zheng-zhong²

(1. Academy of Equipment Command and Technology, Beijing 101416, China;

2. Remote sensing Information Research Institute, Beijing 100192, China)

Abstract: Most of recursive algorithms based on QR decomposition(QRD) for adaptive beamforming are deduced by assuming that at each incoming snapshot an additional data vector is added. However, in practical use, the sample size which can be used is always limited. After the sample data are cumulated to a certain amount, an old data vector should be removed at the same time when a new data vector is added. For this way of data updating, a more practical recursive QRD algorithm was proposed utilizing hyperbolic householder transformation. To reduce the computational cost of the algorithm a simplified method was also studied. Then based on the theory of the inverse QRD, further improvements were made for the algorithm more fit for real time use. Finally, the recursive algorithm was applied to the diagonal loading beamformer which had more robust capabilities. Several computer simulations illustrate that, when the sample size is limited, the proposed algorithms provide large array gain and better beam pattern than the traditional QRD algorithm.

Key words: array signal processing; beamforming; QR decomposition; finite snapshots

1 引言

自适应波束形成是信号处理领域的一个重要分支, 它广泛地应用于雷达、声纳和通信等多种领域。经过几十年的发展, 其基本理论已经成熟, 研究热点转到了其工程实现上。常规的采样矩阵求逆算法(SMI)属于均方域算法, 需要显示计算采样

协方差矩阵, 相当于对采样数据进行平方运算, 使得许多实际情况下协方差矩阵出现病态, 严重影响算法的数值特性, 更为重要的是, 这种实现方法不利于系统的并行实现, 这大大限制了其工程应用。基于均方域算法的不足, 基于 QR 分解的数据域算法得到了越来越多的关注, 它通过对输入数据矩阵做 QR 分解完成协方差矩阵的估计, 进而求得权向

量, 具有更好的数值特性和固有的高度并行性, 因此在实际的各种系统中得到了广泛的应用。

但是, 目前自适应波束形成中大部分 QR 分解递推算法是在样本数不断增加的情况下得出的^[1-3], 而实际中可以利用的有效样本数 K 通常是有限的, 要采用常规的 QR 分解递推算法, 在每积累 K 次样本后, 由于不能继续积累, 需要重新从一个样本开始计算, 这降低了样本数据的利用率。有些文献加入了遗忘因子以减小过去数据的影响^[4], 但仍然不适用于实际应用。实际中比较常用的数据更新方式是滑窗方式, 即每增加一个新样本的同时删除一个旧样本^[5], 保持所用样本数目不变。实现滑窗 QR 分解的方法主要有 2 种: 一种是采用两步 Givens 旋转^[6], 这种方法引入了不必要的重复步骤, 减低了运算效率^[7]; 另一种是采用双曲 Householder 变换^[8]。本文将滑窗 QR 分解方法应用到自适应波束形成的权向量求解中, 利用双曲 Householder 变换, 提出一种固定样本数目的 QR 分解递推算法, 并利用逆 QR 分解的思想对其进行了改进, 在此基础上研究了更为稳健的对角加载逆 QR 分解的递推实现方式。提出的方法不仅具有更稳定的数值特性, 还大大提高了运算速度, 更适用于实际应用。

2 QR 分解算法的基本原理

考虑阵元数为 M 、阵元间距为 $\lambda/2$ 的均匀线阵, 第 $n(n \geq M)$ 次快拍后得到的 $M \times n$ 维输入数据矩阵为

$$\begin{aligned} \mathbf{X}_n &= [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(n)] \\ \mathbf{x}(i) &= [x_1(i), x_2(i), \dots, x_M(i)]^T, \quad i=1, 2, \dots, n \end{aligned} \quad (1)$$

其中, $x_m(i) (m=1, \dots, M)$ 表示第 m 个阵元在第 i 次快拍时接收的样本数据。则输入数据协方差矩阵可以表示为

$$\mathbf{R}_x = \frac{1}{n} \mathbf{X}_n \mathbf{X}_n^H \quad (2)$$

设期望信号的导向矢量为 \mathbf{a}_s , 则 SMI 算法的最优权可以表示为

$$\mathbf{w} = \mu \mathbf{R}_x^{-1} \mathbf{a}_s \quad (3)$$

其中, μ 为任意比例常数, 不妨令 $\mu=1/n$, 用 \mathbf{w}_n 表示第 n 次快拍时的自适应权值, 则有:

$$\mathbf{X}_n \mathbf{X}_n^H \mathbf{w}_n = \mathbf{a}_s \quad (4)$$

用 $n \times n$ 维酉矩阵 \mathbf{Q}_n 对 \mathbf{X}_n^H 进行 QR 分解, 即:

$$\mathbf{Q}_n \mathbf{X}_n^H = \begin{bmatrix} \mathbf{A}_n \\ \mathbf{0} \end{bmatrix} \quad (5)$$

其中, \mathbf{A}_n 为 $M \times M$ 维上三角阵, $\mathbf{0}$ 表示零矩阵。则有:

$$\mathbf{X}_n \mathbf{X}_n^H = \mathbf{X}_n \mathbf{Q}_n^H \mathbf{Q}_n \mathbf{X}_n^H = \begin{bmatrix} \mathbf{A}_n^H & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \mathbf{A}_n^H \mathbf{A}_n \quad (6)$$

这样式 (4) 转换为

$$\mathbf{A}_n^H \mathbf{A}_n \mathbf{w}_n = \mathbf{a}_s \quad (7)$$

改写为

$$\begin{cases} \mathbf{A}_n^H \mathbf{v}_n = \mathbf{a}_s \\ \mathbf{A}_n \mathbf{w}_n = \mathbf{v}_n \end{cases} \quad (8)$$

式 (8) 即为基于 QR 分解的 SMI (QRD-SMI) 算法的基本实现公式, 其中 \mathbf{A}_n 可以采用递推方法进行更新, 2 个三角方程可以采用通过前向回代和后向回代法求解, 但前后向回代的硬件和软件开销大, 所以一般采用逆 QR 分解 (IQRD-SMI) 算法求解。在 QRD-SMI 和 IQRD-SMI 算法的递推实现过程中, 都假设了 $\mathbf{X}_n = [\mathbf{X}_{n-1} \ \mathbf{x}(n)]$ 的样本更新方式, 而实际中样本数不可能无限积累, 在到达一定样本数目后, 每增加一个新样本的同时需要剔除一个旧样本, 保证计算数据协方差矩阵用的样本数始终为 K 。

3 固定样本数目的 QR 分解递推算法

3.1 基本原理

固定样本数目的数据更新方式可以用下式表示:

$$\begin{cases} \mathbf{X}_n = [\mathbf{x}(n-K+1), \dots, \mathbf{x}(n-1), \mathbf{x}(n)] \\ \mathbf{R}_x = \mathbf{X}_n \mathbf{X}_n^H / K \\ \mathbf{X}_n \mathbf{X}_n^H = \mathbf{X}_{n-1} \mathbf{X}_{n-1}^H + \mathbf{x}(n) \mathbf{x}^H(n) - \mathbf{x}(n-K) \mathbf{x}^H(n-K) \end{cases} \quad (9)$$

在 $n > K \geq M$ 时, 设在 $n-1$ 时刻已经实现了将 \mathbf{X}_{n-1}^H 上三角化为 \mathbf{A}_{n-1} , 则有 $\mathbf{X}_{n-1} \mathbf{X}_{n-1}^H = \mathbf{A}_{n-1}^H \mathbf{A}_{n-1}$, 因此可得:

$$\mathbf{X}_n \mathbf{X}_n^H = \mathbf{A}_{n-1}^H \mathbf{A}_{n-1} + \mathbf{x}(n) \mathbf{x}^H(n) - \mathbf{x}(n-K) \mathbf{x}^H(n-K) \quad (10)$$

令 $\tilde{\mathbf{X}}_n = \begin{bmatrix} \mathbf{A}_{n-1}^H & \mathbf{x}(n) & \mathbf{x}(n-K) \end{bmatrix}$, $\Phi = \begin{bmatrix} \mathbf{I}_M & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & 0 \\ \mathbf{0} & 0 & -1 \end{bmatrix}$, \mathbf{I} 表

示为单位矩阵, 下标表示其维数, 则有:

$$\mathbf{X}_n \mathbf{X}_n^H = \tilde{\mathbf{X}}_n \Phi \tilde{\mathbf{X}}_n^H \quad (11)$$

若存在一个变换矩阵 $\tilde{\mathbf{Q}}_n$, 满足如下关系:

$$\begin{cases} \tilde{\mathbf{Q}}_n \tilde{\mathbf{X}}_n^H = \begin{bmatrix} \mathbf{A}_n \\ \mathbf{0} \end{bmatrix} \\ \tilde{\mathbf{Q}}_n^H \Phi \tilde{\mathbf{Q}}_n = \Phi \end{cases} \quad (12)$$

由于 A_n 为 $M \times M$ 维，可以得到：

$$\begin{aligned} X_n X_n^H &= \tilde{X}_n \Phi \tilde{X}_n^H = \tilde{X}_n \tilde{Q}_n^H \Phi \tilde{Q}_n \tilde{X}_n^H \\ &= \begin{bmatrix} A_n^H & \mathbf{0} \end{bmatrix} \Phi \begin{bmatrix} A_n \\ \mathbf{0} \end{bmatrix} = A_n^H A_n \end{aligned} \quad (13)$$

这样就利用 A_{n-1} 递推地计算出了 A_n 。可见问题的关键在于寻找满足式 (12) 的变换矩阵 \tilde{Q}_n 。考虑到双曲线 Householder 变换满足式 (12) 的条件^[8]，所以通过一系列的双曲线 Householder 变换来构造矩阵 \tilde{Q}_n ，即

$$\begin{aligned} \tilde{X}_n^{(1)} &= H_1 \tilde{X}_n^H \\ \tilde{X}_n^{(2)} &= H_2 \tilde{X}_n^{(1)} = H_2 H_1 \tilde{X}_n^H \\ &\dots \\ \tilde{X}_n^{(M)} &= H_M \tilde{X}_n^{(M-1)} = H_M \dots H_2 H_1 \tilde{X}_n^H = \tilde{Q}_n \tilde{X}_n^H \end{aligned} \quad (14)$$

其中， H_1, H_2, \dots, H_M 都为 $(M+2) \times (M+2)$ 维双曲线 Householder 变换矩阵， $\tilde{Q}_n = H_M \dots H_2 H_1$ ，得到的 $\tilde{X}_n^{(M)}$ 就是所求的上三角矩阵。 H_i 把 $\tilde{X}_n^{(i-1)}$ 的第 i 列的除前 i 个元素以外的元素消除，而保持前 $i-1$ 列不变，因此 H_i 又可以表示为

$$H_i = \begin{bmatrix} I_{i-1} & \mathbf{0} \\ \mathbf{0} & \tilde{H}_i \end{bmatrix} \quad (15)$$

其中， I_{i-1} 为单位矩阵，下标表示其维数， $\tilde{H}_i = \Phi_i - 2v_i v_i^H / v_i^H \Phi_i v_i$ 为 $L_i \times L_i$ ($L_i = M+3-i$) 维双曲线 Householder 变换矩阵， $\Phi_i = \begin{bmatrix} I_{L_i-2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -1 \end{bmatrix}$ ，

$v_i = \Phi_i u_i + d_i e_{L_i}$ ， u_i 为 $\tilde{X}_n^{(i-1)}$ 的第 i 列的除前 $i-1$ 个元素外的元素组成的 L_i 维列向量， $d_i = \sqrt{u_i^H \Phi_i u_i} / \|u_{i1}\|$ ， u_{i1} 为 u_i 的第 1 个元素， e_{L_i} 表示第 1 个元素为 1，其余为 0 的 L_i 维列向量。

3.2 算法简化

实际上，由 $\tilde{X}_n = [A_{n-1}^H \ x(n) \ x(n-K)]$ 可知， u_i 中只有 3 个非零元素，分别为第 1 个元素 u_{i1} 和最后 2 个元素 u_{i,L_i-1} 、 u_{i,L_i} ，因此 \tilde{H}_i 只会对这 3 个元素对应的行进行变换，省略 \tilde{H}_i 变换时输出保持不变的行，将 u_i 简化为 $u_i' = [u_{i1}, u_{i,L_i-1}, u_{i,L_i}]^T$ ， Φ_i 简化为

$$\begin{aligned} \Phi &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad e_i \text{ 简化为 } e_i = [1 \ 0 \ 0]^T, \text{ 则有:} \\ \left\{ \begin{aligned} d_i' &= \sqrt{(u_i')^H \Phi u_i'} / |u_{i1}| = \sqrt{u_i^H \Phi_i u_i} / |u_{i1}| = d_i \\ v_i' &= \Phi u_i' + d_i e_i = [v_{i1}, v_{i,L_i-1}, v_{i,L_i}]^T \end{aligned} \right. \end{aligned} \quad (16)$$

可见 v_i' 即为 v_i 的 3 个非零元素组成的列向量。

令：

$$\tilde{H}_i' = \Phi - 2v_i'(v_i')^H / (v_i')^H \Phi v_i' \quad (17)$$

用 $\tilde{H}_i'(s,t)$ 表示 \tilde{H}_i' 的第 s 行第 t 列元素，则有：

$$\tilde{H}_i' = \begin{bmatrix} \tilde{H}_i'(1,1) & \mathbf{0} & \tilde{H}_i'(1,2) & \tilde{H}_i'(1,3) \\ \mathbf{0} & I_{L_i-3} & \mathbf{0} & \mathbf{0} \\ \tilde{H}_i'(2,1) & \mathbf{0} & \tilde{H}_i'(2,2) & \tilde{H}_i'(2,3) \\ \tilde{H}_i'(3,1) & \mathbf{0} & \tilde{H}_i'(3,2) & \tilde{H}_i'(3,3) \end{bmatrix} \quad (18)$$

这样通过求解 \tilde{H}_i' 就可以得到了 \tilde{H}_i ，将双曲线 Householder 变换的问题由 L_i 维降为 3 维，大大降低了更新 A_n 的计算量。

把固定样本数目的 QR 分解递推算法记为 MQRD-SMI 算法，实现步骤如下。

步骤 1 初始化：初始状态是在积累够 K 个样本数据时，上三角矩阵 A_K 可以利用 QRD-SMI 递推算法得到。

步骤 2 $n=K+1, K+2, \dots$

1) 更新上三角矩阵 A_n ：设 $\tilde{X}_n^{(0)} = \tilde{X}_n^H$ ，在 $i=1, 2, \dots, M$ 时，利用式 (17) 和式 (18) 得出 \tilde{H}_i' ，将 $\tilde{X}_n^{(i-1)}$ 变为 $\tilde{X}_n^{(i)}$ ，得出的 $\tilde{X}_n^{(M)}$ 即为上三角阵 $\begin{bmatrix} A_n \\ \mathbf{0} \end{bmatrix}$ ；

2) 前向代入求解下三角方程 $A_n^H v_n = a_s$ ，得到中间向量 v_n ；

3) 后向代入求解上三角方程 $A_n w_n = v_n$ ，得到中间向量 w_n 。

对于第 n 次迭代，步骤 2 的 1) 中求解 \tilde{H}_i' 需要 12 次复乘，用 \tilde{H}_i' 对数据进行变换的需要大约 $9(M-i+1)$ ，因此更新上三角阵 A_n 的总计算量约为 $4.5M^2 + 16.5M$ ；2) 和 3) 中求解三角方程的总计算量为 $M(M-1)$ ，这样每次更新自适应权的运算量为 $5.5M^2 + 15.5M$ ，复杂度为 $O(5.5M^2)$ ，低于常规矩阵求逆算法 $O(M^3)$ 的计算复杂度。并且由于不需要显示计算采样协方差矩阵，数值特性远优于常规 SMI 算法。

4 固定样本数目的逆 QR 分解递推算法

4.1 MIQRD-SMI 算法

前后向回代不利用并行计算，硬件和软件开销大^[1]，为了克服 MQRD-SMI 算法中的这个缺点，利用逆 QR 分解的思想对算法做进一步改进，记为 MIQRD-SMI 算法。MIQRD-SMI 算法将自适应权的递推求解问题转换为中间变量 v_n 和下三角矩阵

A_n^{-H} 的更新问题, 即

$$\begin{cases} \mathbf{v}_n = A_n^{-H} \mathbf{a}_s \\ \mathbf{w}_n^H = \mathbf{v}_n^H A_n^{-H} \end{cases} \quad (19)$$

由于 $A_n^H \mathbf{v}_n = A_{n-1}^H \mathbf{v}_{n-1} = \mathbf{a}_s$, 而 $A_n^H \mathbf{v}_n$ 、 $A_{n-1}^H \mathbf{v}_{n-1}$ 又可以分别表示为

$$\begin{cases} A_n^H \mathbf{v}_n = \begin{bmatrix} A_n^H & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}_n \\ \# \end{bmatrix} \\ A_{n-1}^H \mathbf{v}_{n-1} = \begin{bmatrix} A_{n-1}^H & \mathbf{x}(n) & \mathbf{x}(n-K) \end{bmatrix} \Phi \begin{bmatrix} \mathbf{v}_{n-1} \\ \mathbf{0} \end{bmatrix} \\ = \tilde{X}_n \tilde{Q}_n^H \Phi \tilde{Q}_n \begin{bmatrix} \mathbf{v}_{n-1} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} A_n^H & \mathbf{0} \end{bmatrix} \Phi \tilde{Q}_n \begin{bmatrix} \mathbf{v}_{n-1} \\ \mathbf{0} \end{bmatrix} \end{cases} \quad (20)$$

其中 # 表示不关心的部分。因此可得:

$$\Phi \tilde{Q}_n \begin{bmatrix} \mathbf{v}_{n-1} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_n \\ \# \end{bmatrix} \quad (21)$$

由于 $\Phi^{-1} = \Phi$, 对式 (21) 左右两边同乘以 Φ^{-1} 得:

$$\tilde{Q}_n \begin{bmatrix} \mathbf{v}_{n-1} \\ \mathbf{0} \end{bmatrix} = \Phi \begin{bmatrix} \mathbf{v}_n \\ \# \end{bmatrix} = \begin{bmatrix} \mathbf{v}_n \\ \# \end{bmatrix} \quad (22)$$

这样就利用 \tilde{Q}_n 实现了 \mathbf{v}_n 的更新。而对于下三角矩阵 A_n^{-H} , 由于 $A_n^H A_n^{-H} = A_{n-1}^H A_{n-1}^{-H} = I_M$, 而 $A_n^H A_n^{-H}$ 、 $A_{n-1}^H A_{n-1}^{-H}$ 又可以分别表示为

$$\begin{cases} A_n^H A_n^{-H} = \begin{bmatrix} A_n^H & \mathbf{0} \end{bmatrix} \begin{bmatrix} A_n^{-H} \\ \# \end{bmatrix} \\ A_{n-1}^H A_{n-1}^{-H} = \begin{bmatrix} A_{n-1}^H & \mathbf{x}(n) & \mathbf{x}(n-K) \end{bmatrix} \Phi \begin{bmatrix} A_{n-1}^{-H} \\ \mathbf{0} \end{bmatrix} \\ = \begin{bmatrix} A_n^H & \mathbf{0} \end{bmatrix} \Phi \tilde{Q}_n \begin{bmatrix} A_{n-1}^{-H} \\ \mathbf{0} \end{bmatrix} \end{cases} \quad (23)$$

因此可得:

$$\Phi \tilde{Q}_n \begin{bmatrix} A_{n-1}^{-H} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} A_n^{-H} \\ \# \end{bmatrix} \quad (24)$$

对式 (24) 左右两边同乘以 Φ^{-1} 有:

$$\tilde{Q}_n \begin{bmatrix} A_{n-1}^{-H} \\ \mathbf{0} \end{bmatrix} = \Phi \begin{bmatrix} A_n^{-H} \\ \# \end{bmatrix} = \begin{bmatrix} A_n^{-H} \\ \# \end{bmatrix} \quad (25)$$

综合式 (12)、式 (22) 和式 (25) 可以看出, 当用 \tilde{Q}_n 把 A_{n-1} 更新为 A_n 的同时, 相同的变换矩阵可以同时把 \mathbf{v}_{n-1} 和 A_{n-1}^{-H} 更新为 \mathbf{v}_n 和 A_n^{-H} 。

总结 MIQRD-SMI 算法的递推实现步骤如下。

步骤 1 初始化: 初始化状态在积累 K 个样本数据后, 对应中间向量 \mathbf{v}_K 和下三角矩阵 A_K^{-H} 可以利用 IQRD-SMI 算法递推得到。

步骤 2 $n=K+1, K+2, \dots$

1) 利用式 (18)、式 (19) 和式 (16) 计算变换矩阵 H_1, \dots, H_M ;

2) 利用 1) 确定的 H_1, \dots, H_M 更新得到中间向量 \mathbf{v}_n 和下三角矩阵 A_n^{-H} , 即

$$H_M \cdots H_2 H_1 \begin{bmatrix} A_{n-1}^{-H} & \mathbf{v}_{n-1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} A_n^{-H} & \mathbf{v}_n \\ \# & \# \end{bmatrix}$$

3) 更新自适应权: $\mathbf{w}_n = (A_n^{-H} \mathbf{v}_n^H)^H = (A_n^{-H})^H \mathbf{v}_n$ 。

步骤 2 的 1) 中求解 H_1, \dots, H_M 的过程也就是对 A_n 进行更新的过程, 与 MQRD-SMI 算法相比, 增加了对 A_n^{-H} 和 \mathbf{v}_n 的更新过程, 可见 IMQRD-SMI 算法相对于 MQRD-SMI 算法在运算量上没有优势。但是, 由于对 A_n^{-H} 、 \mathbf{v}_n 和更新和对 A_n 的更新可以同时进行, 与只更新 A_n 具有相当的运算速度, 而步骤 2 的 3) 中的矩阵乘法不但运算量小于前后向回代, 而且可以采用并行运算, 速度比前后向回代快很多, 因此 MIQRD-SMI 算法具有更好的并行性, 更快的运算速度, 更有利于系统的实时实现。

4.2 MIQRD-LSMI 算法

为了提高算法的稳健性, 对 MIQRD-SMI 算法进行对角加载处理, 记为 MIQRD-LSMI 算法。加载后的协方差矩阵可以表示为

$$\begin{aligned} \mathbf{R}_L &= \mathbf{R}_x + \sigma^2 \mathbf{I}_M = \frac{1}{K} \mathbf{X}_n \mathbf{X}_n^H + \sigma^2 \mathbf{I}_M \\ &= \frac{1}{K} \begin{bmatrix} \sqrt{K} \sigma \mathbf{I}_M & \mathbf{X}_n \\ \mathbf{X}_n^H & \end{bmatrix} \end{aligned} \quad (26)$$

令 $\mathbf{Y}_n = \begin{bmatrix} \sqrt{K} \sigma \mathbf{I}_M & \mathbf{X}_n \end{bmatrix}$, 则有:

$$\begin{cases} \mathbf{R}_L = \mathbf{Y}_n \mathbf{Y}_n^H / K \\ \mathbf{Y}_n \mathbf{Y}_n^H = \mathbf{Y}_{n-1} \mathbf{Y}_{n-1}^H + \mathbf{x}(n) \mathbf{x}^H(n) - \mathbf{x}(n-K) \mathbf{x}^H(n-K) \end{cases} \quad (27)$$

对 \mathbf{Y}_n^H 进行 QR 分解, 得:

$$\mathbf{Q}_n \mathbf{Y}_n^H = \begin{bmatrix} \mathbf{B}_n \\ \mathbf{0} \end{bmatrix} \quad (28)$$

这样, 对角加载后的自适应权的求解公式 $\mathbf{w} = \mathbf{R}_L^{-1} \mathbf{a}_s$ 可以转换为

$$\begin{cases} \mathbf{v}_n = \mathbf{B}_n^{-H} \mathbf{a}_s \\ \mathbf{w}_n^H = \mathbf{v}_n^H \mathbf{B}_n^{-H} \end{cases} \quad (29)$$

设在 $n-1$ 时刻已经实现了对 \mathbf{Y}_{n-1}^H 的 QR 分解, 即 $\mathbf{Y}_{n-1} \mathbf{Y}_{n-1}^H = \mathbf{B}_{n-1}^H \mathbf{B}_{n-1}$, 则有:

$$\mathbf{Y}_n \mathbf{Y}_n^H = \mathbf{B}_{n-1}^H \mathbf{B}_{n-1} + \mathbf{x}(n) \mathbf{x}^H(n) - \mathbf{x}(n-K) \mathbf{x}^H(n-K) \quad (30)$$

式(30)与式(10)具有完全相同的形式, 可以利用

更新 A_n 的方法更新 B_n ，并且不进行对角加载时的递推公式只适用 $K \geq M$ 时，而对角加载后的递推公式也适用于 $K < M$ 的情况。

5 计算机仿真

仿真针对有限样本数目的实际应用环境，可以利用的有效样本数为 K 。由于 MQRD-SMI 算法与 MIQRD-SMI 算法本质上相同，只是具体的实现方式不同，因此这里只对文中提出的 MIQRD-SMI 算法和 MIQRD-LSMI 算法进行仿真，并将其与样本积累方式的 IQRD-SMI 算法进行比较。

考虑阵元间距为半波长的 10 元等距线阵，期望信号方向为 0° ，有 2 个干扰信号，入射方向分别为 40° 、 -30° ，干扰噪声比分别为 30dB、35dB；设样本数据中不含期望信号。

仿真 1 观察几种算法的性能随迭代次数 n 变化的曲线如图 1 所示，有效样本数分别取 $K=20$ 、 $K=100$ 。可以看出，IQRD-SMI 算法在每次积累够 K 个样本后重新开始积累，其阵增益随迭代次数呈周期性变化，最优性能在每次正好积累到 K 个样本时

获得，而 MIQRD-SMI 算法和 MIQRD-LSMI 算法在 $n \geq K$ 时阵增益基本保持不变，其中前者的阵增益与 IQRD-SMI 的最优情况相等，而后的阵增益比 IQRD-SMI 的最优情况还要大。可见 2 种固定样本数目的算法的阵增益损失远小于常规算法，其中 MIQRD-LSMI 算法最小。随着有效样本数 K 的增多，几种算法的阵增益损失逐渐减小，其中 MIQRD-SMI 算法最为明显，这是由于小特征值分散程度变小，加载量的作用相对变弱，因此 MIQRD-SMI 算法在 $n \geq K$ 时的性能越来越接近于 MIQRD-SMI 算法。

仿真 2 设 $K=100$ ，观察不同迭代次数 n 时几种算法的波束图如图 2 所示。可以看出，在 $n=100$ 时，由于所用的样本数都为 100，IQRD-SMI 算法和 MIQRD-LSMI 算法的波束形状一致，而 MIQRD-LSMI 算法的旁瓣水平较前 2 种方法有所改善；在 $n=120$ 时，IQRD-SMI 利用的样本数仅为 20，而其他 2 种方法所用的样本数仍为 100，因此 IQRD-SMI 算法的波束性能最差，MIQRD-SMI 算法、

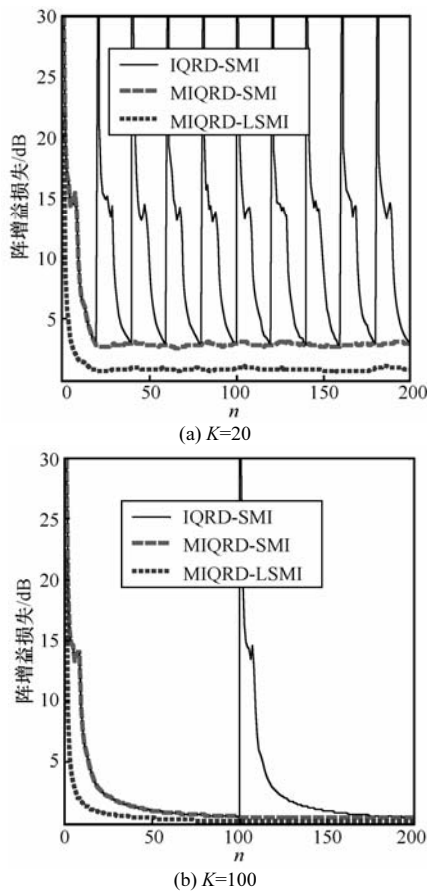


图 1 几种算法的性能随迭代次数变化曲线

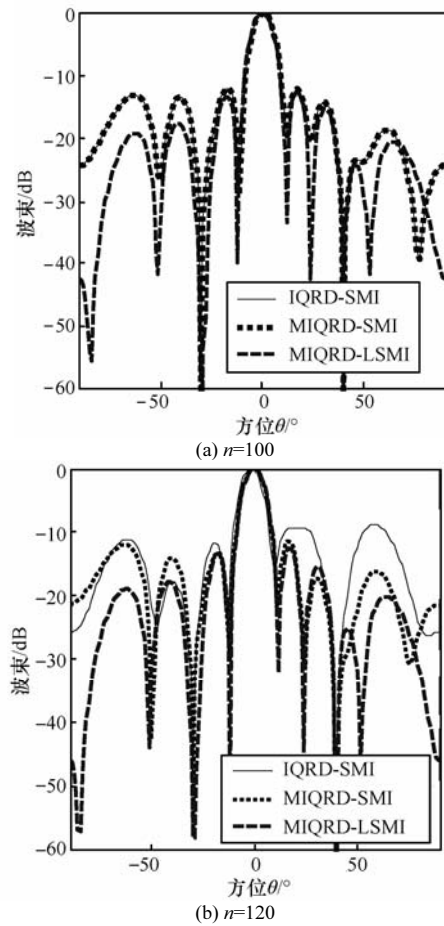


图 2 不同迭代次数时几种算法的波束图

MIQRD-LSMI 算法对其改善非常明显,其中又以后者更优。对于 2 种固定样本数目的方法,在不同迭代次数 $n(n \geq K)$ 时由于采用的样本数据不同,波束形状不完全相同,但由于样本数目相同,其旁瓣水平相当,波束形成性能没有明显差别。

6 结束语

本文提出了固定样本数目下的 QR 分解递推算法,并对其运算速度和稳健性进行了一系列的改进,最后得出的 MIQRD-LSMI 递推算法不仅具有良好的波束形成性能,运算速度也大大提高,能够实时地更新自适应权,数值特性也优于常规算法。与常规的 QR 分解的递推算法相比,在有限样本数目的实际应用环境下,该方法具有更高的阵增益和更好的波束性能。

参考文献:

- [1] 冯地耕. 基于 QR 分解的 ADBF 算法及其 DSP 实现研究[D]. 成都: 电子科技大学, 2005.
FENG D G. ADBF Algorithm Based on QR Decomposition and Its Implementation[D]. Chengdu: University of Electronic Science and Technology of China, 2005.
- [2] 李荣锋, 王永良, 万山虎. 基于快速逆 QR 分解的自适应波束形成方法[J]. 系统工程与电子技术, 2002, 24(9): 27-29.
LI R F, WANG Y L, WAN S H. A square-root-free approach for IQRD-based adaptive beamforming[J]. Systems Engineering and Electronics, 2002, 24(9): 27-29.
- [3] CAO J S, WANG X G. LSMI algorithm based on inverse QR decomposition[J]. IEEE Proc on Communications, Circuits and Systems, 2006, (6): 262-265.
- [4] CHEN J W, NI J L, WANG Y L. Inverse QR iterative algorithm for space-time adaptive processing[J]. IEEE Trans on Signal Processing, 2002, 2(8): 1429-1432.
- [5] 吴建新, 王彤, 索志勇等. 空时自适应处理的滑窗递推 QR 算法[J]. 电子与信息学报, 2008, 30(10): 2338-2342.
- [6] WU J X, WANG T, SUO Z Y, *et al.* Sliding window recursive QR factorization of space-time adaptive processing algorithm[J]. Journal of Electronic & Information Technology, 2008, 30(10): 2338-2342.
- [7] BAYKAL B, CONSTANTINIDES A G. Sliding window adaptive fast QR and QR-lattice algorithms[J]. IEEE Trans on Signal Processing, 1998, 46 (11): 2877-2887.
- [8] 谢磊, 梁武星, 张泉灵等. 基于快速滑窗 QR 分解的自适应子空间辨识[J]. 化工学报, 2008, 59(6): 1448-1453.
XIE L, LIANG W X, ZHANG Q L, *et al.* Adaptive subspace identification based on fast moving window QR decomposition[J]. Journal of Chemical Industry and Engineering, 2008, 59(6): 1448-1453.
- [9] BOJANCZYK A D, STEINHARDT A O. Stabilized hyperbolic householder transformations[J]. IEEE Trans on Acoustics, Speech and Signal Processing, 1989, 37(8): 1286-1288.

作者简介:



倪淑燕 (1981-), 女, 河北清河人, 装备指挥技术学院博士生, 主要研究方向为阵列信号处理。



程乃平 (1962-), 男, 陕西扶风人, 装备指挥技术学院教授、博士生导师, 主要研究方向为空间信息传输技术。



倪正中 (1985-), 男, 河北邢台人, 遥感信息研究所助理研究员, 主要研究方向为航天摄影技术。