

一种基于子阵列合成的 DOA 估计算法

许文龙, 蒋 伟, 尚 勇, 项海格
(北京大学电子学系卫星通信研究室, 北京 100871)

摘 要: 在卫星通信中, 由于星上资源有限, 导致星载信号处理接收机的数目一般少于阵列天线的阵元数. 在某些应用中需要在这样的条件下实现 DOA 估计: 信号达波数大于星载信号处理接收机数目但是可能小于阵列天线的阵元数. 实质上是要求能够用较少数量的接收机进行接收、处理含有较多阵元的阵列天线上的信号. 针对这类应用要求, 本文提出一种基于子阵列合成的 DOA 估计算法. 通过子阵列输出协方差矩阵合成等效的大阵列输出协方差矩阵, 基于等效大阵列输出协方差矩阵实现达波 DOA 超分辨率估计; 讨论了子阵列天线选择优化问题. 对算法性能进行了仿真验证, 结果表明算法是有效的.

关键词: 阵列信号处理; DOA 估计; 子阵列合成; 时分复用 TDM; 特征空间分解

中图分类号: TN927 **文献标识码:** A **文章编号:** 0372-2112 (2006) 09-1571-07

A Subarray-Synthesis Based DOA Estimation Algorithm

XU Wen-long, JIANG Wei, SHANG Yong, XIANG Haige

(Satellite Communications Lab, Department of Electronics, Peking University, Beijing 100871, China)

Abstract In some satellite communications applications, we need to perform Direction of Arrival (DOA) angle estimation under the restriction that shorting of receivers when receiving and processing the signals impinged on an array antenna. In essence, this demands that using a limited number of receivers to process a large number of DOA signal. This paper proposes a method named Subarray synthesis based Direction of Arrival angle estimation. An equivalent output covariance matrix is firstly synthesized based on a given set of subarray antennas' output covariance matrices, impinged signal's DOA estimation is then performed based on the equivalent covariance matrix. Subarray optimization criterion and method are also discussed. Monte Carlo simulations verified the effectiveness of the estimation method.

Key words array signal processing; direction of arrival angle (DOA) estimation; subarray synthesis; time division multiplexing (TDM); eigenspace decomposition

1 引言

卫星通信中的星上资源有限, 星载设备的数量、体积、重量、总功耗等受到严格限制. 相应地, 星载信号处理接收机的数量受到限制, 其数量通常比星载阵列天线阵元数少. 另外, 在某些应用中, 信源波达数大于接收机数目, 但是可能少于星载阵列天线阵元数. 在这种条件下采用通常的 DOA 算法, 或者由于接收机数量少于信号波达数, 无法实现 DOA 估计; 或者由于信号处理接收机数目少于阵列天线阵元数, 只能处理少量阵元的输出快拍数据, 而使 DOA 估计的分辨率降低. 这是因为接收机不能接收到其余阵元上的输出快拍, 使阵列天线实际被使用的孔径变小了, 天线资源没有得到充分利用^[1~5]. 这类 DOA 估计算法的优点

是易于实现, 天线阵元与接收机直接相连. 缺点是在接收机数量少于阵元数的条件下, 阵列天线的资源不能得到充分利用. 由于星上资源异常宝贵, 不允许浪费, 这种 DOA 估计方案不适合在卫星通信中应用. 针对这种情况, 本文提出一种 DOA 超分辨率估计算法, 算法在满足卫星通信星载设备要求的条件下, 使用少量信号处理接收机处理阵列天线全部阵元的输出快拍, 充分利用了阵列天线的孔径资源.

本算法基于子阵列天线合成原理. 首先, 在每个快拍取样时刻对各子阵列天线阵元输出快拍进行取样; 顺序计算各子阵列天线全部阵元的输出协方差矩阵. 经过确定的取样次数后, 可以计算出全部子阵列天线输出的协方差矩阵. 原理上, 通过有限数量的接收机可以处理包含任意阵

元数的阵列天线输出信号,因而可以实现对大阵列天线信号的接收和处理.第二,基于所求出的各子阵列天线输出协方差,基于一定准则,合成等效大阵列输出协方差矩阵,最后实现信号波达方向(DOA)的超分辨估计.

与其他算法相比,本文所提出的算法能够在满足星载通信条件的要求下实现达波的DOA估计.

所提出的算法虽然是基于星载系统的特点提出的,但是,在一定条件下也可以应用在其他诸如机载雷达,移动通信等系统中,实现达波DOA估计或数字波束形成等.

2 阵列天线全输入、接收方程^[6]

全输入信号定义为全部波达信号混合后在阵列天线输入端形成的入射信号与天线加性噪声信号之和,全接收信号指阵列天线的输出信号、接收机的输入信号 $Y(n)$.

设:均匀线阵

(ULA)如图1所示.

阵列天线阵元数为 M ,阵元间距为 d 以阵列左边第一个阵元作为参考,编号为 $0 \sim P-1$ 个窄带远场波达信号.

波达信号中心频率为 f_0 ,波达方向为 $\{\theta_p, p=1, \dots, P\}$,各信源在基准点的复包络分别为 $\{s_p(n)\}$,括号中的 n 为快拍取样时刻, $s_p(n)$ 表示在取样时刻 n 来自波达方向 θ_p 的入射信号复包络, $p=1, 2, \dots, P$.阵列天线加性噪声 $N(n) = \{n_m(n)\}$,为时域和空域上的复高斯白噪声, $E\{n_m(n)\} = 0$, $E\{n_m(n)n_m^H(n)\} = \sigma^2 \delta_{m,n}$.噪声之间、噪声与波达信号之间、波达信号之间均互不相关.下标 m, n 表示阵元编号,

上标 H 表示共轭转置, $\delta_{m,n} = \begin{cases} 1 & m=n \\ 0 & m \neq n \end{cases}$.第 m 个阵元上第 n 次快拍取样值可表示为:

$$y_m(n) = \sum_{p=1}^P s_p(n) \exp\left[-j \frac{2\pi}{\lambda} (m-1)d \sin\theta_p\right] + n_m(n) \quad (1)$$

式中 $n_m(n)$ 表示第 m 个阵元上噪声的第 n 个快拍.式(1)可简记为

$$y_m = \sum_{p=1}^P s_p \exp\left[-j \frac{2\pi}{\lambda} (m-1)d \sin\theta_p\right] + n_m \quad (2)$$

阵列天线第 n 个输出快拍可写成向量形式:

$$Y(n) = AS(n) + N(n) \quad (3)$$

式中, A 是阵列天线方向矩阵, $S(n)$ 是波达信号向量, $N(n)$ 是阵列天线加性噪声.

$$Y(n)^T = [y_0(n), \dots, y_m(n), \dots, y_{M-1}(n)]$$

$$a(\theta_p)^T = \left[1, \dots, \exp\left[-j \frac{2\pi d \sin\theta_p}{\lambda}\right], \dots, \exp\left[-j \frac{2\pi d (M-1) \sin\theta_p}{\lambda}\right] \right]$$

$$A = [a(\theta_1), \dots, a(\theta_p), \dots, a(\theta_P)]$$

$$S(n)^T = [s_1(n), \dots, s_p(n), \dots, s_P(n)]$$

$$N(n)^T = [n_0(n), \dots, n_m(n), \dots, n_{M-1}(n)]$$

3 基于子阵列合成大协方差矩阵

在通常的DOA估计算法例如基于协方差矩阵特征值分解的MUSIC算法中,DOA估计的分辨率在很大程度上取决于阵列天线系统输出协方差矩阵的秩.在一定条件下,阵列天线有效孔径越大,输出协方差矩阵的秩越大,则DOA估计的分辨率越高.在波达信号之间以及阵元之间均互不相关时,协方差矩阵的维数取决于阵列天线的阵元数.因此,只有使阵列天线的孔径达到最大才能使输出协方差矩阵的维数达到最大值.在这种DOA估计算法中,隐含条件是信号处理接收机的数量等于阵列天线的阵元数,信号处理接收机的输出协方差矩阵等效于阵列天线阵元输出协方差矩阵.

如引言所述,在卫星通信中,星载信号处理接收机的数量通常比阵列天线的阵元数少,信号处理接收机输出协方差矩阵的维数小于阵列天线输出协方差矩阵的维数.这时,阵列天线输出协方差矩阵与信号处理接收机输出协方差矩阵不再等效.因为实际能够得到的快拍样本是星载信号处理接收机输出的信号,所以,星载接收机输出协方差矩阵是进行DOA估计的唯一依据,因而估计的分辨率取决于星载接收机输出协方差矩阵,不取决于阵列天线输出的协方差矩阵,这样就没法充分利用星载阵列天线的阵元(孔径)资源.导致这一损失的根源是信号处理接收机只采集了阵列天线中部分阵元的输出信号样本,而丢失了其余阵元的输出信号信息,使得信号处理接收机的输出协方差矩阵的维数小于阵列天线的维数.

从物理意义上看,不论是信号处理接收机输出的协方差矩阵,还是阵列天线输出的协方差矩阵,描述的都是阵列天线阵元的输出信号间的相关程度,差别在于:信号处理接收机输出协方差矩阵只包含那些将信号送到接收机的阵元之间的协方差,而阵列天线输出协方差包含全部阵元之间的协方差.协方差矩阵的元素是两个阵元输出信号之间的相关值.在天线阵元数大于信号处理接收机数量的条件下,不论采用何种方式,只要能够在信号处理接收机输出端获得阵列天线全部阵元的输出快拍样本,就可以计算出协方差矩阵进而实现DOA估计.

为此,我们提出一种基于子阵列合成的DOA估计算法.其思想如下:在一定条件下,通过时分复用接收机,基于子阵列天线输出合成等效的大阵列天线输出协方差矩阵.实现方法如下所示.

波达信号在时域、空域满足分组(Block by block)广义平稳、遍历的条件时,我们基于接收机数目,依据规定的划分准则,对阵列天线(本文称之为大阵列天线)进行拆分,组成一系列包含较少数量阵元的小阵列天线,本文称之为

子阵列天线. 每个子阵列天线包含的阵元数等于信号处理接收机的数量. 这样, 在每个取样时刻, 接收机处理某一个子阵列天线的输出快拍; 通过时分复用, 经过一定数量的取样周期后, 可以在接收机输出端获得大阵列天线全部阵元的输出快拍. 子阵列天线的样本数由要求的估计误差精度决定.

基于子阵列天线输出快拍样本可以计算子阵列输出协方差矩阵.

根据子阵列天线阵元与大阵列天线阵元的对应关系, 可以基于子阵列天线的输出协方差矩阵合成大阵列天线等效输出协方差矩阵, 从而基于等效输出协方差矩阵的特征分解实现 DOA 估计.

我们结合实例证明上述算法的有效性, 实质上是算法的一个具体实现. 首先直接用大阵列天线输出快拍计算协方差矩阵, 然后用基于子阵列合成的方法构造等效输出协方差矩阵, 进行 DOA 估计.

假设: 等距线阵 (ULA), 大阵列天线阵元数 $M = 4$ 相邻阵元间距 $d = \lambda/2$ 表示波达信号载波中心波长; 每个子阵列天线阵元数 $M_s = 2$ 波达信号数 $P = 3$ 波达信号复振幅为向量 $S(n)$, 波达角为 $\{\theta_p, p = 1, 2, 3\} \in \{\theta_1, \theta_2, \theta_3\}$; 波达信号之间互不相关, 在时域、空域服从分组广义平稳、遍历, 分组长度等于子阵列数与子阵列取样快拍数之积, $L \times N$ 其中 L 为子阵列数, N 为子阵列取样快拍数; 加性噪声 $N(n)$ 服从复高斯分布, 均值为零, 方差为 σ^2 , 噪声之间、噪声与波达信号之间互不相关.

由于 $Y(n) = A(\theta)S(n) + N(n)$

大阵列天线输出的协方差矩阵 R 为:

$$R = E\{Y(n)Y^H(n)\} \quad (4)$$

其中 $E\{\cdot\}$ 表示统计均值. 展开式 (4) 可得:

$$\begin{aligned} R &= E\{[A(\theta)S(n) + N(n)][A^H(\theta)S^H(n) + N^H(n)]\} \\ &= E\{A(\theta)S(n)S^H(n)A^H(\theta) + N(n)S^H(n)A^H(\theta) \\ &\quad + A(\theta)S(n)N^H(n) + N(n)N^H(n)\} \end{aligned} \quad (5)$$

由于信源信号与噪声相互独立, 式 (5) 中的交叉乘积项等于零, 所以

$$\begin{aligned} R &= E\{A(\theta)S(n)S^H(n)A^H(\theta)\} + E\{N(n)N^H(n)\} \\ &= A(\theta)R_s(n)A^H(\theta) + \sigma^2 I \end{aligned} \quad (6)$$

其中, I 是 4×4 阶单位阵.

由式 (6) 可知, 求出波达信号协方差矩阵 R_s , 可相应地求出阵列天线输出信号的协方差矩阵 R . R 和 R_s 还可以分别写成如下形式:

$$R_s = \begin{bmatrix} r_{s11} & 0 & 0 \\ 0 & r_{s22} & 0 \\ 0 & 0 & r_{s33} \end{bmatrix} \quad (7)$$

$$R = \begin{bmatrix} r_{00} & r_{01} & r_{02} & r_{03} \\ r_{10} & r_{11} & r_{12} & r_{13} \\ r_{20} & r_{21} & r_{22} & r_{23} \\ r_{30} & r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (8)$$

R_s 中的 $r_{pp}, p = 1, 2, 3$ 表示来自波达方向 θ_p 的入射信号协方差, 由于假设信号之间不相关, 所以是对角阵. R 中的 $r_{mn}, m, n = 0, 1, 2, 3$ 表示阵元 m, n 输出的协方差. 由式 (6) 可知, 在大阵列天线阵元数等于信号处理接收机数目时, 接收机在每个取样时刻都能获得全部阵元的输出快拍样本, 因而可以根据式 (6) 直接计算出阵列输出协方差矩阵. 在大阵列天线阵元数大于信号处理接收机数目时则做不到这一点.

基于子阵列合成等效大阵列协方差矩阵时, 首先基于子阵列天线输出信号计算子阵列的输出协方差矩阵 R_{mn} , 然后根据子阵列天线与大阵列天线阵元的对应关系, 基于子阵列协方差矩阵元素合成等效大阵列天线协方差矩阵. 为便于区别, 大阵列天线的等效输出协方差矩阵用 R_{eq} 表示, 简称为等效输出协方差矩阵.

如图 1 所示, 阵元编号依次为 0, 1, 2, 3 遍历选取阵元, 则子阵列天线数 $L = C_4^2 = 6$ 个子阵列天线分别由阵元 (0, 1), (1, 2), (2, 3), (3, 0), (0, 2), (1, 3) 组成. 在这种情况下, 除去排列顺序外, 子阵列选择方案是唯一的, 子阵列阵元的选择也是最优的.

设 $v_1 = \exp\left[-j\frac{2\pi d}{\lambda}\sin(\theta_1)\right]$, $v_2 = \exp\left[-j\frac{2\pi d}{\lambda}\sin(\theta_2)\right]$, $v_3 = \exp\left[-j\frac{2\pi d}{\lambda}\sin(\theta_3)\right]$, 则子阵列天线方向函数分别为:

$$A_1(\theta) = [\alpha_1(\theta_1), \alpha_1(\theta_2), \alpha_1(\theta_3)] = \begin{bmatrix} 1 & 1 & 1 \\ v_1 & v_2 & v_3 \end{bmatrix} \quad (9)$$

$$A_2(\theta) = [\alpha_2(\theta_1), \alpha_2(\theta_2), \alpha_2(\theta_3)] = \begin{bmatrix} v_1 & v_2 & v_3 \\ v_1^2 & v_2^2 & v_3^2 \end{bmatrix} \quad (10)$$

$$A_3(\theta) = [\alpha_3(\theta_1), \alpha_3(\theta_2), \alpha_3(\theta_3)] = \begin{bmatrix} v_1^2 & v_2^2 & v_3^2 \\ v_1^3 & v_2^3 & v_3^3 \end{bmatrix} \quad (11)$$

$$A_4(\theta) = [\alpha_4(\theta_1), \alpha_4(\theta_2), \alpha_4(\theta_3)] = \begin{bmatrix} v_1^3 & v_2^3 & v_3^3 \\ 1 & 1 & 1 \end{bmatrix} \quad (12)$$

$$A_5(\theta) = [\alpha_5(\theta_1), \alpha_5(\theta_2), \alpha_5(\theta_3)] = \begin{bmatrix} 1 & 1 & 1 \\ v_1^2 & v_2^2 & v_3^2 \end{bmatrix} \quad (13)$$

$$A_6(\theta) = [\alpha_6(\theta_1), \alpha_6(\theta_2), \alpha_6(\theta_3)] = \begin{bmatrix} v_1 & v_2 & v_3 \\ v_1^3 & v_2^3 & v_3^3 \end{bmatrix} \quad (14)$$

设子阵列天线输出的协方差矩阵为 $R_i, i \in (1, 2, \dots, 6)$, 则由式 (6) 可得

$$R_i = A_i(\theta)R_{si}A_i^H(\theta) + \sigma^2 I \quad (15)$$

其中, R_{si} 为波达信号协方差矩阵, I 为 2×2 阶单位阵.

在形式上 R_{si} 与 R_s 是相同的, 都是 $P \times P$ 维矩阵, 此处为 3×3 维, 都表示信源波达信号之间的协方差矩阵; 当波达信号之间互不相关时, R_{si}, R_s 均为对角阵. 实质上, 它们是完全相同的. 这是因为, 虽然不同快拍时刻接收的波达信号矢量 $S_p(n)$ 以及加性噪声矢量 $N(n)$ 不完全相同, 但是, 由于是广义平稳、遍历的随机过程, 在处理周期内它们的统计特性是相同的. 所以, 为了保证在处理周期内统计

特性相同, 要求波达信号在每一个处理周期 $T_{\text{proc}} = T \times L$ 内是广义平稳、在空域和时域遍历的随机过程, 即是分组广义平稳、遍历的, 其中 T 为一个快拍取样周期时间, L 为子阵列天线数. 在这种条件下, 则可以认为 $R_{si} = R_s$, $\sigma_i^2 = \sigma^2$, 因而 $R_{eq} = R$ 其中 σ_i^2 是子阵列天线 i 的加性噪声方差, σ^2 是大阵列天线的噪声方差.

一种建议的快拍取样方案如下: 在第一个快拍时刻对第一个子阵列全部阵元输出信号取样, 第二个快拍时刻对第二个子阵列全部阵元输出信号取样, ..., 第 L 个快拍时刻对第 L 个子阵列全部阵元输出信号取样, 这样依次对各子阵列输出数据进行采样; 完成第 L 个子阵列天线输出信号的采样之后, 开始新的循环取样操作, 重新对第一个子阵列天线输出信号采样, 如此循环进行 N 次. 取样次数 N 可根据估计误差精度要求确定. 我们定义: 对全部 L 个子阵列天线分别进行一次采样所需的快拍取样次数称为子阵列天线的一个采样处理周期. 因此, 基于子阵列合成的采样处理周期等于基于大阵列直接处理的快拍采样周期 T 的 L 倍, 这里是子阵列天线数.

将式 (15) 展开更便于观察等效大阵列天线输出协方差矩阵的合成过程.

$$\begin{aligned} R_1 &= \begin{bmatrix} 1 & 1 & 1 \\ v_1 & v_2 & v_3 \end{bmatrix} R_s \begin{bmatrix} 1 & 1 & 1 \\ v_1 & v_2 & v_3 \end{bmatrix}^H + \sigma_1^2 I \\ &= \begin{bmatrix} 1 & 1 & 1 \\ v_1 & v_2 & v_3 \end{bmatrix} R_s \begin{bmatrix} 1 & 1 & 1 \\ v_1 & v_2 & v_3 \end{bmatrix}^H + \sigma^2 E \begin{bmatrix} r_{00} & r_{01} \\ r_{10} & r_{11} \end{bmatrix} \end{aligned} \quad (16)$$

式 (16) 中, r_{mn} 表示子阵列天线阵元 m 、 n 之间的协方差, 从子阵列天线与大阵列天线阵元的对应关系上看, 它正好也是大阵列天线阵元 m 、 n 之间的协方差, 其中, $m, n = 0 \sim 1$

同理, 可写出其余子阵列天线的输出协方差矩阵, 分别为 R_2, R_3, R_4, R_5, R_6 , 为便于观察, 连同 R_1 一并列出:

$$R_1 = \begin{bmatrix} r_{00} & r_{01} \\ r_{10} & r_{11} \end{bmatrix} = \begin{bmatrix} r_{01}(1) & r_{01}(1) \\ r_{10}(1) & r_{11}(1) \end{bmatrix} \quad (17)$$

$$R_2 = \begin{bmatrix} r_{00} & r_{01} \\ r_{10} & r_{11} \end{bmatrix} = \begin{bmatrix} r_{11}(2) & r_{12}(1) \\ r_{21}(1) & r_{22}(1) \end{bmatrix} \quad (18)$$

$$R_3 = \begin{bmatrix} r_{00} & r_{01} \\ r_{10} & r_{11} \end{bmatrix} = \begin{bmatrix} r_{22}(2) & r_{23}(1) \\ r_{32}(1) & r_{33}(1) \end{bmatrix} \quad (19)$$

$$R_4 = \begin{bmatrix} r_{00} & r_{01} \\ r_{10} & r_{11} \end{bmatrix} = \begin{bmatrix} r_{33}(2) & r_{30}(1) \\ r_{03}(1) & r_{00}(2) \end{bmatrix} \quad (20)$$

$$R_5 = \begin{bmatrix} r_{00} & r_{01} \\ r_{10} & r_{11} \end{bmatrix} = \begin{bmatrix} r_{00}(3) & r_{02}(1) \\ r_{20}(1) & r_{22}(3) \end{bmatrix} \quad (21)$$

$$R_6 = \begin{bmatrix} r_{00} & r_{01} \\ r_{10} & r_{11} \end{bmatrix} = \begin{bmatrix} r_{11}(2) & r_{13}(1) \\ r_{31}(1) & r_{33}(3) \end{bmatrix} \quad (22)$$

在式 (17) ~ 式 (22) 中, 第一个等号后面矩阵中的元

素 r_{mn} 表示子阵列天线中的阵元 m 和 n 的输出信号之间的协方差, 由于每个子阵列天线只包含两个阵元, 按照约定的阵元编号方法, 在每个子阵列天线中, 阵元编号只有 $Q = 1$ 二值. 因此, 第一个等号后面的协方差矩阵形式完全相同. 在第二个等号后面矩阵中的元素 $r_{mn}(l)$ 表示这一项对应于大阵列天线中的阵元 m 和 n 的输出信号之间的协方差, 由于大阵列天线由四个阵元组成, 所以阵元编号 m 和 n 的取值为 $Q = 1, 2, 3$ 括号中的值 l 表示协方差项 $r_{mn}(l)$ 在基于子阵列计算协方差矩阵时出现的次数. 例如, 式 (18) 中的 $r_{11}(2)$ 表示协方差项 r_{11} 第二次出现, 也就是第一次重复出现. 由式 (17)、(18) 可见, r_{11} 第一次出现在协方差矩阵 R_1 中, 第二次出现在协方差矩阵 R_2 中.

实际上, 式 (17) ~ 式 (22) 给出了等效大阵列协方差矩阵 R_{eq} 的全部协方差元素:

$$R_{eq} = \begin{bmatrix} R_1(0,0) & R_1(0,1) & R_5(0,1) & R_4(1,0) \\ R_1(1,0) & R_1(1,1) & R_2(0,1) & R_6(0,1) \\ R_5(1,0) & R_2(1,0) & R_2(1,1) & R_3(0,1) \\ R_4(0,1) & R_6(1,0) & R_3(1,0) & R_3(1,1) \end{bmatrix} \quad (23)$$

式 (23) 中, 协方差项 $R_i(m, n)$ 表示取矩阵 R_i 中第 m 行、第 n 列的元素. 在 R_i 中, m, n 只取 0 或 1 从式 (23) 可知, 此例中的等效大阵列输出协方差矩阵 R_{eq} 是 4×4 维的方阵.

同理, 当大阵列天线包含八个阵元、子阵列天线包含四个阵元 (即系统有四个信号处理接收机) 时, 可基于下列六个子阵列天线, 合成等效大阵列天线输出协方差矩阵 R_{eq} : 1 2 3 4 5 6 7 8 1 2 5 6 1 2 7 8 3 4 5 6 3 4 7 8 对应的协方差矩阵如下所示, 每个协方差矩阵分别对应一个子阵列天线.

$$R_1 = \begin{bmatrix} \textcircled{R} & \textcircled{R} & \textcircled{R} & \textcircled{R} & \times & \times & \times & \times \\ \textcircled{R} & \textcircled{R} & \textcircled{R} & \textcircled{R} & \times & \times & \times & \times \\ \textcircled{R} & \textcircled{R} & \textcircled{R} & \textcircled{R} & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \end{bmatrix} \quad (24)$$

$$R_2 = \begin{bmatrix} \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \textcircled{R} & \textcircled{R} & \textcircled{R} & \textcircled{R} \\ \times & \times & \times & \times & \textcircled{R} & \textcircled{R} & \textcircled{R} & \textcircled{R} \\ \times & \times & \times & \times & \textcircled{R} & \textcircled{R} & \textcircled{R} & \textcircled{R} \\ \times & \times & \times & \times & \textcircled{R} & \textcircled{R} & \textcircled{R} & \textcircled{R} \end{bmatrix} \quad (25)$$

$$R_3 = \begin{bmatrix} \textcircled{R} & \textcircled{R} & \times & \times & \textcircled{R} & \textcircled{R} & \times & \times \\ \textcircled{R} & \textcircled{R} & \times & \times & \textcircled{R} & \textcircled{R} & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \textcircled{R} & \textcircled{R} & \times & \times & \textcircled{R} & \textcircled{R} & \times & \times \\ \textcircled{R} & \textcircled{R} & \times & \times & \textcircled{R} & \textcircled{R} & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \end{bmatrix} \quad (26)$$

$$R_4 = \begin{bmatrix} \textcircled{R} & \textcircled{R} & \times & \times & \times & \times & \textcircled{R} & \textcircled{R} \\ \textcircled{R} & \textcircled{R} & \times & \times & \times & \times & \textcircled{R} & \textcircled{R} \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \textcircled{R} & \textcircled{R} & \times & \times & \times & \times & \textcircled{R} & \textcircled{R} \\ \textcircled{R} & \textcircled{R} & \times & \times & \times & \times & \textcircled{R} & \textcircled{R} \end{bmatrix} \quad (27)$$

$$R_5 = \begin{bmatrix} \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \textcircled{R} & \textcircled{R} & \textcircled{R} & \textcircled{R} & \times & \times \\ \times & \times & \textcircled{R} & \textcircled{R} & \textcircled{R} & \textcircled{R} & \times & \times \\ \times & \times & \textcircled{R} & \textcircled{R} & \textcircled{R} & \textcircled{R} & \times & \times \\ \times & \times & \textcircled{R} & \textcircled{R} & \textcircled{R} & \textcircled{R} & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \end{bmatrix} \quad (28)$$

$$R_6 = \begin{bmatrix} \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \textcircled{R} & \textcircled{R} & \times & \times & \textcircled{R} & \textcircled{R} \\ \times & \times & \textcircled{R} & \textcircled{R} & \times & \times & \textcircled{R} & \textcircled{R} \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \textcircled{R} & \textcircled{R} & \times & \times & \textcircled{R} & \textcircled{R} \\ \times & \times & \textcircled{R} & \textcircled{R} & \times & \times & \textcircled{R} & \textcircled{R} \end{bmatrix} \quad (29)$$

其中, \textcircled{R} 表示基于对应子阵列天线所计算出的协方差项, \times 表示基于对应的子阵列天线所无法计算出来的协方差项。

基于式 (24) ~ 式 (29) 可合成等效大阵列天线输出协方差矩阵 R_{eq} , 此时, R_{eq} 为 8×8 维方阵。

从式 (23) 可知, 在合成协方差矩阵 R_{eq} 时, R_{eq} 中的每一项都是在子阵列协方差矩阵中第一次出现的。实质上, 还可以采用其他方案处理重复出现的协方差项。处理方法归纳如下:

(1) 采用第一次计算出的协方差项, 舍弃后续子阵列输出协方差矩阵中重复出现的值:

$$r_{m,n} = r_{mm} (1) \quad (30)$$

式 (30) 中, $r_{mm} (1)$ 是在全部子阵列输出协方差矩阵中首次计算出的协方差项。本文实例采用了这种方案。

(2) 取各子阵列天线输出协方差矩阵计算结果的算术

平均值作为等效大阵列天线输出协方差矩阵对应项的值:

$$r_{m,n} = \frac{1}{P} \sum_{l=1}^P r_{m,n} (l) \quad (31)$$

式 (31) 中, P 表示协方差项 $r_{m,n}$ 在全部子阵列天线输出协方差矩阵中出现的总次数。

(3) 取每个等效协方差矩阵的平均值作为等效协方差矩阵的最终值, 每个等效协方差矩阵 $r_{m,n}$ 的值分别从 $r_{m,n} (l)$ 中遍历选取:

$$R_{eq} = \frac{1}{P} \sum_{l=1}^P R_{eq} (l) \quad (32)$$

P 是取样周期数。

得到等效阵列天线输出的协方差矩阵后, 可基于通常的算法进行 DOA 估计。需要注意的是本文提出的 DOA 估计算法与通常的 DOA 估计算法不同。在信号处理接收机数量比天线阵元数量少的情况下, 采用本算法的 DOA 估计的分辨率不受信号处理接收机数量的限制, 只取决于大阵列天线的孔径。这是因为该 DOA 估计算法是基于等效大阵列天线输出协方差矩阵, 此协方差矩阵的秩取决于大阵列天线的孔径, 即取决于大阵列天线阵元数, 而不是信号接收机的数量。所以, 采用本文提出的 DOA 估计算法的最大可估计信号 DOA 数目高于通常的 DOA 估计算法, 这是本算法的意义所在。

4 子阵列的选择优化

在本文所举的第一个例子中, 在大阵列天线包含四个阵元、子阵列天线包含两个阵元的情况下, 需要计算六个子阵列天线的协方差才能得到合成等效大阵列天线输出协方差矩阵所需的全部协方差元素。实质上, 这也是全部可能的子阵列天线数目。在第二个例子中, 大阵列天线包含八个阵元、子阵列天线包含四个阵元。这时, 通过六个子阵列天线输出协方差矩阵也得到了合成等效大阵列天线输出协方差矩阵所需的全部协方差元素。而第二个例子中可能的子阵列天线总数不是六个, 而是七十个。显然, 逐一计算出这些子阵列天线输出的协方差一定能够合成等效大阵列天线输出协方差矩阵, 但实际上是不必要; 另一方面, 容易证明, 少于六个子阵列将无法得到等效大阵列天线输出协方差矩阵中的全部元素。实际上这六个子阵列天线不是任意选择的, 而是优化选择的结果; 对等效大阵列协方差矩阵而言, 它们构成一个最小完备集。子阵列的选择优化涉及到以下几个方面:

(1) 在遍历选择阵元的情况下, 如何计算子阵列天线的数目。

(2) 如何计算最小完备集中的子阵列天线数。

(3) 如何选择阵元才能构成最小完备集。

在 M 个阵元, M_r 个接收机的情况下, 每个子阵列天线包含的阵元数等于 M_r 。

在遍历选取阵元的情况下, 从大阵列天线的 M 个阵元

中随机选择 M_r 个, 可以组成一个子阵列. 因此, 子阵列总数 $L = C_M^{M_r}$ 个. 每个子阵列输出协方差矩阵的元素数为 M_r^2 个. 全部子阵列天线输出协方差矩阵的元素数为 $M_r^2 \times C_M^{M_r}$. 等效大阵列天线输出协方差矩阵的元素数为 M^2 . 由于阵元选择具有对称性, 所有阵元被选中的概率相等. 这时, 等效大阵列输出协方差矩阵中的每个元素的平均计算次数为 $\frac{M_r}{M} \times C_M^{M_r-1}$. 随着大阵列天线阵元数 M 的增加, 子阵列天线的数量 $C_M^{M_r}$ 迅速增加, 等效大阵列天线输出协方差矩阵元素的冗余计算量也迅速增加, 因而信号处理的总计算量将随之增加. 若对每一个子阵列天线的输出进行 N 次相关取样, 以减小噪声对协方差的扰动影响, 则计算一次等效大阵列天线输出协方差矩阵需要的快拍取样数为 $N \times L$.

为了减小运算量, 一般不采用遍历选取阵元的方法, 而总是采用优化选取阵元的方法构造子阵列天线. 这样, 合成等效大阵列天线输出协方差矩阵时, 不需要计算全部可能的子阵列天线输出协方差矩阵, 只需选择其中一部分子阵列, 计算其协方差矩阵即可. 优化选取阵元的方案应使所需的子阵列天线数较少, 这些子阵列天线应构成一个完备子集. 遍历全部两两阵元组. 为此, 我们提出下列优化选取阵元准则:

- (1) 任意一个阵元在不同子阵列出现的次数尽量少.
- (2) 任意两个阵元至少在一个子阵列中同时出现一次.
- (3) 在 M_r 可选的条件下, 取其最大值.
- (4) 在 M_r 确定的条件下, 基于组合设计理论选取阵元, 使完备子集成为最小子集.

研究表明, 分组选择阵元的方案是一种优化选取阵元方案.

在一般情况下, 不存在最小完备集中的子阵列天线数的通用计算公式. 但是, 存在最小完备子集子阵列天线数的下确界公式.

$$K_{\inf}(M, M_r) = \begin{cases} \left(\frac{M}{M_r}\right)^2, & M \text{ 整除 } M_r, \\ \left\lceil \frac{M}{M_r} \right\rceil^2 + 1, & M \text{ 不能整除 } M_r, \end{cases} \quad (33)$$

其中, 方括号下标“ \lceil ”表示取括号内计算结果的整数部分. 对某些特定阵元数的阵列天线, 可以得到更为准确的计算公式.

利用阵列协方差矩阵的共轭对称性, 可以选择出子阵列天线阵元, 组成优化子阵列天线子集.

5 仿真结果及分析

采用基本 MUSIC 算法对基于子阵列合成的 DOA 估计算法的性能进行仿真.

仿真条件: ULA, 波达信号数 $P = 6$ 大阵列天线包含 $M = 8$ 阵元, 子阵列天线包含 $M_r = 4$ 个阵元;

图 2~5 是在不同条件下的仿真结果曲线. 每组图由上

下两个仿真曲线组成. 上面的图是直接通过 8 阵元估计的结果, 下面的图是通过子阵列合成实现 DOA 估计的仿真结果.

图 2 快拍数 $N = 200$ 信噪比 $\text{SNR} = 35 \text{ dB}$ 波达角 $[\theta_p, p = 1, \dots, 6] = -15^\circ, 3^\circ, 8^\circ, 18^\circ, 29^\circ, 38^\circ$; 图 3 $N = 800$ $\text{SNR} = 35 \text{ dB}$ $[\theta_p, p = 1, \dots, 6] = -15^\circ, 3^\circ, 6^\circ, 18^\circ, 29^\circ, 38^\circ$; 图 4 $N = 200$ $\text{SNR} = 45 \text{ dB}$ $[\theta_p, p = 1, \dots, 6] = -15^\circ, 3^\circ, 6^\circ, 18^\circ, 29^\circ, 38^\circ$; 图 5 $N = 200$ $\text{SNR} = 30 \text{ dB}$ $[\theta_p, p = 1, \dots, 6] = -15^\circ, 3^\circ, 8^\circ, 18^\circ, 29^\circ, 60^\circ$.

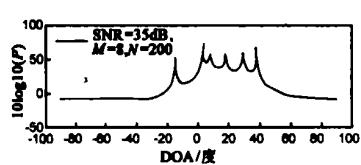
仿真结果分析:

图 2 中, 波达角间隔较大, 在中等信噪比和快拍数条件下可以准确地估计出波达角, 表明基于子阵列合成的 DOA 估计算法是有效的; 还可以看出, 波达角间隔越大, 基于子阵列合成的估计结果与直接估计的结果越接近.

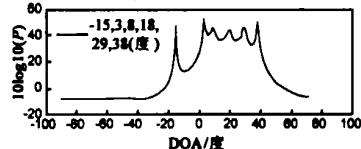
图 3 中, 有两个信号波达角间隔较小, 一个为 3° , 另一个是 6° . 仿真结果表明, 在同样信噪比条件下, 增加快拍数可以提高 DOA 估计分辨率. 这是因为增加快拍样本数可以更好地平滑噪声, 减弱噪声对估计精度的影响. 在这种条件下, 与直接 DOA 估计算法类似, 基于子阵列合成的 DOA 估计算法可以分辨出不同的波达信号.

图 4 中, 波达角间隔同图 3 所示; 减少快拍数, 同时增大信噪比, 仿真结果表明在高信噪比条件下, 较少的快拍数就可以分辨出相距较近的波达方向.

图 5 中, 有一个波达角最大值较大, 取 60° . $N = 200$ $\text{SNR} = 30 \text{ dB}$ 从仿真曲线可知, 在相对较小的信噪比下就可以分辨所有的信号方向. 波达角对 DOA 分辨率有一定影

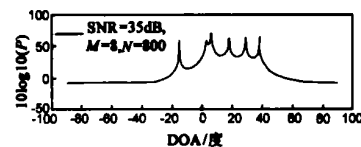


(a) 基于子阵列直接估计

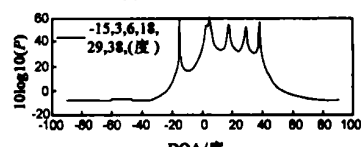


(b) 基于子阵列合成

图 2 DOA 估计性能仿真 1

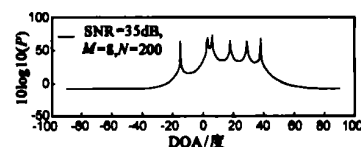


(a) 基于子阵列直接估计

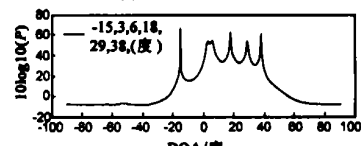


(b) 基于子阵列合成

图 3 DOA 估计性能仿真 2



(a) 基于子阵列直接估计



(b) 基于子阵列合成

图 4 DOA 估计性能仿真 3

响: 波达角增大, DOA 估计分辨率下降。

从仿真曲线图 2 ~ 图 5 可知, 基于子阵列合成的 DOA 算法在上述各种条件下是有效的; 在波达信号满足广义平稳、遍历的条件下, 与直接采用大阵列输出协方差矩阵进行 DOA 估计的仿真结果是相近的。

6 结束语

本文在分析星载系统 DOA 估计特点的基础上, 提出了一种基于子阵列合成的 DOA 估计算法。该算法满足星载系统的要求, 用少量信号处理接收机实现对大阵列天线输出信号的处理, 基于子阵列合成等效大阵列天线输出协方差矩阵, 通过等效协方差矩阵实现 DOA 估计。仿真结果表明所提出的算法是有效的。同时, 提出了选取子阵列的优化准则。基于优化准则所形成的完备子集包含较少的子阵列天线, 因此可以降低 DOA 估计的计算量。

在一定条件下, 本文提出的算法不仅适用于卫星通信系统, 同样可应用于其他如机载雷达、无线通信系统等信号处理领域。

算法采用时分复用处理方式, 存在一定处理时延。然而, 在取样频率较高的情况下, 这一处理时延对系统性能的影响可以忽略。

致谢 本文第一作者向好友梁欣博士致以由衷的感谢。在几年的学习过程中, 作者在多方面得到他的帮助、激励和启迪。

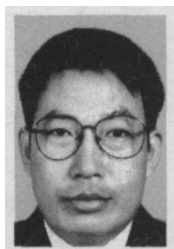
参考文献:

- [1] JA Cadzow. Multiple source location the signal subspace approach[J]. IEEE, 1990 ASSP-38(7): 1110-1125
- [2] R Schmidt Multiple emitter location and signal parameter estimation[J]. IEEE Trans, 1986 AP-34(3): 276-280
- [3] D G Manolakis V K Ingle, SM Kogon Statistical and

Adaptive Signal Processing [M]. USA: McGraw-Hill Companies Inc, 2000 484-488

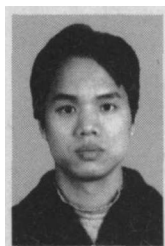
- [4] I Ziskind, M Wax Maximum likelihood location of multiple sources by alternating projection[J]. IEEE Trans 1988, ASSP-36(10): 1553-1560
- [5] J Xin, A Sano Direction estimation of coherent signal using spatial signature[J]. IEEE Signal Processing Letters 2002, 9(12): 414-417.
- [6] 张贤达, 保铮. 通信信号处理 [M]. 北京: 国防工业出版社, 2000 316-331.
- Xianda Zhang, Zheng Bao Communications Signal Processing [M]. Beijing National Defense Industry Publishing House 2000 316-331 (in Chinese)
- [7] J Xin, Sano A. Computationally efficient subspace-based method for direction-of-arrival estimation without Eigen decomposition[J]. IEEE Trans 2004, SP-52(4): 876-893

作者简介:



许文龙 男, 博士研究生, 北京大学电子学系, 主要研究方向: 数字通信, 卫星通信, 通信中的信号处理, 阵列信号处理。

E-mail wku@public.bta.net.cn xw@pku.edu.cn



蒋伟 男, 博士, 北京大学电子学系, 主要研究方向: 数字通信, 卫星通信, 通信中的信号处理。

E-mail jiangwei@ele.pku.edu.cn

尚勇 男, 博士后, 副教授, 北京大学电子学系, 主要研究方向: 阵列信号处理, 数字通信, 高速并行信号处理。

E-mail shangyong@ele.pku.edu.cn

项海格 男, 教授, 博士生导师, 学术带头人, 北京大学电子学系, 主要研究方向: 卫星通信, 数字通信, 网络通信, 通信中的信号处理。E-mail xianghe@pku.edu.cn