

利用导向向量旋转和联合迭代优化的 自适应波束形成算法研究^{*}

谢建平^{1†} 王 瑞² 何熊熊² 李 胜²

(1 丽水学院 工程与设计学院 丽水 323000)

(2 浙江工业大学 信息工程学院 杭州 310023)

2014 年 12 月 6 日收到

2015 年 7 月 15 日定稿

摘要 针对大型阵列中自适应波束形成技术的实时性和鲁棒性问题, 基于最小方差无失真响应 (Minimum Variance Distortionless Response, MVDR) 波束形成的信号模型框架, 提出一种通过对导向矢量进行处理以降低干扰的自适应波束形成算法——稳健联合迭代优化-导向自适应 (Robust Joint Iterative Optimization-Direction Adaptive, RJIO-DA) 算法。在联合迭代优化的基础上, 将降维变换矩阵的每一个列向量看作独立的方向向量, 引导子空间内每一个维度上的权值迭代, 同时旋转导向向量, 减小了由于导向误差的不确定性而导致的性能下降。仿真实验结果表明, 与现有的降维算法相比, RJIO-DA 算法计算复杂度低、收敛率高、鲁棒性好, 可在期望方向上稳健地聚集波束, 更好地形成干扰方向的自适应零陷。
PACS 数: 43.30, 43.60

Adaptive beamforming algorithm based on direction vector rotation and joint iterative optimization

XIE Jianping¹ WANG Rui² HE Xiongxiang² LI Sheng²

(1 Lishui University Lishui 323000)

(2 Zhejiang University of Technology Hangzhou 310023)

Received Dec. 6, 2014

Revised Jul. 15, 2015

Abstract A robust reduced-rank algorithm named RJIO-DA (Robust Joint Iterative Optimization-Direction Adaptive) is proposed for adaptive beamforming in large array scenarios. Based on a MVDR (Minimum Variance Distortionless Response) framework, the proposed algorithm jointly updates a transforming matrix and a reduced-rank filter. In order to reduce the complexity and improve the performance, each column of the transforming matrix is regarded as a direction vector on each dimension of the subspace and is updated individually. In addition, the limited uncertainties caused by the direction error can be overcome with the proposed algorithm. The simulation results show that RJIO-DA algorithm has lower complexity and faster convergence compared with the conventional reduced-rank algorithms.

DOI:10.15949/j.cnki.0371-0025.2016.03.009

引言

自适应波束形成技术是阵列信号处理领域的关键技术之一, 广泛应用于雷达、通信、声呐等领域^[1]。为获得较高的阵列增益和较窄主瓣的波束,

实际中广泛采用大型天线阵列。但是实际系统中用于训练得到稳态的快拍数往往会受到限制, 故而在移动通信环境中, 需要利用快速收敛的波束形成算法来提高波束跟踪的实时性^[1]。另一方面, 自适应波束形成对于误差非常敏感, 例如波达角/定位的误差^[2], 局部散射^[3], 信号传播媒质的不均匀^[4], 天线外

^{*} 国家科技支撑计划课题 (2013BAF07B03) 和浙江省自然科学基金 (LY13F010009) 资助

[†] 通讯作者: 谢建平, Email: xjp1386@hotmail.com

形缺陷引起的失真^[5], 远近效应^[6]等因素, 都会导致波束严重的性能衰落。因此, 复杂的大型阵列中, 依旧保持良好的实时性和鲁棒性, 是当前研究的两大重要课题。

针对高维度接收阵列的条件下波束形成的实时性, 许多学者主要致力于研究阵列信号的降维处理。降维处理是将接收到的高维信号投射到低维子空间, 通过低维子空间的优化达到快速收敛。较为成熟的降维处理算法通常分为三大类: 基于特征分解的方法、基于 Krylov 子空间的方法和联合迭代优化方法 (JIO, Joint iterative optimization)。基于特征分解的方法主要包括特征干扰相消、主分量求逆法^[7]、正交投影^[8]、交叉谱法^[9]等。此类方法是基于接受信号的协方差矩阵的特征分解, 每次优化需遍历整个信号子空间, 并且对维估计有严格要求, 不易实现。基于 Krylov 子空间的方法包括 Power of R (POR)^[10]、降维多级维纳滤波 (MSWF, Multistage Wiener filter)^[11]、辅助向量滤波 (AVF, Auxiliary vector filtering)^[12]方法等。此类方法不需特征分解, 且降维性能优于基于特征分解的方法, 但是高复杂度依然有待解决。联合迭代优化 (JIO)^[13-14]方法对降维变换矩阵和权值进行联合迭代, 获得更好的性能。较现有的众多降维算法, JIO 算法的算法复杂度和鲁棒性有所改进。阵列信号降维处理的关键在于对降维变换矩阵的设计与优化。文献 14 中提出一种基于 CCM (Constrained constant modulus) 准则的 JIO 方案, 应用于 DS-UWB (Direct-sequence ultra-wideband) 领域, 这一方式在处理数据的维度很高时也能够提供很好的收敛性能, 速度快, 且复杂度不高。最小方差无失真响应 (Minimum variance distortionless response, MVDR) 是一种在波束形成领域广泛使用的抗干扰自适应波束形成技术^[15-18]。本文研究的主要内容是高鲁棒性、高收敛速度的波束形成算法, 因此, 提出了一种新颖的基于 MVDR 波束形成的 JIO-DA (Joint iterative optimization-direction adaptive) 算法, 在 JIO 算法的基础上, 将降维变换矩阵的每一个列向量看作独立的方向向量^[19], 引导子空间内的每一个维度的权值迭代, 在降低复杂度的同时提高自适应波束形成算法的性能, 后续数值模拟的结果证明这一方式的确能够提高收敛的速度, 并且通过复杂度分析可见其复杂度比一些现有的降维算法更低。

针对波束形成的鲁棒性, 各种改进方案时见报道。20 世纪 80 年代, Cox 等人提出^[20]对角加载的思想: 在 Capon 中原有的采样协方差矩阵添加对角因子, 提高其可逆性以增强鲁棒性。90 年代初,

Feldman 和 Chang^[21]等人提出基于特征空间分解的算法, 利用了信号-干扰子空间和噪声子空间的正交性。在此之后, 加宽自适应波束零陷带的思想逐渐发展起来, 针对协方差矩阵的优化也逐渐受到重视。Gershman^[22]等人用修正后的协方差矩阵替代 Capon 中的原有的采样协方差矩阵, 提出数据微分约束方法。此外, Riba^[23]等人设计了一种代替采样协方差矩阵的优化设计方法-锥形协方差矩阵, 但在快速移动的干扰源环境下有一定局限性。21 世纪初, Bell 实验室提出一种基于后验概率值的算法-基于贝叶斯方法^[24]的自适应波束形成。2003 到 2005 年期间, Vorobyov^[25]和 Lorenz 等人在指向误差环境下, 提出基于 worst-case 情况下的性能优化算法, 引入不等式约束, 根据导向误差的失配程度估计出最优的加载因子。为了提高算法的鲁棒性, 本文也在算法设计时就同时考虑导向误差的影响, 提出一种通过对导向矢量处理来降低干扰影响的稳健波束形成算法, 即结合 JIO-DA 算法和稳健算法, 得到一种收敛速度快、鲁棒性好的自适应波束形成算法-RJIO-DA (Robust joint iterative optimization-direction adaptive)。

1 信号模型及波束形成技术背景

为增加信号的利用率并使得信号的波束形状可控, 通常将一组传感器按某种方式在空间不同的位置上排列, 组成一个传感器阵列, 用于信号的接收和提取。一般来说, 构成阵列的阵元可以按照任意的方式进行排列。线性阵列模型是最为简单常用的阵列形式, 如图 1 所示。

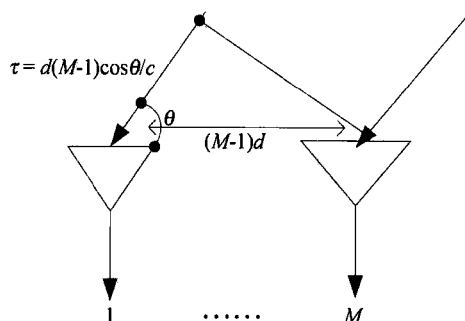


图 1 阵列信号模型

假设阵元等间距 d 分布的 M 元线阵接收一个远场平面单频波窄带信号, 令第一个阵元接收信号 $se^{j\omega t}$, 第 m 个阵元接收信号可表示为 $se^{j\omega(t-\tau_m)}$, 其中 $\tau_m = d(m-1)\cos\theta/c$, θ 为信号入射角度, c 为波速, ω 为信号角频率, s 为信号复幅度。由于讨论的是单频窄带信号, 所以下文公式中忽略 $e^{j\omega t}$ 。如果第 k 个快拍时刻、 P 个信号 $s(k) = [s_1(k), s_2(k), \dots, s_P(k)]^T$ 同时从不同角度 $\theta = [\theta_1, \theta_2, \dots, \theta_P]$ 入射, 第 m 个阵

元输出信号为：

$$x_m(k) = \sum_{i=1}^P s_i(k) e^{-j\omega d(m-1) \cos \theta_i / c} + n_m(k), \quad (1)$$

其中 $n_m(k)$ 为第 m 个阵元、第 k 个快拍的噪声。阵列输出信号向量形式表示为：

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{N}(k),$$

其中：

$$\begin{aligned} \mathbf{A} &= [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_P)], \\ \mathbf{a}(\theta) &= [1, e^{-j\omega d \cos \theta / c}, e^{-j\omega d(M-1) \cos \theta / c}]^T, \\ \mathbf{n}(k) &= [n_1(k), n_2(k), \dots, n_m(k)]^T. \end{aligned}$$

在关注某特定方向 θ_d 信号的阵列信号模型中，信号输出表达式可以表示为：

$$\mathbf{x}(k) = \mathbf{a}(\theta_d)\mathbf{d}(k) + \sum_{j=1}^J \mathbf{a}(\theta_{ij})\mathbf{i}_j(k) + \mathbf{n}(k), \quad (2)$$

其中 $\mathbf{a}(\theta_d)$ 为期望方向的导向向量， $\mathbf{a}(\theta_{ij})$ 为干扰方向的导向向量， $\mathbf{d}(k)$ 为期望信号， $\mathbf{i}_j(k)$ 为干扰信号。

波束形成即输入信号经过全向的阵列天线的整合，使得阵列输出的信号方向增益聚集在期望方向。整合过程通过自适应调整权值的系数完成，令权值向量 $\mathbf{w} = [w_1^T, w_2^T, \dots, w_M^T]^T$ ，阵列输出信号可表示为：

$$y(k) = \mathbf{w}^H \mathbf{x}(k) = \sum_{m=1}^M w_m^* x_m(k). \quad (3)$$

对于权值系数的优化，最小方差无失真 (MVDR) 是最常用的自适应波束形成方法之一，它利用约束条件使得期望方向输出功率不变，同时满足干扰信号和噪声的功率达到最小化：

$$\min \mathbf{w}^H \mathbf{R} \mathbf{w} \quad \text{subject to} \quad \mathbf{w}^H \mathbf{a}(\theta_0) = 1. \quad (4)$$

由拉格朗日对权值向量 \mathbf{w} 求解得到最优权值为：

$$\mathbf{w}_{\text{opt}} = \frac{\mathbf{R}^{-1} \mathbf{a}(\theta_0)}{\mathbf{a}^H(\theta_0) \mathbf{R}^{-1} \mathbf{a}(\theta_0)}, \quad (5)$$

其中 \mathbf{R} 为自相关函数：

$$\begin{aligned} \mathbf{R} &= \mathbb{E}[\mathbf{x}(k) \mathbf{x}^H(k)] = \\ &\mathbb{E}[(\mathbf{A}\mathbf{s}(k) + \mathbf{n}(k))(\mathbf{A}\mathbf{s}(k) + \mathbf{n}(k))^H]. \end{aligned}$$

由于假设噪声子空间与信号子空间正交，经过充分的统计后，自相关函数 \mathbf{R} 可以表示为：

$$\begin{aligned} \mathbf{R} &= \mathbb{E}[\mathbf{A}\mathbf{s}(k) \mathbf{s}(k)^H \mathbf{A}^H] + \sigma_n^2 \mathbf{I} = \\ &\underbrace{\mathbb{E}[|\mathbf{d}(k)|^2] \mathbf{w}_{\text{opt}}^H \mathbf{a}(\theta_d) \mathbf{a}(\theta_d)^H}_{R_s} + \underbrace{\sum_{j=1}^J \mathbb{E}[|\mathbf{i}_j(k)|^2] \mathbf{w}_{\text{opt}}^H \mathbf{a}(\theta_{ij}) \mathbf{a}(\theta_{ij})^H}_{R_{ij}} + \underbrace{\sigma_n^2 \mathbf{I}}_{R_n}. \end{aligned} \quad (6)$$

在后续的实验仿真过程中，采用随机的 sign 信号， $\mathbf{s}(k) \mathbf{s}^H(k)$ 为一个定值， \mathbf{w}_{opt} 可以准确求解，且所用阵列为半波长间隔均匀线阵列。

信干噪声比 (SINR) 表征期望信号的功率与干扰信号 (包括噪声加上干扰) 的功率之比，使用 \mathbf{w}_{opt} 开展波束形成，得到的波束输出的信干噪声比 (SINR) 为 $\text{SINR}_{\text{opt}} = \mathbf{w}_{\text{opt}}^H \mathbf{R}_s \mathbf{w}_{\text{opt}} / \mathbf{w}_{\text{opt}}^H \mathbf{R}_{i+n} \mathbf{w}_{\text{opt}}$ ，其中 \mathbf{R}_s 和 $\mathbf{R}_{i+n} = \mathbf{R}_{ij} + \mathbf{R}_n$ 的值参考式 (6)。

2 JIO 降维处理及改进方案

下面将首先介绍传统的 JIO 降维处理框架，然后给出改进后的 JIO-DA 方法，并分析其计算复杂度。

2.1 JIO 框架

降维处理是通过设计降维变换矩阵 $\mathbf{T} \in \mathbb{C}^{M \times D}$ 将接收到的 M 维信号 $\mathbf{x}(k)$ 投射到 D 维子空间 $\bar{\mathbf{x}}(k) = \mathbf{T}^H(k) \mathbf{x}(k)$ ，其中 $D \ll M$ ，设计权值系数 $\bar{\mathbf{w}}(k)$ 对 $\bar{\mathbf{x}}(k)$ 在 D 维子空间进行优化。降维算法将高维空间的参数更新简化为低维子空间中的优化，由于权重系数数量的大幅降低，算法的收敛速度得到了大幅提高。在低维子空间中，基于 MVDR 波束形成算法的优化过程可表示为：

$$J = \mathbb{E} [|\bar{\mathbf{w}}^H(k) \mathbf{T}^H(k) \mathbf{x}(k)|^2] \quad \text{s.t.} \quad \bar{\mathbf{w}}^H(k) \bar{\mathbf{a}}(\theta_0) = 1,$$

其中导向向量 $\bar{\mathbf{a}}(\theta_0) = \mathbf{T}^H(k) \mathbf{a}(\theta_0)$ 。

将上述代价函数展开，利用拉格朗日最小二乘法可以得到需要优化的代价函数：

$$L = \mathbb{E} [|\bar{y}(k)|^2] + 2\Re [\lambda (\bar{\mathbf{w}}^H(k) \bar{\mathbf{a}}(\theta_0) - 1)], \quad (7)$$

其中 $\bar{y}(k) = \bar{\mathbf{w}}^H(k) \bar{\mathbf{x}}(k) = \bar{\mathbf{w}}^H(k) \mathbf{T}^H(k) \mathbf{x}(k)$ 。

如图 2 所示，JIO 算法即联合迭代降维变换矩阵 \mathbf{T} 和权值 $\bar{\mathbf{w}}$ 。这一方法的目标是给定一个随机的初始值，通过联合的迭代优化，逐步逼近降维变换矩阵和权值的最优值。常规 JIO 算法实现过程中，每次迭代首先优化降维变换矩阵，之后利用更新后的变换矩阵更新权值，算法有着很好的收敛特性。联合优化过程如下。

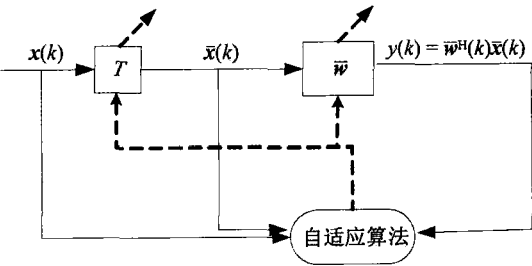


图 2 JIO 降维框架图

首先假设 \bar{w} 不变, 对降维变换矩阵 T 进行优化, 此时将代价函数式 (7) 视为变换矩阵的函数, 对其求导并求出最小值, 得到的结果表示为:

$$T(k) = \frac{R^{-1}a(\theta_0)\bar{w}^H(k)\bar{R}_{\bar{w}}^{-1}}{\bar{w}^H(k)\bar{R}_{\bar{w}}^{-1}\bar{w}(k)a^H(\theta_0)R^{-1}a(\theta_0)}, \quad (8)$$

其中 $R = E[x(k)x^H(k)]$, $\bar{R}_w = E[\bar{w}(k)\bar{w}^H(k)]$.

更新了降维变换矩阵之后, 再假设 T 不变, 将代价函数式 (7) 视为权值的函数对其求导并求出最小值, 求得 \bar{w} 的表达式:

$$\bar{w}(k) = \frac{\bar{R}^{-1}a(\theta_0)}{a^H(\theta_0)\bar{R}^{-1}a(\theta_0)}, \quad (9)$$

其中 $\bar{R} = E[\bar{x}(k)\bar{x}^H(k)] = T^H R T$.

可见, T 的迭代与 \bar{w} 相关联, \bar{w} 的迭代与 T 相关联, 利用 JIO 算法联合迭代可以得到最优解.

2.2 JIO-DA 框架

传统 JIO 算法将降维变换矩阵视为一个整体, 每次都对维数很大的矩阵进行更新, 所需的计算复杂度非常高. 尽管低维子空间内部的优化过程在降维算法中得到了简化, 但是高维到低维的投射过程依旧存在高复杂性. 因此, 设计一个有效的 T 是获得高收敛速率、降低复杂度的关键.

JIO 方案中, $T \in \mathbb{C}^{M \times D}$ 作为一个完整的矩阵参数. 但投影过程可以看成是将一个高维数向量投影到一个低维度的子空间中去, 降维变换矩阵的每

一列可以看作是低维度子空间的一个方向向量, 当设计的子空间足够有效时, 每一个方向向量都有着一定的独立性, 因此, 本节将对 T 的整体更新转化成 D 列独立的列向量, 即表示矩阵 $T(k) = [T_1(k), T_2(k), \dots, T_D(k)]$, 每一个列向量被看作独立的方向向量, 引导子空间内的每一个维度上的权值迭代. 在每个方向向量的优化过程中, 能更迅速地产生独立的最优降维变换矩阵方向向量, 从而进一步加速收敛.

定义最小二乘法 (RLS) 算法的约束代价函数来联合优化 JIO-DA 算法:

$$L = \sum_{l=1}^k \alpha^{k-l} |\bar{w}^H(k)T^H(k)x(l)|^2 + \Re \left[\lambda (\bar{w}^H(k) \sum_{d=1}^D t_d^H(k)a(\theta_0)q_d - 1) \right], \quad (10)$$

其中 α 为遗忘因子, 阵列输出信号可以表示为:

$$y(k) = \bar{w}^H(k) \sum_{d=1}^D t_d^H(k)x(k)q_d = \bar{w}_d^*(k)\bar{x}_d(k) + \bar{w}^H(k)\bar{x}_e(k), \quad (11)$$

其中 q_d 是第 d 行为 1 而其它元素是 0 的向量, $\bar{x}(k)$ 的第 d 行元素为 $\bar{x}_d(k) = T_d^H(k)x(k)$, $\bar{w}_d(k)$ 是 $\bar{w}(k)$ 的第 d 行数据, $\bar{x}_e(k)$ 是将 $\bar{x}(k)$ 第 d 行取零而其余不变的向量.

根据代价函数式 (10) 对 $t_d(k)$ 求导:

$$\begin{aligned} \nabla L &= \sum_{i=1}^k \alpha^{k-i} y^*(i) \bar{w}_d^*(k) x(i) + \lambda_{t,d}(k) \bar{w}_d^*(k) a(\theta_0) = \\ &= \sum_{i=1}^k \alpha^{k-i} (\bar{w}_d^*(k) \bar{x}_d(i) + \bar{w}^H(k) \bar{x}_e(i))^* \bar{w}_d^*(k) x(i) + \lambda_{t,d}(k) \bar{w}_d^*(k) a(\theta_0) = \\ &= \sum_{i=1}^k \alpha^{k-i} x(i) x^H(i) t_d(k) |\bar{w}_d^*(k)|^2 + x(i) \bar{w}_d^*(k) \bar{x}_e^H(i) \bar{w}(k) + \lambda_{t,d}(k) \bar{w}_d^*(k) a(\theta_0), \end{aligned}$$

其中 $\lambda_{t,d}(k)$ 为复数拉格朗日乘子.

令求导后的代价函数等于零, 求得:

$$t_d(k) = -\hat{R}_d^{-1}(k) (\lambda_{t,d}(k) \bar{w}_d^*(k) a(\theta_0) + v_r(k)), \quad (12)$$

其中:

$$\hat{R}_d(k) = \sum_{i=1}^k \alpha^{k-i} x(i) x^H(i) |\bar{w}_d^*(k)|^2 = \hat{R}(k) |\bar{w}_d^*(k)|^2.$$

由于 $\hat{R}(k) = \alpha \hat{R}(k-1) + x(k) x^H(k)$, 令 $\hat{\Phi}(k) = \hat{R}^{-1}(k)$, 利用矩阵求逆原理求得时变增益向量 $P(k)$

和协相关矩阵的逆 $\hat{\Phi}(k)$:

$$P(k) = \frac{\alpha^{-1} \hat{\Phi}(k-1) x(k)}{1 + \alpha^{-1} x^H(k) \hat{\Phi}(k-1) x(k)},$$

$$\hat{\Phi}(k) = \frac{1}{\alpha} [\hat{\Phi}(k-1) + P(k) x^H(k) \hat{\Phi}(k-1)].$$

而式 (12) 中 $v_r(k)$ 则表示为:

$$v_r(k) = \sum_{l=1}^k \alpha^{k-l} x(l) \bar{w}_d^*(k) \bar{x}_e^H(l) \bar{w}(k). \quad (13)$$

表 1 复杂度比较

算法	加法	乘法
Full-rank RLS	$4M^2 - M - 1$	$5M^2 + 5M - 1$
MSWF-RLS	$DM^2 + (D + 1)M + 4D^2 - 3D - 1$	$(D + 1)M^2 + 2DM + 5D^2 + 4D$
AVF-RLS	$(4D + 5)M^2 + (D - 1)M - 2D - 1$	$(5D + 8)M^2 + (3D + 2)M$
JIO-DA-RLS	$(D + 2)M^2 + 6DM - M + 5D^2 - 5D$	$(D + 4)M^2 + 10DM + 2M + 5D^2 + 8D$

由于 $v_r(k)$ 在每一个快拍数都需要复杂的重复运算, 对式 (13) 简化, 使其在前一刻的基础上递归更新:

$$v_r(k) \approx \alpha v_r(k-1) + x(k)\overline{w}_d^*(k)\overline{x}_e^H(k)\overline{w}(k).$$

(14)

将 $\widehat{\Phi}(k)$ 代入式 (12) 中, 用 $\widehat{\Phi}(k)$ 表示 $t_d(k)$, 得:

$$t_d(k) = \frac{\widehat{\Phi}(k) [\lambda_{t,d}(k)\overline{w}_d^*(k)\alpha(\theta_0) + v_r(k)]}{-|\overline{w}_d^*(k)|^2}.$$

(15)

由式 (15) 可得, $|\overline{w}_d^*(k)|$ 数值较小, $|\overline{w}_d^*(k)|$ 的轻微变动会引起 $t_d(k)$ 在后续运算中较大的变化。因此后续的迭代过程中对 $t_d(k)$ 进行归一化 $t_d(k) = t_d(k)/\|t_d(k)\|$ 。

同时, 利用约束条件:

$$\overline{w}^H(k)\mathbf{T}^H(k)\mathbf{a}(\theta_0) = \overline{w}^H(k) \sum_{d=1}^D t_d^H(k)\mathbf{a}(\theta_0)\mathbf{q}_d = 1,$$

得到:

$$\lambda_{t,d}(k) = \left[\frac{I + II}{-\overline{w}_d^*(k)\mathbf{a}^H(\theta_0)\widehat{\Phi}(k)\mathbf{a}(\theta_0)} \right]^*,$$

其中:

$$I = \overline{w}_d^*(k)\mathbf{V}r^H(k)\widehat{\Phi}(k)\mathbf{a}(\theta_0),$$
$$II = [1 - \overline{w}^H(k)\overline{\mathbf{a}}_e(\theta_0)] |\overline{w}_d^*(k)|^2,$$

权值更新公式为:

$$\overline{w}(k) = \frac{\widehat{\Phi}(k)\overline{\mathbf{a}}(\theta_0)}{\overline{\mathbf{a}}^H(\theta_0)\widehat{\Phi}(k)\overline{\mathbf{a}}(\theta_0)},$$

(16)

其中:

$$\widehat{\Phi}(k) = \widehat{\mathbf{R}}_x^{-1}(k)\widehat{\mathbf{R}}(k) = \sum_{l=1}^k \alpha^{k-l}\overline{\mathbf{x}}(l)\overline{\mathbf{x}}^H(l).$$

同理可以利用矩阵求逆原理求得时变增益向量 $\widehat{\mathbf{P}}(k)$ 和协相关矩阵的逆 $\widehat{\Phi}(k)$ 。

2.3 复杂度分析

本节对新提出的 JIO-DA-RLS 算法与满维的 RLS 算法、基于 RLS 算法的经典降维算法 MSWF 和 AVF 进行了复杂度比较。即将不同算法每一次更新需要进行的复数加法和乘法进行了比较。如表 1 所示, 因为满维空间维数 M 远远大于降维后的子空间维数 D , 所以 D 对复杂度的影响远远小于 M , 通过与现有的满维和降维的算法比较, JIO-DA 算法的复杂度与 MSWF 类似, 但是低于 AVF 算法。

3 导向误差的修正方案

实际应用中, 由于测量误差和干扰的存在, 导向向量往往会有个微小的偏移, 为了进一步提高自适应波束形成的鲁棒性, 本文提出一种能有效改善因导向误差引起的性能下降的算法。在一定误差范围内, 当半波长等间距分布的 M 阵元线阵导向向量进行了旋转形成新的导向向量, 假设旋转角度为 $\Delta\theta$ 且 $\Delta\theta \leq 0.8M^{-1[26]}$, 算法依旧要保持系统的工作稳定性。

假设将 θ_0 向左或向右扭转 $\Delta\theta$ 角度分别得到 $\theta_0 + \Delta\theta$ 和 $\theta_0 - \Delta\theta$, 其中 $\Delta\theta$ 小且为固定值, 由此 2 个期望信号的导向向量分别变为:

$$\mathbf{a}(\theta_0 + \Delta\theta) = [1, \exp(j\phi_1), \cdots, \exp(j(M-1)\phi_1)]^T,$$
$$\mathbf{a}(\theta_0 - \Delta\theta) = [1, \exp(j\phi_2), \cdots, \exp(j(M-1)\phi_2)]^T,$$

(17)

其中 2 个导向矢量的相位偏差 ϕ_1 和 ϕ_2 分别为 $\phi_1 = \pi \cos(\theta_0 + \Delta\theta)$, $\phi_2 = \pi \cos(\theta_0 - \Delta\theta)$ 。

假设期望方向 θ_0 已知, 当 $\Delta\theta$ 非常小时, 那么 $\sin \Delta\theta$ 和 $\cos \Delta\theta$ 都进行近似简化, 则:

$$\phi_1 \approx \pi(\cos \theta_0 - \Delta\theta \sin \theta_0) = \phi - \Delta\phi,$$
$$\phi_2 \approx \pi(\cos \theta_0 + \Delta\theta \sin \theta_0) = \phi + \Delta\phi,$$

其中 $\phi = \pi \cos \theta_0$, $\Delta\phi = \pi \sin \theta_0$ 。

因此可获得 2 个新的导向矢量:

$$\mathbf{a}(\theta_0 + \Delta\theta) \approx [1, \exp(j\phi) \exp(-j\Delta\phi), \cdots, \exp(j(M-1)\phi) \exp(-j(M-1)\Delta\phi)]^T \triangleq \mathbf{B}^H \mathbf{a}(\theta_0),$$
$$\mathbf{a}(\theta_0 - \Delta\theta) \approx [1, \exp(j\phi) \exp(j\Delta\phi), \cdots, \exp(j(M-1)\phi) \exp(j(M-1)\Delta\phi)]^T \triangleq \mathbf{B} \mathbf{a}(\theta_0).$$

其中:

$$B = \text{diag}\{1, \exp(j\Delta\phi), \cdots, \exp(j(M-1)\Delta\phi)\}.$$

最后 $\boldsymbol{a}(\theta_0 - \Delta\theta)$ 和 $\boldsymbol{a}(\theta_0 + \Delta\theta)$ 进行平均, 得到新的阵列导向矢量:

$$\begin{aligned} \boldsymbol{b}(\theta_0) &= \frac{1}{2}[\boldsymbol{a}(\theta_0 + \Delta\theta) + \boldsymbol{a}(\theta_0 - \Delta\theta)] \approx \\ &\frac{1}{2}\{\boldsymbol{B}^H + \boldsymbol{B}\}\boldsymbol{a}(\theta_0). \end{aligned} \tag{18}$$

接下来用 $\boldsymbol{b}(\theta_0)$ 代替 MVDR 波束形成算法中的 $\boldsymbol{a}(\theta_0)$, 与快速降维 JIO-DA 算法相结合, 得到快速稳健的 RJIO-DA 自适应波束形成算法。在更新降维变换矩阵 $\boldsymbol{T}(k)$ 的基础上, 更新得到新的导向向量 $\bar{\boldsymbol{b}}(\theta_0) = \boldsymbol{T}^H(k+1)\boldsymbol{b}(\theta_0)$ 和 $\bar{\boldsymbol{x}}(k) = \boldsymbol{T}^H(k+1)\boldsymbol{x}(k)$, 同样的, 分别更新时变增益向量 $\bar{\boldsymbol{P}}(k)$ 和相关矩阵的逆

矩阵 $\hat{\boldsymbol{\Phi}}(k)$:

$$\begin{aligned} \bar{\boldsymbol{P}}(k) &= \frac{\alpha^{-1}\hat{\boldsymbol{\Phi}}(k-1)\bar{\boldsymbol{x}}(k)}{1 + \alpha^{-1}\bar{\boldsymbol{x}}^H(k)\hat{\boldsymbol{\Phi}}(k-1)\bar{\boldsymbol{x}}(k)}, \\ \hat{\boldsymbol{\Phi}}(k) &= \alpha^{-1}\hat{\boldsymbol{\Phi}}(k-1) + \alpha^{-1}\bar{\boldsymbol{P}}(k)\bar{\boldsymbol{x}}^H(k)\hat{\boldsymbol{\Phi}}(k-1). \end{aligned}$$

根据已知的 $\hat{\boldsymbol{\Phi}}(k)$, 迭代得到:

$$\bar{\boldsymbol{w}}(k+1) = \frac{\hat{\boldsymbol{\Phi}}(k)\bar{\boldsymbol{b}}(\theta_0)}{\bar{\boldsymbol{b}}^H(\theta_0)\hat{\boldsymbol{\Phi}}(k)\bar{\boldsymbol{b}}(\theta_0)}.$$

此算法能改进导向误差引起的输出性能下降。自适应最佳权值向量表示为:

$$\boldsymbol{w}_{\text{opt}} = \frac{\boldsymbol{R}_x^{-1}\boldsymbol{b}(\theta_0)}{\boldsymbol{b}^H(\theta_0)\boldsymbol{R}_x^{-1}\boldsymbol{b}(\theta_0)}. \tag{19}$$

RJIO-DA-RLS 算法的更新过程如表 2 所示。

表 2 RJIO-DA-RLS 算法

<p>初始化:</p> <p>$\boldsymbol{T}(1) = [\boldsymbol{I}_{D \times D} \quad \boldsymbol{0}_{D \times (M-D)}]^T$, $\bar{\boldsymbol{w}}(1) = [1, 1, 1, \cdots, 1]$, $\hat{\boldsymbol{\Phi}}(0) = \delta \boldsymbol{I}_{M \times M}$, $\hat{\boldsymbol{\Phi}}(0) = \delta \boldsymbol{I}_{D \times D}$,</p> <p>$\boldsymbol{t}_d(1) = [1, 0, 0, \cdots, 0]$ ($d = 1, 2, \cdots, D$), $\Delta\theta = 0.5M^{-1}$.</p>
<p>对导向向量旋转均值后得到 $\boldsymbol{b}(\theta_0)$:</p> <p>$\boldsymbol{a}(\theta_0 + \Delta\theta) \approx [1, \exp(j\phi) \exp(-j\Delta\phi), \cdots, \exp(j(M-1)\phi) \exp(-j(M-1)\Delta\phi)]^T \triangleq \boldsymbol{B}^H \boldsymbol{a}(\theta_0)$,</p> <p>$\boldsymbol{a}(\theta_0 - \Delta\theta) \approx [1, \exp(j\phi) \exp(j\Delta\phi), \cdots, \exp(j(M-1)\phi) \exp(j(M-1)\Delta\phi)]^T \triangleq \boldsymbol{B} \boldsymbol{a}(\theta_0)$,</p> <p>其中: $\boldsymbol{B} = \text{diag}\{1, \exp(j\Delta\phi), \cdots, \exp(j(M-1)\Delta\phi)\}$.</p> <p>求均值后: $\boldsymbol{b}(\theta_0) = \frac{1}{2}[\boldsymbol{a}(\theta_0 + \Delta\theta) + \boldsymbol{a}(\theta_0 - \Delta\theta)] \approx \frac{1}{2}\{\boldsymbol{B}^H + \boldsymbol{B}\}\boldsymbol{a}(\theta_0)$.</p>
<p>迭代降秩变换矩阵 $\boldsymbol{T}(k)$ 过程:</p> <p>随着 $k = 1, 2, \cdots, N$, 分别更新时变增益向量 $\boldsymbol{P}(k)$ 和相关矩阵的逆矩阵 $\hat{\boldsymbol{\Phi}}(k)$:</p> <p>$\boldsymbol{P}(k) = \frac{\alpha^{-1}\hat{\boldsymbol{\Phi}}(k-1)\boldsymbol{x}(k)}{1 + \alpha^{-1}\boldsymbol{x}^H(k)\hat{\boldsymbol{\Phi}}(k-1)\boldsymbol{x}(k)}$, $\hat{\boldsymbol{\Phi}}(k) = \alpha^{-1}\hat{\boldsymbol{\Phi}}(k-1) + \alpha^{-1}\boldsymbol{P}(k)\boldsymbol{x}^H(k)\hat{\boldsymbol{\Phi}}(k-1)$.</p> <p>迭代降秩变换矩阵的列向量:</p> <p>计算阵列输出信号:</p> <p>$\boldsymbol{y}(k) = \bar{\boldsymbol{w}}^H(k) \sum_{d=1}^D \boldsymbol{t}_d^H(k)\boldsymbol{x}(k)\boldsymbol{q}_d = \bar{\boldsymbol{w}}_d^*(k)\bar{\boldsymbol{x}}_d(k) + \bar{\boldsymbol{w}}^H(k)\bar{\boldsymbol{x}}_e(k)$,</p> <p>其中 \boldsymbol{q}_d 是第 d 行为 1, 而其它元素是 0 的向量, $\bar{\boldsymbol{x}}(k)$ 的第 d 行元素为 $\bar{\boldsymbol{x}}_d(k) = \boldsymbol{t}_d^H(k)\boldsymbol{x}(k)$, $\bar{\boldsymbol{w}}_d(k)$ 是 $\bar{\boldsymbol{w}}(k)$ 的第 d 行数据, $\bar{\boldsymbol{x}}_e(k)$ 是将 $\bar{\boldsymbol{x}}(k)$ 第 d 行取零而其余不变的向量。</p> <p>随着 $d = 1, 2, \cdots, D$, 更新 $\boldsymbol{V}r(k) = \alpha \boldsymbol{V}r(k-1) + \boldsymbol{x}(k)\bar{\boldsymbol{w}}_d^*(k)\bar{\boldsymbol{x}}_e^H(k)\bar{\boldsymbol{w}}(k)$, 更新拉格朗日因子</p> <p>$\lambda_{t,d}(k) = \left[\frac{\bar{\boldsymbol{w}}_d^*(k)\boldsymbol{V}r^H(k)\hat{\boldsymbol{\Phi}}(k)\boldsymbol{a}(\theta_0) + (1 - \bar{\boldsymbol{w}}^H(k)\bar{\boldsymbol{a}}_e(\theta_0)) \bar{\boldsymbol{w}}_d^*(k) ^2}{-\bar{\boldsymbol{w}}_d^*(k)\boldsymbol{a}^H(\theta_0)\hat{\boldsymbol{\Phi}}(k)\boldsymbol{a}(\theta_0)} \right]^*$, 迭代得到降秩变换矩阵的列向量:</p> <p>$\boldsymbol{t}_d(k) = -\frac{\hat{\boldsymbol{\Phi}}(k)}{ \bar{\boldsymbol{w}}_d^*(k) ^2}(\lambda_{t,d}(k)\bar{\boldsymbol{w}}_d^*(k)\boldsymbol{a}(\theta_0) + \boldsymbol{V}r(k))$.</p> <p>在完成 $d = 1, 2, \cdots, D$ 的更新后, $\boldsymbol{T}(k) = [\boldsymbol{t}_1(k), \boldsymbol{t}_2(k), \cdots, \boldsymbol{t}_D(k)]$.</p>
<p>迭代权值矢量 $\bar{\boldsymbol{w}}(k)$ 过程:</p> <p>在更新降秩变换矩阵 $\boldsymbol{T}(k)$ 的基础上, 更新得到新的导向向量 $\bar{\boldsymbol{b}}(\theta_0) = \boldsymbol{T}^H(k+1)\boldsymbol{b}(\theta_0)$ 和 $\bar{\boldsymbol{x}}(k) = \boldsymbol{T}^H(k+1)\boldsymbol{x}(k)$, 同样的, 分别更新时变增益向量 $\bar{\boldsymbol{P}}(k)$ 和相关矩阵的逆矩阵 $\hat{\boldsymbol{\Phi}}(k)$:</p> <p>$\bar{\boldsymbol{P}}(k) = \frac{\alpha^{-1}\hat{\boldsymbol{\Phi}}(k-1)\bar{\boldsymbol{x}}(k)}{1 + \alpha^{-1}\bar{\boldsymbol{x}}^H(k)\hat{\boldsymbol{\Phi}}(k-1)\bar{\boldsymbol{x}}(k)}$, $\hat{\boldsymbol{\Phi}}(k) = \alpha^{-1}\hat{\boldsymbol{\Phi}}(k-1) + \alpha^{-1}\bar{\boldsymbol{P}}(k)\bar{\boldsymbol{x}}^H(k)\hat{\boldsymbol{\Phi}}(k-1)$, 根据已知的 $\hat{\boldsymbol{\Phi}}(k)$,</p> <p>迭代得到: $\bar{\boldsymbol{w}}(k+1) = \frac{\hat{\boldsymbol{\Phi}}(k)\bar{\boldsymbol{b}}(\theta_0)}{\bar{\boldsymbol{b}}^H(\theta_0)\hat{\boldsymbol{\Phi}}(k)\bar{\boldsymbol{b}}(\theta_0)}$.</p>

4 仿真实验和结果分析

分别对不同的子空间维度 (D)、阵元数 (M) 和干扰源数 (L) 条件进行实验仿真验证, 通过比较分析不同算法在同一条件下的波束形成图和输出 SINR 图, 验证算法的鲁棒性和实时性。分析波束形成图时, 目标主瓣方向 (本文中统一设定为 0° 方向) 的增益越大、旁瓣的增益越小且变化越平缓为佳; 对于输出 SINR 图, 以 SINR 值越高、收敛速度越快为佳。SINR 的计算公式即文中第 2 部分推导所得。

4.1 JIO-DA 中维的选择

在 JIO-DA 自适应波束形成过程中, 子空间维数 D 对于算法复杂度和波束形成性能有着重大的影响。由于本算法将降维变换矩阵的每一列都看作是子空间中的方向向量, 那么 D 的最优值就是低维度子空间的独立方向向量个数, 当 D 小于这一数量时, 系统性能会由于没有完整投影而下降, 而当 D 大于这一数量时, 每一个方向向量的独立性受到影响, 同时算法复杂度会变高, 进而也会影响性能。而实际应用中, 子空间维数决定于系统工作时的参数, 例如干扰源数量、阵列规模、发射功率和接收端噪声的影响等, 很难被确定。因此, 仿真实验中我们参照文献 14, 将 D 看作一个时变的系统参数, 使用基于后验最小二乘法来对参数 D 的选取进行权衡。在此基础上定义了一个新的代价函数:

$$J_D(i) = \sum_k \lambda_D^{i-k} |\bar{\mathbf{w}}^H(k) \mathbf{T}^H(k) \mathbf{x}(i)|^2.$$

此代价函数中的变量为维度 D , 其中 λ_D 为遗忘因子, 利用这一函数, 对第 i 个时刻进行一次判断, 从而得到此刻在一定范围内的最优子空间维数 D 。具体来说, 首先需要设置一个最大的维度, 这里用 L_{\max} 来表示。之后, 在每次进行自适应更新时, 对 $\mathbf{T}(k)$ 和 $\bar{\mathbf{w}}$ 分别进行扩展得到:

$$\mathbf{T}(k) = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,L_{\min}} & \cdots & t_{1,L_{\max}} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,L_{\min}} & \cdots & t_{2,L_{\max}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{M,1} & t_{M,2} & \cdots & t_{M,L_{\min}} & \cdots & t_{M,L_{\max}} \end{bmatrix},$$
$$\bar{\mathbf{w}}(k) = [\bar{w}_1 \ \bar{w}_2 \ \cdots \ \bar{w}_{L_{\min}} \ \cdots \ \bar{w}_{L_{\max}}].$$

联合相关参数 $\hat{\Phi}(k)$ 和 $\bar{\mathbf{a}}(\theta_0)$, 将扩展后的 $\mathbf{T}(k)$ 和 $\bar{\mathbf{w}}$ 带入代价函数联合迭代, 选取代价函数值最小的那个维度作为此时刻最优的子空间维数, 即:

$$L_{\text{opt}}(k) = \arg \min_{L \in \{L_{\min}, \cdots, L_{\max}\}} J_L(k).$$

为了说明参数 D 对波束形成的影响, 进行以下仿真实验: 仿真平台采用 $M=64, d=\lambda/2$ 的大型均匀 ULA 阵列。假设有 4 组来自不同方向的互不相关的远场窄带信号, 均为随机的 sign 信号, 与加性噪声也不相关。假设期望信号 SOI 来自 0° , $(\sigma_s/\sigma_n)^2 = 10$, 4 个干扰源的方向分别为 $-40^\circ, -20^\circ, 20^\circ, 40^\circ$, $\sigma_s/\sigma_i = 0.1$ 。仿真实验的快拍数为 500, 最终对 500 次试验的结果取均值。

本实验将子空间维数 D 作为一个可变参数研究, 由于系统能够接受的复杂度和计算能力是有限的, 我们关注的范围为 $D_{\min}=3, D_{\max}=8^{[23]}$ 。如图 3 所示, 在本实验的参数设置下, 当 $D=4$ 时, SINR 性能得到最大值, 为了降低算法复杂度, 在后续仿真中我们采用一个固定值 $D=4$ 。通过大量仿真我们发现新算法在给定的 D 范围下都能工作并提供较好的性能, 本文主要研究内容是 JIO-DA 算法框架、导向误差的修正算法, 最优 D 的选取可以作为一个后续的研究方向进行拓展研究。

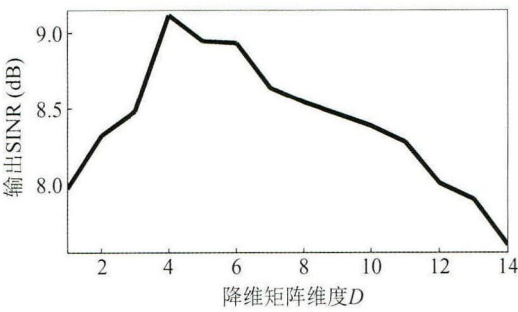


图 3 输出 SINR 性能随着降维矩阵维度 D 的变化图

4.2 不同阵元数 M 环境下的算法比较

测试不同阵元数 M 环境下, RJIO-DA 算法及 JIO-DA 算法与其他降维和满维算法的优势所在。因此, 仿真平台分别采用 $M=32, 64$ 的两种大型均匀 ULA 阵列, 其余实验参数与 4.1 节一致。

仿真结果如图 4 和图 5 所示, 可见降维算法 (MSWF, AVF, JIO-RLS, JIO-DA-RLS, RJIO-DA) 相比于满维算法, 能够更好的聚集期望信号的能量, 即在期望信号方向形成更高的波峰, 同时能更好地抑制旁瓣信号, 即波束形成中的旁瓣高度较低, 且变化较平缓。并且伴随着 M 的增加, 波形形成显得越来越密, 满维算法的旁瓣干扰越来越大, 即最高的旁瓣高度越来越高, 而降维算法在期望方向上的能量愈发集中, 优势愈发明显。

更为重要的一点是, 与其它降维算法比较, RJIO-DA 算法不仅仅能很好地在期望方向 0° 上集中能量, 而且能够实现干扰方向的精确辨识, 确定 4 个干扰源的方向分别为 $-40^\circ, -20^\circ, 20^\circ, 40^\circ$, 形成干扰

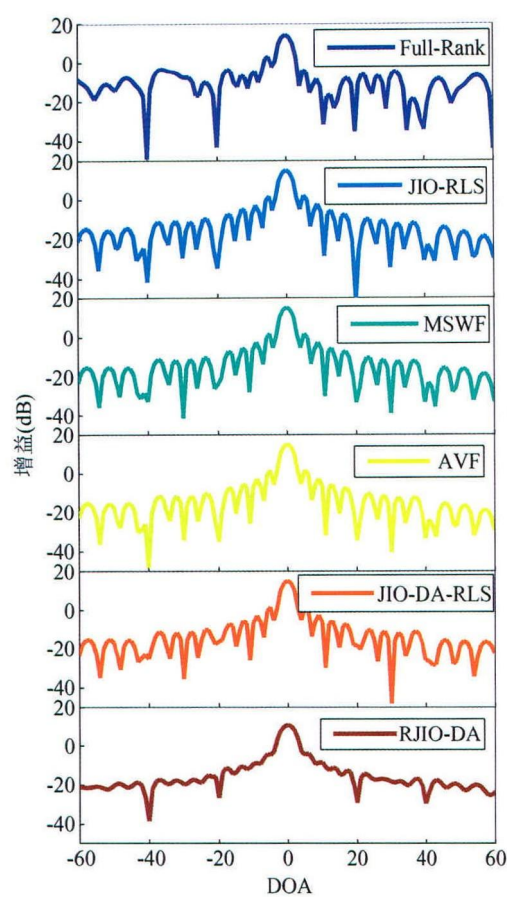


图 4 阵元数为 32 的 RJIO-DA 及现有的降维算法的波束形成比较图

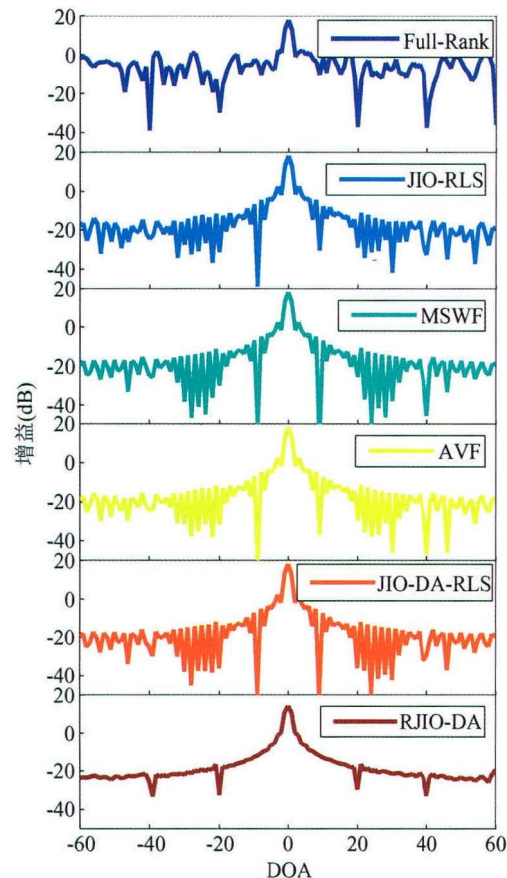


图 5 阵元数为 64 的 RJIO-DA 及现有的降维算法的波束形成比较图

方向上的显著的自适应零陷, 有助于我们排除干扰。

如图 6 所示, 在阵元数 $M=32, 64$ 的大型阵列环境中, 对 RJIO-DA 算法及 JIO-DA 算法与其他降维和满维算法的 SINR 性能进行比较分析。如图 6 可得, 降维算法相比于满维算法, 能更好的实现 SINR 的收敛, 使得 SINR 在较少的快拍数就能收敛到一个稳定的状态, 并且获得较高的 SINR。在降维算法中, 与较成熟的降维算法 (MSWF, AVF) 相比, 在阵元数 M 较小的时候, 例如 $M=32$ 时, JIO-DA-RLS 的性能优势不突出, 但是随着 M 增加, RJIO-DA 算法以及 JIO-DA-RLS 的 SINR 性能获得一个明显的提升, 能够更快速收敛到一个高于 MSWF、AVF 算法的 SINR 值, 其收敛优势越来越明显。对于 RJIO-DA 算法来说, 由于增加了对导向向量偏差的考虑, 能够达到的最优 SINR 性能也获得了一定的提升, 如图中虚直线所示。同时利用了这一模型设计的自适应算法也获得了更佳的收敛性能、更好的稳态 SINR 指标。

4.3 不同干扰源数 L 的算法比较

本小节实验测试不同干扰源数目 L 情况下, RJIO-DA 算法及 JIO-DA 算法与其它降维和满维算

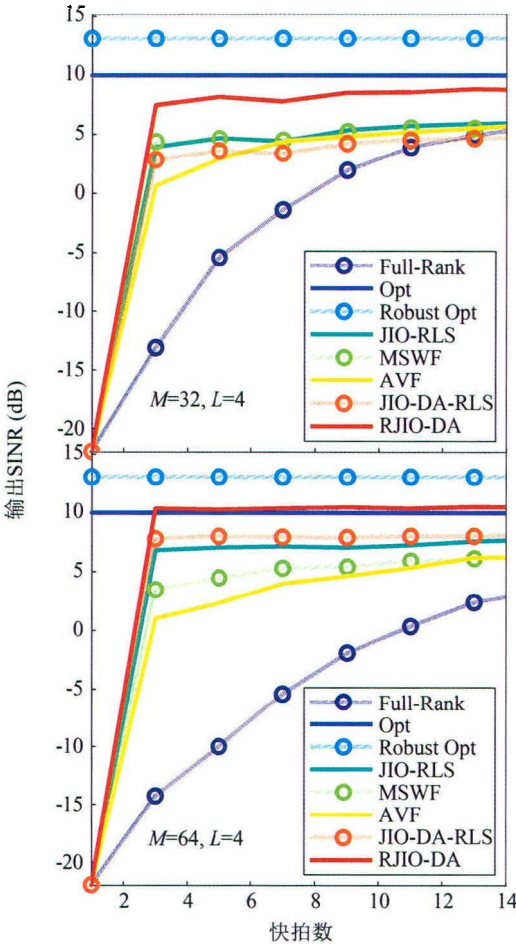


图 6 阵元数为 32, 64 的 RJIO-DA 及现有的降维算法的 SINR 性能比较图

法的波束形成情况。因此，仿真平台采用 $M=64$ 的大
型均匀 ULA 阵列，拥有 7 组来自不同方向的远场窄
带干扰信号，分别来自 $-20^\circ, -30^\circ, -40^\circ, 20^\circ, 30^\circ,$
 $40^\circ, 50^\circ$ 方向，其余实验参数与 4.1 章节一致。

如图 7 所示，当干扰源数目 L 较多，新算法也可
以有效地进行自适应波束形成。此时，满维算法旁瓣
的幅度变得很高，因为干扰增多之后，高纬度应用时
满维算法收敛所需的快拍数量也就越多，在这个实验
环境下 500 个快拍数显然不足以让满维算法收敛。
对其它降维算法来说，虽然在期望方向上有明显集
中，但是对于干扰的来源变得很不敏感，无法明确指出
7 个干扰源的位置。而此时，当 $L=7$, RJIO-DA 算法
很好地辨析出干扰源的来源方向 $-20^\circ, -30^\circ, -40^\circ,$
 $20^\circ, 30^\circ, 40^\circ, 50^\circ$ 。因此，RJIO-DA 算法较好地解决
了因导向向量的误差而造成的失模现象，不仅仅在
期望方向上稳健地聚集波束，降低旁瓣，而且精确地
形成干扰方向的自适应零陷。

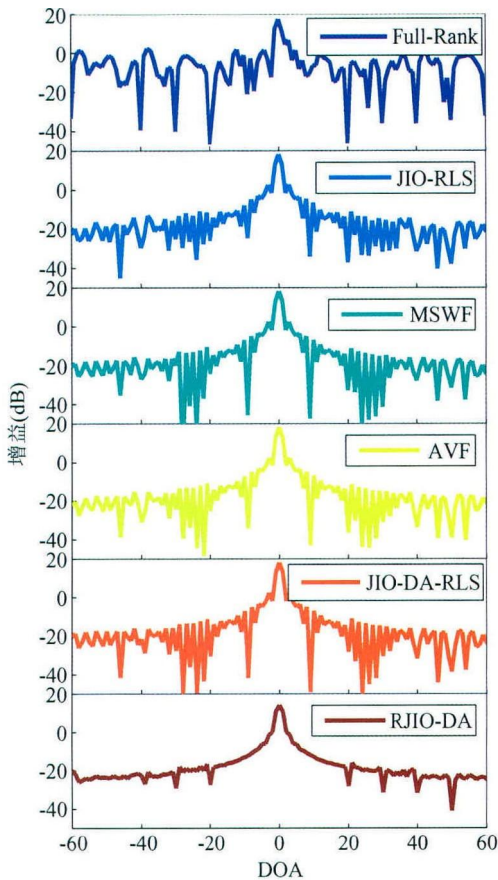


图 7 干扰源数 $L=7$ 下的 RJIO-DA 及现有
的降维算法的波束形成比较图

5 结束语

是否具有鲁棒性和实时性，是自适应波束形成
技术应用于实际的主要瓶颈。比如，采用多个波段进
行空间目标探测的相控阵雷达系统，往往存在信号指

向误差，自适应波束形成技术可能因此将信号误认为
是干扰而将其抑制，为此需要增强算法的鲁棒性；同
时，随着相控阵雷达阵元数的增加，数据处理量也会
激增，波束形成算法的实时性也是不可或缺的。

本文提出的快速稳健自适应波束形成算法
RJIO-DA 算法，通过降维算法增强了波束形成的实
时性，通过引入导向向量误差提升了鲁棒性。仿真结
果表明，RJIO-DA 方案不仅在期望方向上更稳健地
聚集波束、降低旁瓣，而且可以更精确地在干扰方向
形成自适应零陷，对实时性和鲁棒性都有所改进和
优化。论文研究工作可为雷达、通信等领域的阵列信
号处理提供借鉴。

参 考 文 献

1 Wang L. Array signal processing algorithms for beamforming and direction finding. York: University of York, 2009

2 Godara L. The effect of phase-shifter errors on the performance of an antenna-array beamformer. *IEEE Journal of Oceanic Engineering*, 1985; **10**(3): 278—284

3 Pedersen K I, Mogensen P E, Fleury B H. A stochastic model of the temporal and azimuthal dispersion seen at the base station in outdoor propagation environments. *IEEE Trans. Vehicular Technology*, 2000; **49**(2): 437—447

4 Linsheng L, Wei H, Yan Z *et al.* Design and implementation of an active array antenna with remote controllable radiation patterns for mobile communications. *IEEE Trans. Antennas and Propagation* 2014; **62**(2), 913—921

5 Jablon N K. Adaptive beamforming with the generalized sidelobe canceller in the presence of array imperfections. *IEEE Trans. Antennas and Propagation*, 1986; **34**(8): 996—1012

6 Hong Y J, Yeh C C, Ucci D R. The effect of a finite-distance signal source on a far-field steering Applebaum array-two dimensional array case. *IEEE Trans. Antennas and Propagation*, 1988; **36**(4): 468—475

7 Tufts D W, Shah A A. Estimation of a signal waveform from noisy data using low-rank approximation to a data matrix. *IEEE Trans. Signal Processing*, 1993; **41**(4): 1716—1721

8 Subbaram H, Abend K. Interference suppression via orthogonal projections: A performance analysis. *IEEE Trans. Antennas and Propagation*, 1993; **41**(9): 1187—1194

9 Chouvardas S, Slavakis K, Theodoridis S. Trading off complexity with communication costs in distributed adaptive learning via Krylov subspaces for dimensionality reduction. *IEEE Journal of Selected Topics in Signal Processing*, 2013; **7**(2): 257—273

10 Burykh S, Karim A. Reduced-Rank adaptive filtering using krylov subspace. *EURASIP Journal on Applied Signal Processing*, 2002; **2002**(1): 1387—1400

11 de Lamare R C, Diniz P S R. Blind adaptive interference suppression based on set-membership constrained constant-modulus algorithms with dynamic bounds.

- IEEE Trans. Signal Processing*, 2013; **61**(5): 1288—1301
- 12 Mathews B L, Mili L, Zaghoul A I. Auxiliary vector selection algorithms for adaptive beamforming. *2005 IEEE Antennas and Propagation Society International Symposium*, 2005; **3A**: 271—274
 - 13 Nuan S, Alokozai W U, de Lamare R C *et al.* Adaptive widely linear Reduced-Rank beamforming based on joint iterative optimization. *IEEE Signal Processing Lett.*, 2014; **21**(3): 265—269
 - 14 Sheng L, de Lamare R C. Blind reduced-rank adaptive receivers for DS-UWB systems based on joint iterative optimization and the constrained constant modulus criterion. *IEEE Trans. Vehicular Technology*, 2011; **60**(6): 2505—2518
 - 15 周胜增, 杜选民. 快速收敛最小方差无畸变响应算法研究及应用. *声学学报*. 2009, **34**(6): 515—520
 - 16 徐晓男, 马启明, 杜栓平. 波束空间能量约束的稳健自适应波束形成. *声学学报*. 2013, **38**(3): 258—264
 - 17 闫超, 郭良浩, 汪福全等. 一种鲁棒性的最小方差无失真响应波束形成算法及其应用. *声学学报*. 2011; **36**(6): 605—610
 - 18 林鹏, 宫在晓, 郭永刚等. 拖线阵机动时的自适应波束形成. *声学学报*. 2013, **38**(3): 251—257
 - 19 Wang R, Li S, He X *et al.* Novel direction-adaptive based reduced-rank beamforming algorithm. *Proceedings of the Tenth International Symposium on Wireless Communication Systems (ISWCS 2013)*, 2013; 1—4
 - 20 Jwu-Sheng H, Ming-Tang L, Chia-Hsin Y. Robust adaptive beamformer for speech enhancement using the second-order extended H_{∞} filter. *IEEE Trans. Audio, Speech, and Language Processing*, 2013; **21**(1): 39—50
 - 21 Chang L, Yeh C C. Performance of DMI and eigenspace-based beamformers. *IEEE Trans. Antennas and Propagation*, 1992; **40**(11): 1336—1347
 - 22 Gershman A B, Serebryakov G V, Bă Hme J F. Constrained Hung-Turner adaptive beam-forming algorithm with additional robustness to wideband and moving jammers. *IEEE Trans. Antennas and Propagation*, 1996; **44**(3): 361—367
 - 23 Riba J, Goldberg J, Vazquez G. Robust beamforming for interference rejection in mobile communications. *IEEE Trans. Signal Processing*, 1997; **45**(1): 271—275
 - 24 Bell K L, Ephraim Y, Van Trees H L. A Bayesian approach to robust adaptive beamforming. *IEEE Trans. Signal Processing*, 2000; **48**(2): 386—398
 - 25 Landau L, de Lamare R C, Haardt M. Robust adaptive beamforming algorithms using the constrained constant modulus criterion. *IET Signal Processing*, 2014; **8**(5): 447—457
 - 26 武思军. 稳健的自适应波束形成算法研究. 哈尔滨工程大学博士学位论文, 2005