A Low-Complexity Two-Dimensional DOA Estimation Algorithm Based on an L-Shaped Sensor Array

Jingjing Cai, Yuyan Tan, Huanyin Zhang, Wei Liu, Fuwei Tan and Yangyang Dong

***Abstract*—In this paper, a derivative-based MUSIC (multiple signal classification) algorithm for a mixture of circular and noncircular signals (DB-MUSIC-M) is proposed for two-dimensional (2D) direction of arrival (DOA) estimation employing an L-shaped uniform array. The DB-MUSIC-M transforms the 2D DOA estimation problem into a single one-dimensional (1D) estimation by finding the derivative of the objective function of the 2D improved MUSIC (I-MUSIC) algorithm, which greatly reduces its computational complexity. As it utilizes both the pseudo covariance matrix and the covariance matrix of the array data, the maximum number of signals that can be estimated is much higher than the total number of sensors. As a special case, the derivative-based MUSIC (DB-MUSIC) for circular signals is also proposed. There is no need for angle pairing for the proposed algorithms and they can handle the angle ambiguity problem effectively. As shown by simulation results, the proposed DB-MUSIC-M algorithm outperforms existing algorithms, with significantly reduced complexity compared to a direct 2D search method. Moreover, the proposed approach can be applied to some other array structures such as the uniform planar array.**

***Index Terms*—L-shaped array; two-dimensional direction of arrival estimation; derivative-based MUSIC; angle pairing**

I. Introduction

Direction of arrival (DOA) estimation has found a wide range of applications in radar, sonar, and wireless communications. For the two-dimensional (2D) DOA estimation problem, different 2D array structures can be employed, and one choice is the L-shaped array given its simple structure and good performance [1]–[4]. There are mainly two types of 2D DOA estimation algorithms. For the first one, it transforms the 2D DOA estimation problem into two one-dimensional (1D) estimations, and then angle pairing is performed to recover the 2D angles of each source [5]–[12]. For the second type, it estimates the 2D DOAs simultaneously with automatic angle pairing [13]–[20]. One issue associated with the first type is that angle pairing may not work in some cases, so the second type may be preferred in practice.

Among the second type, in [14], an algorithm based on the joint singular value decomposition (SVD) technique (JSVD) is proposed in [13], which performs simultaneous SVD of two cross-correlation matrices. In [14], an algorithm using parallel factor analysis (PARAFAC) is proposed, which effectively uses the second-order statistics of some sensors selected from the L-shaped array to construct a PARAFAC model. In [19], an algorithm combining the propagator method (PM) and the ESPRIT algorithm (PM-ES) is proposed, taking advantage of the conjugate symmetry property of the array manifold matrix to increase the effective array aperture. In [20], the conjugate symmetry property of the array manifold matrix (CS2R) is exploited, to increase the effective array aperture and the number of virtual snapshots simultaneously. In addition, a rectangular array based 2D DOA estimation algorithm is proposed in [21], which provides a way to transform the 2D DOA estimation into a single 1D DOA estimation without angle pairing; the 1D DOA estimation is achieved by the derivation of a Lagrange function with respect to another 1D steering vector, where the constraint of the Lagrange function limits the first element of the steering vector to be 1. However, although it can be extended to the L-shaped array in theory, its performance is not satisfactory.

Recently, many DOA estimation algorithms for a mixture of circular and noncircular signals have been developed as noncircular signals are widely used in practice [22]–[24], such as binary phase shift keying (BPSK), minimum shift keying (MSK), Gaussian MSK (GMSK), pulse amplitude modulation (PAM), and unbalance quadrature PSK (UQPSK) signals, etc. The DOA estimation algorithms for noncircular signals usually consider both the covariance matrix and the pseudo covariance matrix, which further improves the performance of those algorithms based on the covariance matrix only [22], [23]. Overall, the performance of the 2D DOA estimation algorithms can be improved by considering the noncircularity property of the signals.

In this paper, a novel 2D DOA estimation algorithm based on the L-shaped uniform array for a mixture of circular and noncircular signals is proposed. Firstly, a derivative-based MUSIC algorithm for a mixture of circular and noncircular signals (DB-MUSIC-M) is proposed based on the objective function of the 2D extension of the improved MUSIC (I-MUSIC) algorithm [23], [25]. The DB-MUSIC-M algorithm transforms the 2D DOA estimation problem into a single 1D DOA estimation using a derivative based optimization method, and no additional angle pairing is required. By considering the covariance matrix and pseudo covariance matrix of the signals, the number of signals that can be estimated by the

This work was supported by the National Natural Science Foundation of China (No.61805189 and No.61901332), the Fundamental Research Funds for the Central Universities (No.JB210201) and Natural Science Basic Research Program of Shaanxi Provincial of China (No.2018JQ6068).

Jingjing Cai, Yuyan Tan, Huanyin Zhang, Fuwei Tan and Yangyang Dong are with the Department of Electronic and Engineering, Xidian University, China (e-mail: jjcai@mail.xidian.edu.cn, yyantan@163.com, zhyxidian@163.com, fwtan@stu.xidian.edu.cn, dongyangyang2104@126.com).

Wei Liu is with the Department of Electronic and Electrical Engineering, University of Sheffield, UK (e-mail: w.liu@sheffield.ac.uk).

proposed algorithm is much higher than the number of sensors. As a special case, for the situation with circular signals only, the DM-MUSIC algorithm is developed as in our earlier published conference paper [26]. As the Cramer-Rao bound (CRB) provides an important benchmark for assessing the performance of various 2D DOA estimation algorithms [27], [28], the CRB for 2D DOA estimation for a mixture of circular and noncircular signals is derived following the approach in [29], which can handle the general underdetermined problem. As demonstrated by computer simulations, the DB-MUSIC-M and DB-MUSIC algorithms have outperformed some existing corresponding L-shaped array based 2D DOA estimation algorithms. Moreover, although the proposed algorithms are derived based on the L-shaped array, the approach can be applied to some other array structures such as the uniform planar array.

This paper is organized as follows. In Sec. II, the array model with a mixture of circular and noncircular signals is provided, followed by a review of the 2D I-MUSIC algorithm. The proposed algorithms are presented in Sec. III. Simulation results are provided in Sec. IV and conclusions are drawn in Sec. V.

II. The Array Model and the 2D I-MUSIC algorithm

A. The array model

Consider an L-shaped array with an M-element uniform sub-array on the y-axis and an N-element uniform sub-array on the x-axis, and the spacing of the adjacent elements of each sub-array is d, as shown in Fig. 1. Suppose K uncorrelated narrow-band signals are incident on the array with DOAs $(\alpha_k, \beta_k), k = 1, 2, \cdots, K$, where α_k represents the angle between the incident direction of the kth signal and x-axis, while β_k the angle between its incident direction and y-axis, with $\alpha_k, \beta_k \in (0°, 180°]$. Among these K signals, the first K_{nc} signals are assumed to be non-circular and the last $K_{\text{c}} = K - K_{\text{nc}}$ signals are circular. Assume the first K_{snc} signals among the K_{nc} noncircular signals are strictly noncircular. The signal vector can be written as $\mathbf{s}(t) = [s_1(t), \cdots, s_{K_{\text{snc}}}(t), \cdots, s_{K_{\text{nc}}}(t), \cdots, s_K(t)]^{\text{T}}$ with the corresponding source power vector $\boldsymbol{\eta} = [\eta_1, \cdots, \eta_{K_{\text{snc}}}, \cdots, \eta_{K_{\text{nc}}}, \cdots, \eta_K]^{\text{T}}$, where $(\cdot)^{\text{T}}$ denotes the transpose operation. Note that the strictly non-circular signals can be transformed into a real-valued signal by adding an appropriate phase shift ϕ_k, i.e., $e^{j\phi_k} s_k(t)$ will become real-valued.

The two sub-array manifold matrices are given by

$$\begin{aligned}\mathbf{A}_{\text{x}}(\boldsymbol{\alpha}) =& \\ &[\mathbf{a}_{\text{x}}(\alpha_1), \ldots, \mathbf{a}_{\text{x}}(\alpha_{K_{\text{snc}}}), \ldots, \mathbf{a}_{\text{x}}(\alpha_{K_{\text{nc}}}), \ldots, \mathbf{a}_{\text{x}}(\alpha_K)] \\ \mathbf{A}_{\text{y}}(\boldsymbol{\beta}) =& \\ &[\mathbf{a}_{\text{y}}(\beta_1), \ldots, \mathbf{a}_{\text{y}}(\beta_{K_{\text{snc}}}), \ldots, \mathbf{a}_{\text{y}}(\beta_{K_{\text{nc}}}), \ldots, \mathbf{a}_{\text{y}}(\beta_K)]\end{aligned} \tag{1}$$

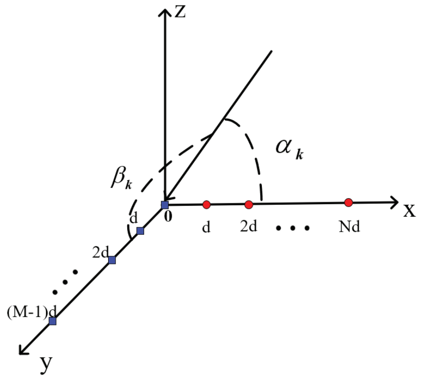

Fig. 1. The array model.

with

$$\begin{aligned}\mathbf{a}_{\text{x}}(\alpha_k) =&[\exp(-j2\pi d\cos\alpha_k/\lambda), \cdots, \\ &\exp(-j2\pi dN\cos\alpha_k/\lambda)]^{\text{T}} \\ \mathbf{a}_{\text{y}}(\beta_k) =&[1, \exp(-j2\pi d\cos\beta_k/\lambda), \cdots, \\ &\exp(-j2\pi d(M-1)\cos\beta_k/\lambda)]^{\text{T}} \\ \boldsymbol{\alpha} =&[\alpha_1, \ldots, \alpha_K]^{\text{T}} \;\; \boldsymbol{\beta} = [\beta_1, \ldots, \beta_K]^{\text{T}}\end{aligned} \tag{2}$$

where λ is the signal wavelength.

Then, the received signal vectors can be represented by

$$\mathbf{z}(t) = \mathbf{A}(\boldsymbol{\alpha}, \boldsymbol{\beta})\mathbf{s}(t) + \mathbf{n}(t) \tag{3}$$

with

$$\begin{aligned}\mathbf{A}(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \begin{bmatrix}\mathbf{A}_{\text{x}}(\boldsymbol{\alpha}) \\ \mathbf{A}_{\text{y}}(\boldsymbol{\beta})\end{bmatrix} = [\mathbf{a}(\alpha_1, \beta_1), \ldots, \mathbf{a}(\alpha_K, \beta_K)] \\ \mathbf{a}(\alpha_k, \beta_k) &= \begin{bmatrix}\mathbf{a}_{\text{x}}(\alpha_k) \\ \mathbf{a}_{\text{y}}(\beta_k)\end{bmatrix}\end{aligned} \tag{4}$$

where $\mathbf{z}(t)$ represents the received signal vector, $\mathbf{n}(t)$ denotes the vector of additive noises which are temporally and spatially white with zero-mean and variance σ^2, and uncorrelated with incident signals.

To make use of the noncircularity information of the signals, we combine the received signal $\mathbf{z}(t)$ and its conjugate as the extended array data vector, given by

$$\begin{aligned}\mathbf{z}_{\text{e}}(t) = \begin{bmatrix}\mathbf{z}(t) \\ \mathbf{z}^*(t)\end{bmatrix} &= \begin{bmatrix}\mathbf{A}(\boldsymbol{\alpha}, \boldsymbol{\beta})\mathbf{s}(t) + \mathbf{n}(t) \\ \mathbf{A}^*(\boldsymbol{\alpha}, \boldsymbol{\beta})\mathbf{s}^*(t) + \mathbf{n}^*(t)\end{bmatrix} \\ &= \mathbf{A}_{\text{e}}(\boldsymbol{\alpha}, \boldsymbol{\beta})\mathbf{s}_{\text{e}}(t) + \mathbf{n}_{\text{e}}(t)\end{aligned} \tag{5}$$

with

$$\begin{aligned}&\mathbf{A}_{\text{e}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = [\mathbf{a}_{\text{snc}}(\alpha_1, \beta_1), \cdots, \mathbf{a}_{\text{snc}}(\alpha_{K_{\text{snc}}}, \beta_{K_{\text{snc}}}), \\ &\mathbf{A}_{\text{nc}}(\alpha_{K_{\text{snc}}+1}, \beta_{K_{\text{snc}}+1}), \cdots, \mathbf{A}_{\text{nc}}(\alpha_{K_{\text{nc}}}, \beta_{K_{\text{nc}}}), \\ &\mathbf{A}_{\text{c}}(\alpha_{K_{\text{nc}}+1}, \beta_{K_{\text{nc}}+1}), \cdots, \mathbf{A}_{\text{c}}(\alpha_K, \beta_K)], \\ &\mathbf{s}_{\text{e}}(t) = \left[e^{j\phi_1}s_1(t), \ldots, e^{j\phi_{K_{\text{snc}}}}s_{K_{\text{snc}}}(t), s_{K_{\text{snc}}+1}(t),\right. \\ &\;\; s^*_{K_{\text{snc}}+1}(t), \ldots, s_{K_{\text{nc}}}(t), s^*_{K_{\text{nc}}}(t), s_{K_{\text{nc}}+1}(t), s^*_{K_{\text{nc}}+1}(t), \\ &\;\; \left.\ldots, s_K(t), s^*_K(t)\right]^{\text{T}}, \\ &\mathbf{n}_{\text{e}}(t) = \begin{bmatrix}\mathbf{n}(t) \\ \mathbf{n}^*(t)\end{bmatrix},\end{aligned} \tag{6}$$

where

$$\begin{aligned}
&\mathbf{a}_{\text{snc}}(\alpha_k,\beta_k)=\left[\begin{array}{c} e^{-j\phi_k}\mathbf{a}(\alpha_k,\beta_k)\\ e^{j\phi_k}\mathbf{a}^*(\alpha_k,\beta_k)\end{array}\right],\\
&\qquad\qquad\qquad\qquad\qquad\qquad k=1,\ldots,K_{\text{snc}}\\
&\mathbf{A}_{\text{nc}}(\alpha_k,\beta_k)=\mathbf{A}_{\text{c}}(\alpha_k,\beta_k)\\
&\qquad=\left[\begin{array}{cc}\mathbf{a}(\alpha_k,\beta_k) & \mathbf{0}_{(N+M)\times 1}\\ \mathbf{0}_{(N+M)\times 1} & \mathbf{a}^*\left(\alpha_k,\beta_k\right)\end{array}\right],\\
&\qquad k=K_{\text{snc}}+1,\ldots,K_{\text{nc}} \text{ and } K_{nc}+1,\ldots,K
\end{aligned} \tag{7}$$

$(\cdot)^*$ is the conjugate operation, and $\mathbf{0}_{h\times l}$ is an $h\times l$ zero matrix.

From the properties of circular and noncircular signals, we have [22]

$$\begin{aligned}
&\mathrm{E}[s_k(t)s_k^*(t)]=\eta_k,\ k=1,\ldots,K\\
&\mathrm{E}[s_k(t)s_k(t)]=\eta_k e^{-j2\phi_k},\ k=1,\ldots,K_{\text{snc}}\\
&\mathrm{E}[s_k(t)s_k(t)]=\eta_k\rho_k e^{j\psi_k}, k=K_{\text{snc}}+1,\ldots,K_{\text{nc}}\\
&\mathrm{E}[s_k(t)s_k(t)]=0,\ k=K_{\text{nc}}+1,\ldots,K
\end{aligned} \tag{8}$$

where $\mathrm{E}[\cdot]$ is the expectation operation, ρ_k and ψ_k are the noncircularity rate and phase of the kth nonstrictly noncircular signal, with $0<\rho_k<1$. For strictly noncircular signals, the noncircularity rate and phase are 1 and $(-2\phi_k)$, respectively.

B. The 2D I-MUSIC algorithm

For a mixture of circular and noncircular signals, a 1D improved MUSIC (I-MUSIC) algorithm has been proposed based on both traditional arrays and the recently developed sum and difference co-array concepts [23], [25]. Here we extend it to the 2D case to provide a basis for our proposed DB-MUSIC-M and DB-MUSIC algorithms.

Firstly, the covariance matrix of the extended output vector $\mathbf{z}_{\text{e}}(t)$ can be written in the following form.

$$\begin{aligned}
\mathbf{R}_{\text{e}}&=\mathrm{E}[\mathbf{z}_{\text{e}}(t)\mathbf{z}_{\text{e}}^{\text{H}}(t)]\\
&=\mathbf{A}_{\text{e}}(\boldsymbol{\alpha},\boldsymbol{\beta})\mathbf{R}_{\text{s}}\mathbf{A}_{\text{e}}^{\text{H}}(\boldsymbol{\alpha},\boldsymbol{\beta})+\sigma^2\mathbf{I}_{2(N+M)}
\end{aligned} \tag{9}$$

with

$$\begin{aligned}
&\mathbf{R}_{\text{s}}=\operatorname{diag}\{\mathbf{R}_{\text{snc}},\mathbf{R}_{\text{nc}},\mathbf{R}_{\text{c}}\}\\
&\mathbf{R}_{\text{snc}}=\operatorname{diag}\{\eta_1,\eta_2,\ldots,\eta_{K_{\text{snc}}}\}\\
&\mathbf{R}_{\text{nc}}=\operatorname{diag}\{\mathbf{B}(K_{\text{snc}}+1),\mathbf{B}(K_{\text{snc}}+2),\ldots,\mathbf{B}(K_{\text{nc}})\}\\
&\mathbf{B}(k)=\eta_k\left[\begin{array}{cc}1 & \rho_k e^{j\psi_k}\\ \rho_k e^{-j\psi_k} & 1\end{array}\right]\\
&\mathbf{R}_{\text{c}}=\operatorname{diag}\{\eta_{K_{\text{nc}}+1},\eta_{K_{\text{nc}}+1},\ldots,\eta_K,\eta_K\}
\end{aligned} \tag{10}$$

where $(\cdot)^{\text{H}}$ denotes the conjugate transpose operation, $\mathbf{I}_h$ is an $h\times h$ identity matrix, and $\operatorname{diag}\{\cdot\}$ is the diagonal operator. The dimensions of $\mathbf{R}_{\text{snc}}$, $\mathbf{R}_{\text{nc}}$ and $\mathbf{R}_{\text{c}}$ are $K_{\text{snc}}\times K_{\text{snc}}$, $2(K_{\text{nc}}-K_{\text{snc}})\times 2(K_{\text{nc}}-K_{\text{snc}})$, and $2(K-K_{\text{nc}})\times 2(K-K_{\text{nc}})$, separately. As a result, the dimension of $\mathbf{R}_{\text{s}}$ is $(2K-K_{\text{snc}})\times(2K-K_{\text{snc}})$.

Since $\mathbf{R}_{\text{s}}$ is a full rank matrix, the MUSIC-type algorithm can be applied here [30], with $\mathbf{A}_{\text{e}}(\boldsymbol{\alpha},\boldsymbol{\beta})$ considered as the steering matrix. Applying eigenvector decomposition on $\mathbf{R}_{\text{e}}$, we can then obtain the signal subspace and noise subspace $\mathbf{U}_{\text{es}}$ and $\mathbf{U}_{\text{en}}$, respectively, where the dimension of $\mathbf{U}_{\text{es}}$ is $2(N+M)\times(2K-K_{\text{snc}})$, while that of $\mathbf{U}_{\text{en}}$ is $2(N+M)\times[2(N+M)-(2K-K_{\text{snc}})]$. It can be deduced that the maximum number of signals to be resolved is $2K-K_{\text{snc}}<2(N+M)$.

Since the noise subspace $\mathbf{U}_{\text{en}}$ is orthogonal to the steering matrix $\mathbf{A}_{\text{e}}(\boldsymbol{\alpha},\boldsymbol{\beta})$, the I-MUSIC algorithm can be obtained as follows [23].

For an arbitrary set of (α,β), the steering matrix $\bar{\mathbf{A}}_{\text{e}}(\alpha,\beta)$ corresponding to $\mathbf{A}_{\text{e}}(\boldsymbol{\alpha},\boldsymbol{\beta})$ is defined as

$$\bar{\mathbf{A}}_{\text{e}}(\alpha,\beta)=\left[\begin{array}{cc}\mathbf{a}(\alpha,\beta) & \mathbf{0}_{(N+M)\times 1}\\ \mathbf{0}_{(N+M)\times 1} & \mathbf{a}^*(\alpha,\beta)\end{array}\right] \tag{11}$$

Then, the objective function to be minimized for the DB-MUSIC-M algrotithm is defined as follows [23].

$$\mathrm{L}_{\text{e}}(\alpha,\beta)=\det[\mathbf{V}_{\text{e}}(\alpha,\beta)] \tag{12}$$

with

$$\mathbf{V}_{\text{e}}(\alpha,\beta)=\bar{\mathbf{A}}_{\text{e}}^{\text{H}}(\alpha,\beta)\mathbf{U}_{\text{en}}\mathbf{U}_{\text{en}}^{\text{H}}\bar{\mathbf{A}}_{\text{e}}(\alpha,\beta) \tag{13}$$

where $\det[\cdot]$ is the determinant operator.

The 2D I-MUSIC algorithm performs spatial spectrum searching using the following function

$$\mathrm{P}_{\text{e}}(\alpha,\beta)=\frac{1}{\mathrm{L}_{\text{e}}(\alpha,\beta)} \tag{14}$$

The DOA estimation results can be obtained by 2D searching for the K peaks of the spatial spectrum $\mathrm{P}_{\text{e}}(\alpha,\beta)$.

III. The Proposed algorithms

A. The Proposed DB-MUSIC-M algorithm

The DB-MUSIC-M algorithm transforms the 2D angle search of the 2D I-MUSIC into 1D search. Firstly, we transform the steering vector into the product of two parts, corresponding to α and β, separately, given below

$$\mathbf{a}(\alpha_k,\beta_k)=\mathbf{A}_\alpha(\alpha_k)\mathbf{a}_\beta(\beta_k) \tag{15}$$

with

$$\begin{aligned}
&\mathbf{A}_\alpha(\alpha_k)=\left[\begin{array}{cc}\operatorname{diag}\{\mathbf{a}_{\text{x}}(\alpha_k)\} & \mathbf{0}_{N\times M}\\ \mathbf{0}_{M\times N} & \mathbf{I}_M\end{array}\right]\\
&\mathbf{a}_\beta(\beta_k)=\left[\begin{array}{c}\mathbf{1}_{N\times 1}\\ \mathbf{a}_{\text{y}}(\beta_k)\end{array}\right]
\end{aligned} \tag{16}$$

where $\mathbf{1}_{N\times 1}$ is an all-one $N\times 1$ vector, $\mathbf{A}_\alpha(\alpha_k)$ is an $(N+M)\times(N+M)$ matrix, and $\mathbf{a}_\beta(\beta_k)$ an $(N+M)\times 1$ column vector. Then, $\bar{\mathbf{A}}_{\text{e}}(\alpha_k,\beta_k)$ is decomposed into the product of two matrices, which are related to α_k and β_k, separately.

$$\begin{aligned}
\bar{\mathbf{A}}_{\text{e}}(\alpha_k,\beta_k)&=\left[\begin{array}{cc}\mathbf{A}_\alpha(\alpha_k)\mathbf{a}_\beta(\beta_k) & \mathbf{0}_{(N+M)\times 1}\\ \mathbf{0}_{(N+M)\times 1} & \mathbf{A}_\alpha^*(\alpha_k)\mathbf{a}_\beta^*(\beta_k)\end{array}\right]\\
&=\left[\begin{array}{cc}\mathbf{A}_\alpha(\alpha_k) & \mathbf{0}_{(N+M)\times(N+M)}\\ \mathbf{0}_{(N+M)\times(N+M)} & \mathbf{A}_\alpha^*(\alpha)\end{array}\right]\\
&\quad\left[\begin{array}{cc}\mathbf{a}_\beta(\beta_k) & \mathbf{0}_{(N+M)\times 1}\\ \mathbf{0}_{(N+M)\times 1} & \mathbf{a}_\beta^*(\beta_k)\end{array}\right]
\end{aligned} \tag{17}$$

Then, $\mathbf{V}_{\text{e}}(\alpha_k,\beta_k)$ can be written as

$$\begin{aligned}
\mathbf{V}_{\text{e}}(\alpha_k,\beta_k)=&\left[\begin{array}{cc}\mathbf{a}_\beta(\beta_k) & \mathbf{0}_{(N+M)\times 1}\\ \mathbf{0}_{(N+M)\times 1} & \mathbf{a}_\beta^*(\beta_k)\end{array}\right]^{\text{H}}\mathbf{G}_{\text{e}}(\alpha_k)\\
&\left[\begin{array}{cc}\mathbf{a}_\beta(\beta_k) & \mathbf{0}_{(N+M)\times 1}\\ \mathbf{0}_{(N+M)\times 1} & \mathbf{a}_\beta^*(\beta_k)\end{array}\right]
\end{aligned} \tag{18}$$

with

$$\mathbf{G}_{\mathrm{e}}(\alpha_k) = \begin{bmatrix} \mathbf{A}_\alpha(\alpha_k) & \mathbf{0}_{(N+M)\times(N+M)} \\ \mathbf{0}_{(N+M)\times(N+M)} & \mathbf{A}_\alpha^*(\alpha_k) \end{bmatrix}^{\mathrm{H}} \mathbf{U}_{\mathrm{en}} \\ \mathbf{U}_{\mathrm{en}}^{\mathrm{H}} \begin{bmatrix} \mathbf{A}_\alpha(\alpha_k) & \mathbf{0}_{(N+M)\times(N+M)} \\ \mathbf{0}_{(N+M)\times(N+M)} & \mathbf{A}_\alpha^*(\alpha_k) \end{bmatrix} \tag{19}$$

Note here that the dimension of $\mathbf{G}_{\mathrm{e}}$ is $2(N+M)\times 2(N+M)$, with a rank of $2(N+M)-(2K-K_{\mathrm{snc}})$, so it is not a full-rank matrix. The idea of the DB-MUSIC-M algorithm is to find β directly by setting the derivative of the function $\mathrm{L}_{\mathrm{e}}(\alpha,\beta)$ with respect to β to zero.

Now, assume α is a fixed value $\hat{\alpha}_{\hat{k}}$ in the set $\{\hat{\alpha}_1,\ldots,\hat{\alpha}_{\hat{k}},\ldots,\hat{\alpha}_{\hat{K}}\}$, $\hat{K} >> K$, so all the elements in $\mathbf{G}_{\mathrm{e}}(\hat{\alpha}_{\hat{k}})$ are fixed. Normally, the set of α is chosen with a uniform interval within the range of interest. Then, the problem of minimizing $\mathrm{L}_{\mathrm{e}}(\hat{\alpha}_{\hat{k}},\beta)$ can be solved by setting the derivative of $\mathrm{L}_{\mathrm{e}}(\hat{\alpha}_{\hat{k}},\beta)$ with respect to β to zero, i.e.,

$$\frac{\partial}{\partial\beta}\det[\mathbf{V}_{\mathrm{e}}\left(\hat{\alpha}_{\hat{k}},\beta\right)] = 0. \tag{20}$$

The variable β is related to the vector $\mathbf{a}_{\mathrm{y}}(\beta)$, which has the following form

$$\mathbf{a}_{\mathrm{y}}(\beta) = \mathbf{a}_{\mathrm{y}}(b) = [1, b, \ldots, b^{M-1}]^{\mathrm{T}} \tag{21}$$

with

$$b = \exp(-j2\pi d cos\beta/\lambda) \tag{22}$$

As a result, the derivative with respect to β can be obtained by

$$\begin{aligned} &\frac{\partial}{\partial\beta}\det[\mathbf{V}_{\mathrm{e}}\left(\hat{\alpha}_{\hat{k}},\beta\right)] = \frac{\partial}{\partial b}\det[\mathbf{V}_{\mathrm{e}}\left(\hat{\alpha}_{\hat{k}},b\right)]\cdot\frac{\partial b}{\partial\beta} = 0 \\ &\Rightarrow \frac{\partial}{\partial b}\det[\mathbf{V}_{\mathrm{e}}\left(\hat{\alpha}_{\hat{k}},b\right)] = 0 \end{aligned} \tag{23}$$

Then, $\mathbf{V}_{\mathrm{e}}\left(\hat{\alpha}_{\hat{k}},b\right)$ can be simplified as

$$\mathbf{V}_{\mathrm{e}}\left(\hat{\alpha}_{\hat{k}},b\right) = \begin{bmatrix} \mathrm{V}_{11}\left(\hat{\alpha}_{\hat{k}},b\right) & \mathrm{V}_{12}\left(\hat{\alpha}_{\hat{k}},b\right) \\ \mathrm{V}_{21}\left(\hat{\alpha}_{\hat{k}},b\right) & \mathrm{V}_{22}\left(\hat{\alpha}_{\hat{k}},b\right) \end{bmatrix} \tag{24}$$

with

$$\begin{aligned} \mathrm{V}_{11}(\hat{\alpha}_{\hat{k}},b) &= \mathbf{a}_\beta^{\mathrm{H}}(b)\mathbf{Q}(\hat{\alpha}_{\hat{k}})\mathbf{a}_\beta(b) \\ \mathrm{V}_{12}(\hat{\alpha}_{\hat{k}},b) &= \mathbf{a}_\beta^{\mathrm{H}}(b)\mathbf{W}(\hat{\alpha}_{\hat{k}})\mathbf{a}_\beta^*(b) \\ \mathrm{V}_{21}(\hat{\alpha}_{\hat{k}},b) &= \mathbf{a}_\beta^{\mathrm{T}}(b)\mathbf{W}^{\mathrm{T}}(\hat{\alpha}_{\hat{k}})\mathbf{a}_\beta(b) \\ \mathrm{V}_{22}(\hat{\alpha}_{\hat{k}},b) &= \mathbf{a}_\beta^{\mathrm{T}}(b)\mathbf{Q}^{\mathrm{T}}(\hat{\alpha}_{\hat{k}})\mathbf{a}_\beta^*(b) \\ \mathbf{a}_\beta(b) &= \begin{bmatrix} \mathbf{1}_{N\times 1} \\ \mathbf{a}_{\mathrm{y}}(b) \end{bmatrix} \end{aligned} \tag{25}$$

Using $q_{h,l}$ and $w_{h,l}$ to represent the elements of $\mathbf{Q}(\hat{\alpha}_{\hat{k}})$ and $\mathbf{W}(\hat{\alpha}_{\hat{k}})$ at row h and column l, separately, the elements $\mathrm{V}_{11}(\hat{\alpha}_{\hat{k}},b)$, $\mathrm{V}_{12}(\hat{\alpha}_{\hat{k}},b)$, $\mathrm{V}_{21}(\hat{\alpha}_{\hat{k}},b)$ and $\mathrm{V}_{22}(\hat{\alpha}_{\hat{k}},b)$ can be further written as

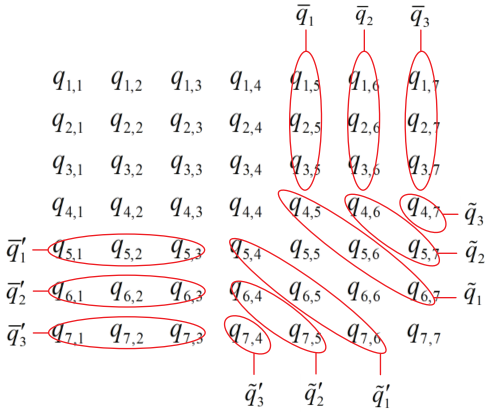
$\bar{q}_1$ $\bar{q}_2$ $\bar{q}_3$
$q_{1,1}$ $q_{1,2}$ $q_{1,3}$ $q_{1,4}$ $q_{1,5}$ $q_{1,6}$ $q_{1,7}$
$q_{2,1}$ $q_{2,2}$ $q_{2,3}$ $q_{2,4}$ $q_{2,5}$ $q_{2,6}$ $q_{2,7}$
$q_{3,1}$ $q_{3,2}$ $q_{3,3}$ $q_{3,4}$ $q_{3,5}$ $q_{3,6}$ $q_{3,7}$
$q_{4,1}$ $q_{4,2}$ $q_{4,3}$ $q_{4,4}$ $q_{4,5}$ $q_{4,6}$ $q_{4,7}$ $\tilde{q}_3$
$\bar{q}'_1$ $q_{5,1}$ $q_{5,2}$ $q_{5,3}$ $q_{5,4}$ $q_{5,5}$ $q_{5,6}$ $q_{5,7}$ $\tilde{q}_2$
$\bar{q}'_2$ $q_{6,1}$ $q_{6,2}$ $q_{6,3}$ $q_{6,4}$ $q_{6,5}$ $q_{6,6}$ $q_{6,7}$ $\tilde{q}_1$
$\bar{q}'_3$ $q_{7,1}$ $q_{7,2}$ $q_{7,3}$ $q_{7,4}$ $q_{7,5}$ $q_{7,6}$ $q_{7,7}$
$\tilde{q}'_3$ $\tilde{q}'_2$ $\tilde{q}'_1$

Fig. 2. An example for the matrix Q with $N = 3$, and $M = 4$.

$$\begin{aligned} \mathrm{V}_{11}(\hat{\alpha}_{\hat{k}},b) = &\sum_{l=1}^{M-1}(\bar{q}_l+\tilde{q}_l)b^l + \left[\sum_{h=1}^{N+1}\sum_{l=1}^{N+1} q_{h,l} + \right. \\ &\left. \sum_{h=l=N+2}^{N+M} q_{h,l}\right] + \sum_{h=1}^{M-1}(\bar{q}'_h+\tilde{q}'_h)b^{-h} \\ \mathrm{V}_{12}(\hat{\alpha}_{\hat{k}},b) = &\sum_{h=l=1}^{M-1}(\bar{w}_h+\bar{w}'_h+\tilde{w}_l)b^{-l} + \\ &\sum_{l=M}^{2M-2}\tilde{w}'_l b^{-l} + \sum_{h=1}^{N+1}\sum_{l=1}^{N+1} w_{h,l} \\ \mathrm{V}_{21}(\hat{\alpha}_{\hat{k}},b) = &\mathrm{V}_{12}^*\left(\hat{\alpha}_{\hat{k}},b\right),\ \mathrm{V}_{22}\left(\hat{\alpha}_{\hat{k}},b\right) = \mathrm{V}_{11}^*\left(\hat{\alpha}_{\hat{k}},b\right) \end{aligned} \tag{26}$$

with

$$\begin{aligned} &\bar{q}_l = \sum_{h=1}^{N} q_{h,(N+1+l)},\ \tilde{q}_l = \sum_{h=N+1}^{N+M-l} q_{h,(h+l)}, \\ &\bar{q}'_h = \sum_{l=1}^{N} q_{(N+1+h),l},\ \tilde{q}'_h = \sum_{l=N+1}^{N+M-h} q_{(h+l),l}, \\ &\bar{w}_h = \sum_{l=1}^{N} w_{l,(N+1+h)},\ \bar{w}'_h = \sum_{l=1}^{N} w_{(N+1+h),l}, \\ &\tilde{w}_l = \sum_{h=N+2}^{N+1+l} w_{h,(2N+2+l-h)},\ \tilde{w}'_l = \sum_{h=N+2}^{N+M} w_{h,(2N+M+2-h)}. \end{aligned} \tag{27}$$

An example is given in Figs. 2 and 3 with $N = 3$, and $M = 4$, which shows calculation of the coefficients in the four elements of $\mathbf{V}_{\mathrm{e}}(\hat{\alpha}_{\hat{k}},b)$ using elements of the matrices $\mathbf{Q}(\hat{\alpha}_{\hat{k}})$ and $\mathbf{W}(\hat{\alpha}_{\hat{k}})$.

Finally, we have

$$\begin{aligned} \frac{\partial}{\partial b}\det[\mathbf{V}_{\mathrm{e}}\left(\hat{\alpha}_{\hat{k}},b\right)] = &\frac{\partial}{\partial b}\left[\mathrm{V}_{11}(\hat{\alpha}_{\hat{k}},b)\mathrm{V}_{22}(\hat{\alpha}_{\hat{k}},b)\right. \\ &\left. -\mathrm{V}_{12}(\hat{\alpha}_{\hat{k}},b)\mathrm{V}_{21}(\hat{\alpha}_{\hat{k}},b)\right] = 0 \end{aligned} \tag{28}$$

There are $(4M-4)$ roots for the polynomial, and only one of them is the desired solution. Firstly, choose the roots $\hat{b}_{\hat{k}}$ on the unit circle, and there may be more than one of them. Then,

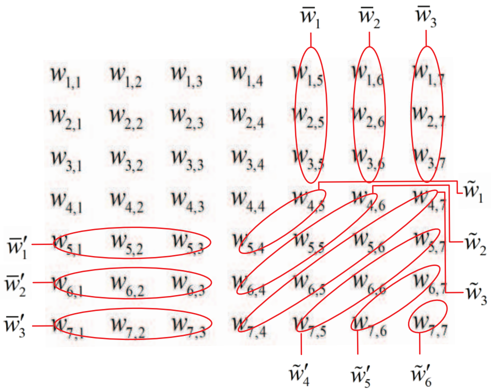
$\bar{w}_1$
$\bar{w}_2$
$\bar{w}_3$
$w_{1,1}$ $w_{1,2}$ $w_{1,3}$ $w_{1,4}$ $w_{1,5}$ $w_{1,6}$ $w_{1,7}$
$w_{2,1}$ $w_{2,2}$ $w_{2,3}$ $w_{2,4}$ $w_{2,5}$ $w_{2,6}$ $w_{2,7}$
$w_{3,1}$ $w_{3,2}$ $w_{3,3}$ $w_{3,4}$ $w_{3,5}$ $w_{3,6}$ $w_{3,7}$
$w_{4,1}$ $w_{4,2}$ $w_{4,3}$ $w_{4,4}$ $w_{4,5}$ $w_{4,6}$ $w_{4,7}$
$\tilde{w}_1$
$\bar{w}'_1$ $w_{5,1}$ $w_{5,2}$ $w_{5,3}$ $w_{5,4}$ $w_{5,5}$ $w_{5,6}$ $w_{5,7}$
$\tilde{w}_2$
$\bar{w}'_2$ $w_{6,1}$ $w_{6,2}$ $w_{6,3}$ $w_{6,4}$ $w_{6,5}$ $w_{6,6}$ $w_{6,7}$
$\tilde{w}_3$
$\bar{w}'_3$ $w_{7,1}$ $w_{7,2}$ $w_{7,3}$ $w_{7,4}$ $w_{7,5}$ $w_{7,6}$ $w_{7,7}$
$\tilde{w}'_4$
$\tilde{w}'_5$
$\tilde{w}'_6$

Fig. 3. An example for the matrix W with $N = 3$, and $M = 4$.

the one giving the minimum value for $\det[\mathbf{V}_\text{e}(\hat{\alpha}_{\hat{k}}, \hat{b}_{\hat{k}})]$ is the desired one. After searching through $\hat{\alpha}_{\hat{k}}$, the K estimated pairs of DOAs $(\bar{\hat{\alpha}}_k, \bar{\hat{b}}_k)$ can be obtained. The estimated angles $\bar{\hat{\beta}}_k$ are then deduced through $\bar{\hat{b}}_k$ by

$$\bar{\hat{\beta}}_k = \arccos\left(\frac{\lambda}{2\pi d}\text{Angle}\left(\bar{\hat{b}}_k\right)\right) \tag{29}$$

where Angle$(\cdot)$ denotes the angle of its argument.

As the proposed algorithm estimates $\bar{\hat{\alpha}}_k$ and $\bar{\hat{\beta}}_k$ simultaneously, no additional pairing operation is needed.

The steps of the DB-MUSIC-M algorithm can be summarized as follows:

Step 1: Select a set of angles $\{\hat{\alpha}_1, \ldots, \hat{\alpha}_{\hat{k}}, \ldots, \hat{\alpha}_{\hat{K}}\}$ with a uniform interval within the range of interest for α.

Step 2: Substitute each $\hat{\alpha}_{\hat{k}}$ into $\mathbf{G}_\text{e}(\hat{\alpha}_{\hat{k}})$, and use its elements to construct the polynomial equation $\frac{\partial}{\partial b}\det[\mathbf{V}_\text{e}\left(\hat{\alpha}_{\hat{k}}, b\right)] = 0$. The desired root $\hat{b}_{\hat{k}}$ is then obtained.

Step 3: After obtaining all pairs of angles $(\hat{\alpha}_{\hat{k}}, \hat{b}_{\hat{k}})$, substitute them into $\text{P}_\text{e}(\hat{\alpha}_{\hat{k}}, \hat{\beta}_{\hat{k}})$ to draw the 1D amplitude spectrum. The K largest peaks then represent the estimated results $(\bar{\hat{\alpha}}_k, \bar{\hat{\beta}}_k)$.

In practice, there may be two or more incident signals with the same α and different β or vice versa, which is called the angle ambiguity problem. As the order of calculating α and β can be changed in the proposed DB-MUSIC-M algorithm, the ambiguity problem can be solved by choosing to calculate α or β firstly, i.e., if ambiguity in α occurs, β can be calculated firstly, and vice versa. The ambiguity angle can be detected before angle estimation by estimating the number of signals based on three different covariance matrices, which are for the x-axis elements, y-axis elements and combined x-axis and y-axis elements, respectively. The number of signals estimated using the x-axis and y-axis covariance matrices implies the number of different angles of α and β, separately, while the number of signals estimated using the combined x-axis and y-axis covariance matrix implies the total number of signals. If the number of different angles of α or β does not coincide with the total number of signals, it indicates that angle ambiguity occurs in either α or β.

The DB-MUSIC-M algorithm uses the same objective function as the 2D I-MUSIC algorithm, so the maximum number of signals to be estimated by DB-MUSIC-M is the same as that of 2D I-MUSIC, which is $2K - K_\text{snc} < 2(N + M)$.

B. The DB-MUSIC algorithm

Now we consider a special case, i.e., only circular signals are present and the number of noncircular signals is zero, or we simply ignore the noncircularity property of the signals and treat them as circular.

In this case, we only consider the original covariance matrix of $\mathbf{z}(t)$, which can be further written as

$$\mathbf{R} = \text{E}[\mathbf{z}(t)\mathbf{z}^\text{H}(t)] = \mathbf{A}(\boldsymbol{\alpha}, \boldsymbol{\beta})\mathbf{R}_\text{sd}\mathbf{A}^\text{H}(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \sigma^2\mathbf{I}_{N+M} \tag{30}$$

with

$$\mathbf{R}_\text{sd} = \text{diag}\{\eta_1, \ldots, \eta_\text{snc}, \ldots, \eta_\text{nc}, \ldots, \eta_K\} \tag{31}$$

Then, the eigen-decomposition of $\mathbf{R}$ gives

$$\mathbf{R} = \mathbf{U}_\text{s}\mathbf{\Lambda}_\text{s}\mathbf{U}_\text{s}^\text{H} + \mathbf{U}_\text{n}\mathbf{\Lambda}_\text{n}\mathbf{U}_\text{n}^\text{H} \tag{32}$$

where $\mathbf{\Lambda}_\text{s}$ and $\mathbf{\Lambda}_\text{n}$ are eigenvalue matrices, corresponding to K largest eigenvalues and the remaining $(N + M - K)$ smallest ones, respectively. $\mathbf{U}_\text{s}$ and $\mathbf{U}_\text{n}$ are constructed by eigenvectors corresponding to $\mathbf{\Lambda}_\text{s}$ and $\mathbf{\Lambda}_\text{n}$, spanning the signal and noise subspaces, separately. It can be seen that it can resolve a maximum number of $K < N + M$ signals, which is also the number of resolvable signals of the following DB-MUSIC algorithm.

For an arbitrary (α, β), the objective function to be minimized for the DB-MUSIC algorithm can be written as

$$\text{V}(\alpha, \beta) = \mathbf{a}_\beta^\text{H}(\beta)\mathbf{G}(\alpha)\mathbf{a}_\beta(\beta) \tag{33}$$

with

$$\mathbf{G}\left(\alpha\right) = \mathbf{A}_\alpha^\text{H}(\alpha)\mathbf{U}_\text{n}\mathbf{U}_\text{n}^\text{H}\mathbf{A}_\alpha(\alpha) \tag{34}$$

Then, the spatial spectrum function is

$$\text{P}(\alpha, \beta) = \frac{1}{\text{V}(\alpha, \beta)} \tag{35}$$

As in the DB-MUSIC-M algorithm, consider $\hat{\alpha}_{\hat{k}}$ as a constant chosen from the set $\{\hat{\alpha}_1, \ldots, \hat{\alpha}_{\hat{k}}, \ldots \hat{\alpha}_{\hat{K}}\}$, $\hat{K} >> K$, and substitute β into b. Then, $\text{V}(\hat{\alpha}_{\hat{k}}, b)$ changes to

$$\begin{aligned}\text{V}(\hat{\alpha}_{\hat{k}}, b) = &\sum_{l=1}^{M-1}(\bar{g}_l + \tilde{g}_l)b^l + \left[\sum_{h=1}^{N+1}\sum_{l=1}^{N+1}g_{h,l} + \sum_{h=l=N+2}^{N+M}g_{h,l}\right] \\ &+ \sum_{h=1}^{M-1}(\bar{g}'_h + \tilde{g}'_h)b^{-h}\end{aligned} \tag{36}$$

with

$$\begin{aligned}\bar{g}_l &= \sum_{h=1}^{N}g_{h,(N+1+l)},\ \tilde{g}_l = \sum_{h=N+1}^{N+M-l}g_{h,(h+l)}, \\ \bar{g}'_h &= \sum_{l=1}^{N}g_{(N+1+h),l},\ \tilde{g}'_h = \sum_{l=N+1}^{N+M-h}g_{(h+l),l}.\end{aligned} \tag{37}$$

where $g_{h,l}$ denotes the element of $\mathbf{G}\left(\alpha\right)$ at row h and column l.

Now, the derivative of $\mathrm{V}(\hat{\alpha}_{\hat{k}}, b)$ with respect to b becomes

$$\begin{aligned}&\frac{\partial}{\partial b}\mathrm{V}\left(\hat{\alpha}_{\hat{k}}, b\right)\\ &=\sum_{l=1}^{M-1} l(\bar{g}_{N+1+l}+\tilde{g}_{l})b^{l-1}-\sum_{h=1}^{M-1} h(\bar{g}'_{N+1+h}+\tilde{g}'_{h})b^{-h-1}\end{aligned} \tag{38}$$

Theoretically, $(2M-2)$ roots can be obtained from the above equation, but only the one $\hat{b}_{\hat{k}}$ on the unit circle and also giving the maximum value for $\mathrm{V}(\hat{\alpha}_{\hat{k}}, \hat{b}_{\hat{k}})$ is the desired one. Suppose $\hat{\beta}_{\hat{k}}$ corresponds to $\hat{b}_{\hat{k}}$. For the whole search area of $\hat{\alpha}_{\hat{k}}$, the K largest peaks represent the estimated results $(\bar{\alpha}_k, \bar{\beta}_k)$.

The DB-MUSIC algorithm is summarized as follows:

Step 1: Select a set of angles $\{\hat{\alpha}_1, \ldots, \hat{\alpha}_{\hat{k}}, \ldots \hat{\alpha}_{\hat{K}}\}$ with a uniform interval within the angle range of interest for α.

Step 2: Substitute each $\hat{\alpha}_{\hat{k}}$ into $\mathbf{G}(\hat{\alpha}_{\hat{k}})$, and use its elements to construct the polynomial equation $\frac{\partial}{\partial b}\mathrm{V}\left(\hat{\alpha}_{\hat{k}}, b\right)=0$, and the desired root $\hat{b}_{\hat{k}}$ is obtained.

Step 3: After obtaining all pairs of angles $(\hat{\alpha}_{\hat{k}}, \hat{b}_{\hat{k}})$, substitute them into $\mathrm{P}(\hat{\alpha}_{\hat{k}}, \hat{\beta}_{\hat{k}})$ to draw the 1D amplitude spectrum. The K largest peaks provide the estimated results $(\bar{\hat{\alpha}}_k, \bar{\hat{\beta}}_k)$.

C. Possible algorithm with $\det[\mathbf{V}_e(\alpha, \beta)] = 0$

In this paper, the minimization of $\det[\mathbf{V}_{\mathrm{e}}(\alpha, \beta)]$ is used as the objective function. One possible alternative is to use the function $\det[\mathbf{V}_{\mathrm{e}}(\alpha, \beta)] = 0$ instead. However, this equation is only valid in the ideal situation, i.e., when α and β represent exact directions of the source signals; when α does not represent true directions of signals, this equation could give values forming false peaks in the final spatial spectrum, leading to very large estimation errors.

To use $\det[\mathbf{V}_{\mathrm{e}}(\alpha, \beta)] = 0$ instead of minimizing $\det[\mathbf{V}_{\mathrm{e}}(\alpha, \beta)]$ in the DB-MUSIC-M, the resultant algorithm can be summarized as follows:

Step 1: Select a set of angles $\{\hat{\alpha}_1, \ldots, \hat{\alpha}_{\hat{k}}, \ldots, \hat{\alpha}_{\hat{K}}\}$ with a uniform interval within the angle range of interest for α.

Step 2: Substitute each $\hat{\alpha}_{\hat{k}}$ into $\mathbf{G}_{\mathrm{e}}(\hat{\alpha}_{\hat{k}})$, and use its elements to construct the polynomial equation $\det[\mathbf{V}_{\mathrm{e}}(\hat{\alpha}_{\hat{k}}, b)] = 0$. The root $\hat{b}_{\hat{k}}$ closest to the unit circle is obtained as the desired one.

Step 3: After obtaining all pairs of angles $(\hat{\alpha}_{\hat{k}}, \hat{b}_{\hat{k}})$, substitute them into $\mathrm{P}_{\mathrm{e}}(\hat{\alpha}_{\hat{k}}, \hat{\beta}_{\hat{k}})$ to draw the 1D amplitude spectrum. The K largest peaks give the estimated results $(\bar{\hat{\alpha}}_k, \bar{\hat{\beta}}_k)$.

As shown later in our simulations, this algorithm based on the function $\det[\mathbf{V}_{\mathrm{e}}(\alpha, \beta)] = 0$ is not an effective way to solve the problem.

D. Computational Complexity Analysis

The computational complexity of the DB-MUSIC-M, DB-MUSIC and 2D I-MUSIC algorithms are shown in Table I, where S_{np} represents the number of snapshots.

Although the 2D I-MUSIC algorithm does not need to consider the computation of $\mathbf{G}_{\mathrm{e}}(\alpha)$ and those associated with the polynomial solver, the computation complexity of the spectrum searching part is much greater than that of DB-MUSIC-M with $\hat{K}^2$ multiplications. In the same way, the computational complexity of DB-MUSIC is also much lower than that of 2D I-MUSIC, and it is a little lower than that of DB-MUSIC-M.

TABLE I
COMPUTATIONAL COMPLEXITY OF DB-MUSIC-M, DB-MUSIC AND 2D I-MUSIC ALGORITHMS.

	DB-MUSIC-M	DB-MUSIC	2D I-MUSIC
Covariance Matrix Computation	$4S_{\mathrm{np}}(M+N)^2$	$S_{\mathrm{np}}(M+N)^2$	$4S_{\mathrm{np}}(M+N)^2$
Eigenvalue Decomposition	$8(M+N)^3$	$(M+N)^3$	$8(M+N)^3$
$\mathbf{G}_{\mathrm{e}}(\alpha)$/$\mathbf{G}(\alpha)$ Computation	$\{16(M+N)^2$ $[2(M+N)-$ $(2K-K_{\mathrm{snc}})]\}\hat{K}$	$[(M+N)^2(M$ $+N-K)]\hat{K}$	$--$
Polynomial Solver	$\{[4(M-1)]^3$ $+4(M-1)\}\hat{K}$	$\{[2(M-1)^3$ $+2(M-1)\}\hat{K}$	$--$
Spectrum Searching	$\{16(M+N)^2$ $[2(M+N)-$ $(2K-K_{\mathrm{snc}})]\}\hat{K}$	$[(M+N)^2$ $(M+N-K)]\hat{K}$	$\{16(M+N)^2$ $[2(M+N)-$ $(2K-K_{\mathrm{snc}})]\}\hat{K}^2$

The computation complexity analysis of some other 2D DOA estimation algorithms can be found in [20], and the computation times of the algorithms for specific scenarios are provided in the simulation section. It can be seen later that, the computation time of the proposed algorithms is much smaller than 2D I-MUSIC, although a little more than some other reduced-dimension DOA estimation algorithms.

The proposed algorithm is basically a spatial searching type of algorithms, so the off-grid problem cannot be avoided. That is, if the selected angle set does not contain the true DOAs, then the estimated DOAs always have bias compared to the true values. This problem can be mitigated by increasing the density of the searching angles, but the computational cost will be greatly increased.

One widely accepted solution to this problem is to adopt a two-step approach. In the first step, the search step size can be relatively large, and a rough estimation of the source directions can then be obtained. In the second step, to improve the estimation accuracy, the step size can be reduced to a much smaller value, but we only need to search the very small areas around those source directions obtained in the first step. In this way, the computational complexity will be significantly reduced compared to a full search over all the whole angle region with a very dense uniform grid.

E. Extended Array Models Analysis

Although the algorithms proposed in the paper are based on the L-shaped array, they can also be applied to some other array structures. Both DB-MUSIC-M and DB-MUSIC are based on the operation to transform the steering vector into the product of two parts which are related to α and β, separately. As an example, like the L-shaped array, the steering vector of the uniform planar array can also be transformed into such a product, as explained below.

Suppose a uniform planar array contains N rows and M columns aligned with the x-axis and y-axis, separately. The steering matrix of the array can be written as

$$\hat{\mathbf{A}}(\boldsymbol{\alpha},\boldsymbol{\beta}) = [\hat{\mathbf{a}}_x(\alpha_1)\otimes\hat{\mathbf{a}}_y(\beta_1),\ldots,\hat{\mathbf{a}}_x(\alpha_K)\otimes\hat{\mathbf{a}}_y(\beta_K)] \quad (39)$$

with

$$\begin{aligned}&\hat{\mathbf{a}}_x(\alpha_k) = [1,\exp(-j2\pi d\cos\alpha_k/\lambda),\ldots,\\&\exp(-j2\pi d(N-1)\cos\alpha_k/\lambda)]^{\mathrm{T}}\\&\hat{\mathbf{a}}_y(\beta_k) = [1,\exp(-j2\pi d\cos\beta_k/\lambda),\ldots,\\&\exp(-j2\pi d(M-1)\cos\beta_k/\lambda)]^{\mathrm{T}}\end{aligned} \quad (40)$$

This steering vector can be written as

$$\hat{\mathbf{a}}_x(\alpha_k)\otimes\hat{\mathbf{a}}_y(\beta_k) = [\hat{\mathbf{a}}_x(\alpha_k)\otimes\mathbf{I}_M]\cdot\hat{\mathbf{a}}_y(\beta_k), \quad (41)$$

which is a product of two vectors related to α_k and β_k, separately. Then, the principle of the DM-MUSIC-M or DB-MUSIC algorithm can be applied.

IV. Simulation Results

In this section, the proposed DB-MUSIC-M and DB-MUSIC algorithms are compared with a set of L-shaped array based 2D DOA estimation algorithms, including PM-ES [19], JSVD [13], PARAFAC [14], CODE [6], CESA [7], AAEA [31] and CS2R [20], all of which can handle the mixture of circular and noncircular signals. In these algorithms, PM-ES, JSVD, PARAFAC and CS2R are automatic angle pairing algorithms, while CODE, CESA and AAEA need to do 2D angle pairing manually. Among these algorithms, only DB-MUSIC-M, DB-MUSIC and CS2R can handle the underdetermined case; DB-MUSIC and CS2R can cope with up to $N+M-1$ signals, while for DB-MUSIC-M, it is $2(N+M)-1$. The CRB used in the simulations is $\mathrm{CRB_e}(\alpha,\beta)$ as derived in Appendix A. Note that, the values in the diagonal of $\mathrm{CRB_e}(\alpha,\beta)$ are the mean squared errors of α or β, which are then converted into the overall root mean square error CRB. The parameters are set to be $N = M = 5$, $d = \lambda/2$, and the number of Monte Carlo simulations is 500. JSVD, CESA, CS2R, DB-MUSIC and DB-MUSIC-M algorithms are based on spectrum search, and the angle searching interval is set as $0.1°$. The RMSE of the 2D angles is defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{2\mathrm{N_{um}}K}\sum_{\mathrm{num}=1}^{\mathrm{N_{um}}}\sum_{k=1}^{K}\left[\left(\bar{\hat{\alpha}}_k^{(\mathrm{num})}-\alpha_k\right)^2+\left(\bar{\hat{\beta}}_k^{(\mathrm{num})}-\beta_k\right)^2\right]} \quad (42)$$

where $\mathrm{N_{um}}$ is the number of Monte Carlo simulations, $\bar{\hat{\alpha}}_k^{(\mathrm{num})}$ and $\bar{\hat{\beta}}_k^{(\mathrm{num})}$ denote the estimates of azimuth and elevation at the num-th Monte Carlo simulation, respectively.

In the first set of simulations, the performance of DB-MUSIC and DB-MUSIC-M are compared with those of PM-ES, JSVD, PARAFAC, CODE, CESA, AAEA and CS2R for a mixture of three circular and noncircular signals. The DOAs of the three signals are $(35°, 66°)$, $(75°, 76°)$ and $(115°, 86°)$, and the first two of them are BPSK signals while the last one is circular. The number of snapshots is 2000, and the SNR varies from -5dB to 10dB with a 3dB interval. The RMSE results versus SNR are shown in Fig. 4. Then, the SNR is set to 10dB, and the number of snapshots varies from 200 to 1000 with an interval of 100, with the corresponding results shown in Fig. 5.

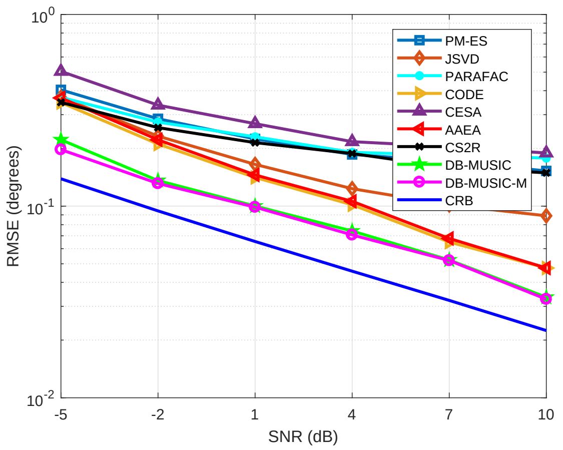

Fig. 4. RMSE versus SNR with 3 signals and 2000 snapshots.

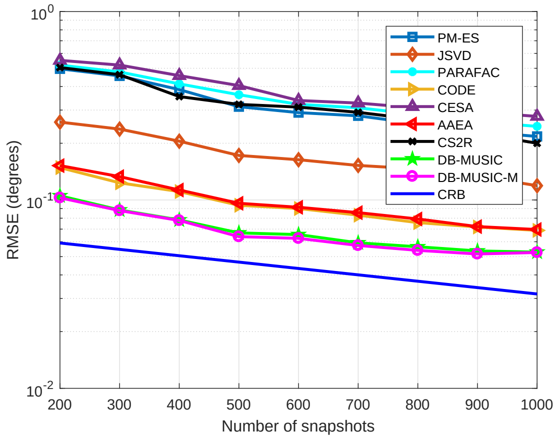

Fig. 5. RMSE versus the number of snapshots with 3 signals and 10dB SNR.

From these two figures, it can be seen that, the DB-MUSIC and DB-MUSIC-M algorithms clearly outperform the other algorithms, while the DB-MUSIC-M performs a little better than DB-MUSIC in the low SNR region around -5dB.

Use the same parameter settings as above, but change the three signals to BPSK, UQPSK and circular signals, separately. The RMSE results versus SNR and the number of snapshots are shown in Figs. 6 and 7, respectively.

It can be seen that the DB-MUSIC and DB-MUSIC-M algorithms still outperform the other algorithms, and RMSE of all the algorithms with non-strictly noncircular signals is lower than that without non-strictly noncircular signals.

In the second set of simulations, the performance of DB-MUSIC and DB-MUSIC-M is compared with that of CS2R for handling five (underdetermined) signals. They have DOAs $(30°, 44°)$, $(50°, 78°)$, $(70°, 82°)$, $(90°, 111°)$ and $(110°, 115°)$, and the first three are BPSK signals while the rest are circular. The number of snapshots is set to 2000, and

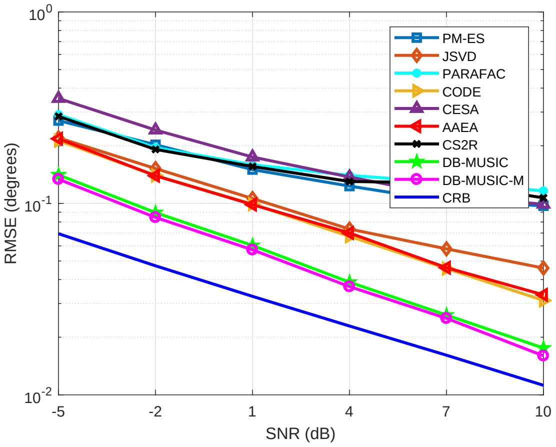

Fig. 6. RMSE versus SNR with 3 signals and 2000 snapshots with non-strictly noncircular signals.

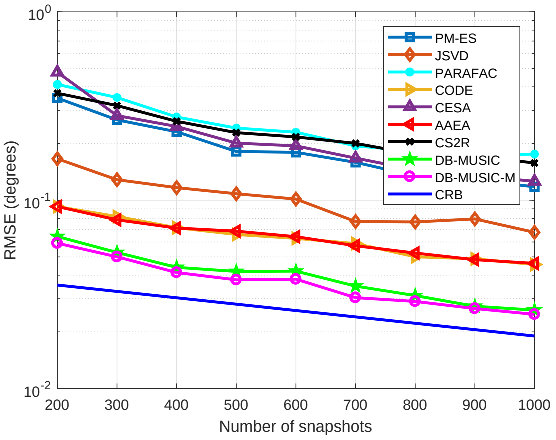

Fig. 7. RMSE versus the number of snapshots with 3 signals and 10dB SNR with non-strictly noncircular signals.

SNR varies from -5dB to 10dB with a 3dB interval. The results are shown in Fig. 8. Then, the SNR is set to 10dB, and the number of snapshots varies from 200 to 1000 with an interval of 100, with the results shown in Fig. 9.

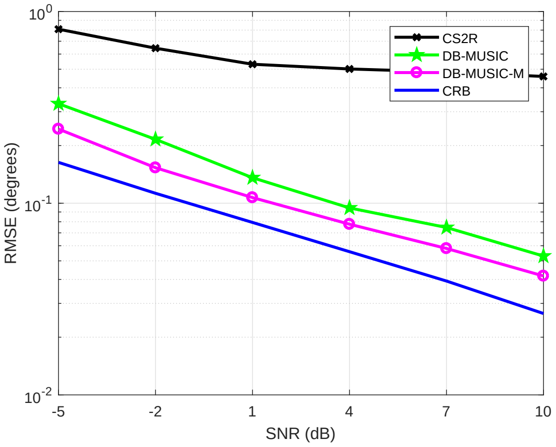

Fig. 8. RMSE versus SNR with 5 signals and 2000 snapshots.

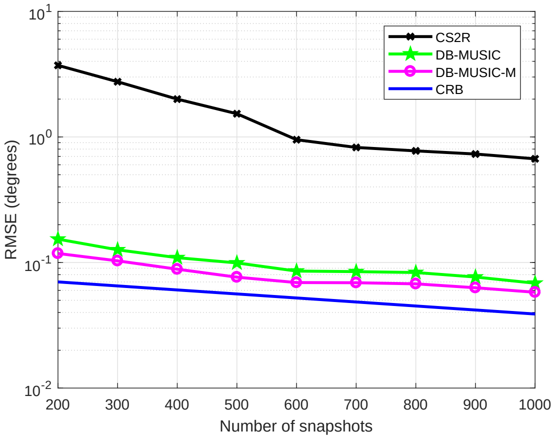

Fig. 9. RMSE versus the number of snapshots with 5 signals and 10dB SNR.

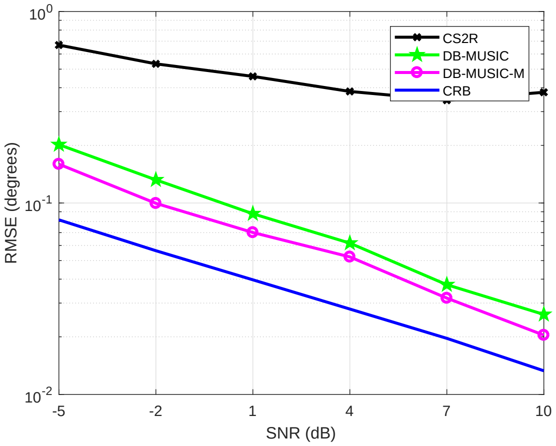

Fig. 10. RMSE versus SNR with 5 signals and 2000 snapshots with non-strictly noncircular signals.

In these two figures, the performance of DB-MUSIC and DB-MUSIC-M is much better than that of CS2R, while the DB-MUSIC-M performs better than DB-MUSIC, with a result closer to the CRB.

With the same parameter settings, the five signals are changed to the following combination: the first two being BPSK, the third one being UQPSK and the last two being circular. The RMSE results versus SNR and the number of snapshots are shown in Figs. 10 and 11, separately.

It can be seen that the DB-MUSIC and DB-MUSIC-M algorithms still perform better than the other algorithms, and the algorithms with non-strictly noncircular signals still have lower RMSE than that without non-strictly noncircular signals.

Moreover, comparing Fig. 4 with Fig. 6 or Fig. 8 with Fig. 10, the same number of signals but with different types is used. It can be seen that RMSEs of all the algorithms decrease with one of the BPSK signals being replaced by the UQPSK signal. In Figs. 4 and 8 or Figs. 6 and 10, a greater number of signals is present and it can be seen that RMSEs of all the algorithms increase with more signals being estimated.

In the third set of simulations, the number of signals to be estimated is increased to nine. In this situation, only DB-MUSIC-M works, while DB-MUSIC, CS2R and all the other algorithms fail. Moreover, the algorithm us-

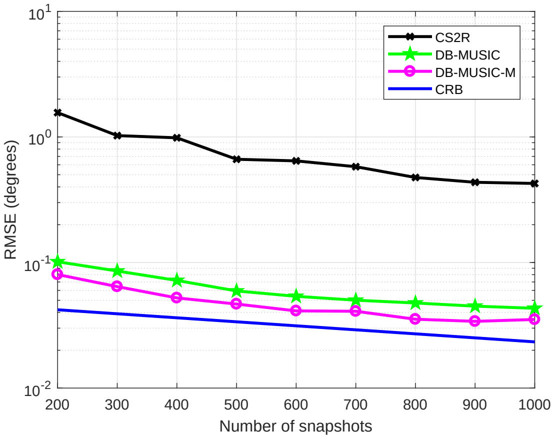

Fig. 11. RMSE versus the number of snapshots with 5 signals and 10dB SNR with non-strictly noncircular signals.

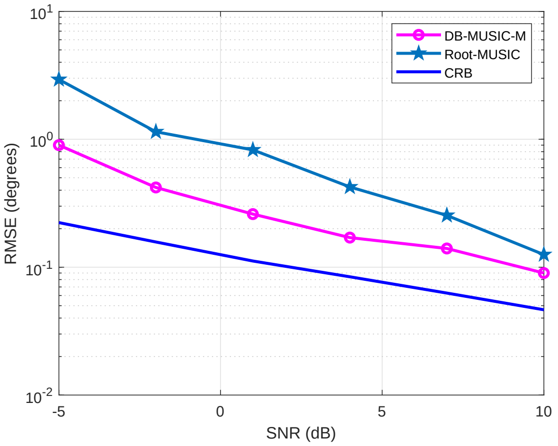

Fig. 12. RMSE versus SNR with 9 signals and 2000 snapshots.

ing $\det[\mathbf{V}_{\mathrm{e}}(\alpha,\beta)] = 0$ is also considered to show its performance. The DOAs of the nine signals are set to be $(33^\circ, 24^\circ)$, $(48^\circ, 116^\circ)$,$(63^\circ, 157^\circ)$,$(78^\circ, 38^\circ)$,$(93^\circ, 86^\circ)$, $(108^\circ, 137^\circ)$, $(123^\circ, 23^\circ)$, $(138^\circ, 34^\circ)$ and $(153^\circ, 111^\circ)$, and the first five are BPSK signals while the rest are circular. The number of snapshots is set to be 2000, and SNR varies from -5dB to 10dB with a 3dB interval, with its results shown in Fig. 12, where Root-MUSIC represents the results using the algorithm based on $\det[\mathbf{V}_{\mathrm{e}}(\alpha,\beta)] = 0$ as analyzed in Sec. III. C. Then, the SNR is set to 10dB, and the number of snapshots varies from 500 to 3000 with an interval of 500. Fig. 13 shows the results.

It can be seen that DB-MUSIC-M still works and its performance is still close to the CRB, while the algorithm with $\det[\mathbf{V}_{\mathrm{e}}(\alpha,\beta)] = 0$ has much larger estimation errors. Note that when calculating the RMSE of the algorithm with $\det[\mathbf{V}_{\mathrm{e}}(\alpha,\beta)] = 0$, we have removed those results with false peaks; otherwise, its RMSE results would be much worse.

In the fourth set of simulations, we consider a scenario with two closely spaced noncircular signals in one dimension, with one at $(53^\circ, 43^\circ)$, and the other at $(43^\circ, \beta)$, and β varies from 29° to 42° with an interval of 1°, as shown in Fig. 14. The SNR is set to 10dB, and the number of snapshots is set to 2000. The RMSE results with respect to β are shown in Fig. 15.

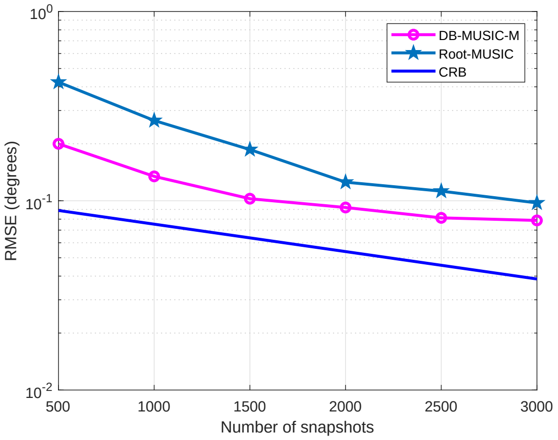

Fig. 13. RMSE versus the number of snapshots with 9 signals and 10dB SNR.

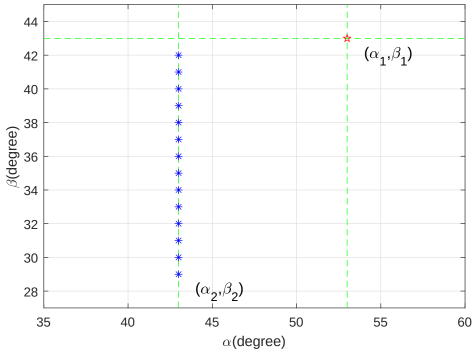

Fig. 14. The DOAs setting.

It can be seen that, the RMSE of PARAFAC, CS2R, DB-MUSIC and DB-MUSIC-M algorithms keeps relatively stable with the varying β, following a similar trend in CRB, while the other algorithms have almost failed when the two β angles are close. Furthermore, the RMSEs of DB-MUSIC and DB-MUSIC-M are lower than those of PARAFAC and CS2R, and DB-MUSIC-M is better than DB-MUSIC.

Finally, we would like to show the significantly reduced

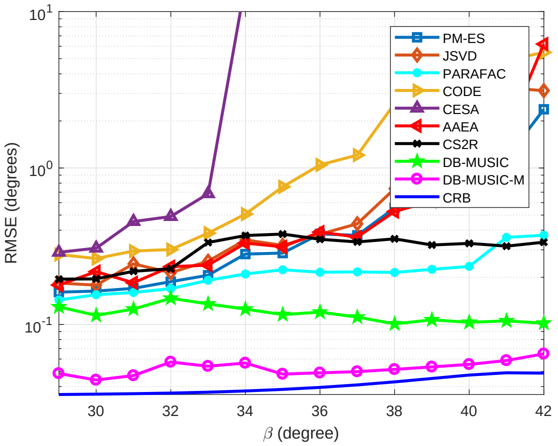

Fig. 15. RMSE versus the distance between two noncircular signals at 10dB SNR and 2000 snapshots.

TABLE II
THE COMPUTATION TIME OF THE ALGORITHMS

Algorithm Type	Computation Time (s)
PM-ES	0.0088
JSVD	0.1146
PARAFAC	0.0888
CODE	0.7471
CESA	0.1081
AAEA	0.0071
CS2R	0.1177
DB-MUSIC	0.2622
DB-MUSIC-M	1.2446
2D I-MUSIC	184.56

computation time by the proposed DB-MUSIC-M algorithm in comparison with the 2D direct search solution of I-MUSIC using Eq.(13). The SNR is set to 4dB, the number of snapshots is set to 2000, and the DOA of a BPSK signal is set as $(35°, 66°)$. The CPU and RAM of the computer are i7-10875h and 16GB, separately, and the number of search angles $\hat{K}$ is 1800. The computation time of all the algorithms are shown in Table II.

It can be seen that the computation time of the proposed DB-MUSIC-M is much smaller than 2D I-MUSIC, although a little greater than those of other algorithms. The computation time of DB-MUSIC-M and 2D I-MUSIC are $1.2446s$ and $184.56s$, separately, i.e., the computation time of 2D I-MUSIC is almost 150 times that used by the DB-MUSIC-M.

V. CONCLUSION

A 2D DOA estimation algorithm called DB-MUSIC-M has been proposed based on the L-shaped array for a mixture of circular and noncircular signals, while for the special case with circular signals only, the developed algorithm is called DB-MUSIC. The proposed algorithms employ a differentiation operation on the objective function of the 2D I-MUSIC algorithm to transform the 2D DOA estimation problem into a single 1D DOA estimation, which greatly reduces the computational complexity. Moreover, there is no need for angle pairing in the process and they can handle the angle ambiguity problem effectively. As demonstrated by computer simulations, the DB-MUSIC-M and DB-MUSIC algorithms have outperformed existing corresponding 2D DOA estimation algorithms in the overdetermined case, and the DB-MUSIC-M algorithm performs better than the DB-MUSIC and other algorithms when the number of signals is larger than the number of physical sensors. Moreover, significantly reduced computation time was achieved by the proposed solution in comparison with a direct 2D search method based on I-MUSIC.

APPENDIX A

In order to assess the performance of the proposed algorithms, the CRB for 2D DOA estimation for a mixture of circular and noncircular signals is derived in this part. According to the result of 1D CRB in [29], the 2D CRB result is

$$\mathrm{CRB}_{\mathrm{e}}(\alpha,\beta)=\frac{2}{\mathrm{S}_{\mathrm{nap}}}\left(\mathbb{C}^{\mathrm{H}}\Pi_{\mathbb{D}}^{\perp}\mathbb{C}\right)^{-1} \tag{43}$$

with

$$\begin{aligned}
&\Pi_{\mathbb{D}}^{\perp}=\mathbf{I}_{4(N+M)^2}-\mathbb{D}\left(\mathbb{D}^{\mathrm{H}}\mathbb{D}\right)^{-1}\mathbb{D}^{\mathrm{H}},\\
&\mathbb{C}=\mathbb{F}\left[\frac{\partial\mathbf{r}_{\mathrm{e}}}{\partial\boldsymbol{\alpha}^{\mathrm{T}}},\frac{\partial\mathbf{r}_{\mathrm{e}}}{\partial\boldsymbol{\beta}^{\mathrm{T}}}\right],\\
&\mathbb{D}=\mathbb{F}\left[\frac{\partial\mathbf{r}_{\mathrm{e}}}{\partial\boldsymbol{\eta}^{\mathrm{T}}},\frac{\partial\mathbf{r}_{\mathrm{e}}}{\partial\boldsymbol{\psi}^{\mathrm{T}}},\frac{\partial\mathbf{r}_{\mathrm{e}}}{\partial\boldsymbol{\rho}^{\mathrm{T}}},\frac{\partial\mathbf{r}_{\mathrm{e}}}{\partial\sigma^{2}}\right],\\
&\mathbb{F}=\left(\mathbf{R}_{\mathrm{e}}^{\mathrm{T}}\otimes\mathbf{R}_{\mathrm{e}}\right)^{-\frac{1}{2}},\ \mathbf{r}_{\mathrm{e}}=\mathrm{vec}\left(\mathbf{R}_{\mathrm{e}}\right),
\end{aligned} \tag{44}$$

where $\mathrm{S}_{\mathrm{nap}}$ is the number of snapshots, $\otimes$ is the Kronecker product operator, and $\mathrm{vec}(\cdot)$ is the vectorization operation.

The matrices $\mathbb{C}$ and $\mathbb{D}$ in the proposed $\mathrm{CRB}_{\mathrm{e}}(\alpha,\beta)$ can be further represented as

$$\begin{aligned}
\mathbb{C}&=\mathbb{F}\mathbb{O}\begin{bmatrix}\left(\frac{\partial\mathbf{r}_{\mathrm{d}}}{\partial\boldsymbol{\alpha}^{\mathrm{T}}}\right)^{\mathrm{T}},\left(\frac{\partial\mathbf{r}_{\mathrm{s}}^{*}}{\partial\boldsymbol{\alpha}^{\mathrm{T}}}\right)^{\mathrm{T}},\left(\frac{\partial\mathbf{r}_{\mathrm{s}}}{\partial\boldsymbol{\alpha}^{\mathrm{T}}}\right)^{\mathrm{T}},\left(\frac{\partial\mathbf{r}_{\mathrm{d}}^{*}}{\partial\boldsymbol{\alpha}^{\mathrm{T}}}\right)^{\mathrm{T}}\\ \left(\frac{\partial\mathbf{r}_{\mathrm{d}}}{\partial\boldsymbol{\beta}^{\mathrm{T}}}\right)^{\mathrm{T}},\left(\frac{\partial\mathbf{r}_{\mathrm{s}}^{*}}{\partial\boldsymbol{\beta}^{\mathrm{T}}}\right)^{\mathrm{T}},\left(\frac{\partial\mathbf{r}_{\mathrm{s}}}{\partial\boldsymbol{\beta}^{\mathrm{T}}}\right)^{\mathrm{T}},\left(\frac{\partial\mathbf{r}_{\mathrm{d}}^{*}}{\partial\boldsymbol{\beta}^{\mathrm{T}}}\right)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\\
\mathbb{D}&=\mathbb{F}\mathbb{O}\begin{bmatrix}\frac{\partial\mathbf{r}_{\mathrm{d}}}{\partial\boldsymbol{\eta}^{\mathrm{T}}}&\frac{\partial\mathbf{r}_{\mathrm{d}}}{\partial\boldsymbol{\psi}^{\mathrm{T}}}&\frac{\partial\mathbf{r}_{\mathrm{d}}}{\partial\boldsymbol{\rho}^{\mathrm{T}}}&\frac{\partial\mathbf{r}_{\mathrm{d}}}{\partial\sigma^{2}}\\ \frac{\partial\mathbf{r}_{\mathrm{s}}^{*}}{\partial\boldsymbol{\eta}^{\mathrm{T}}}&\frac{\partial\mathbf{r}_{\mathrm{s}}^{*}}{\partial\boldsymbol{\psi}^{\mathrm{T}}}&\frac{\partial\mathbf{r}_{\mathrm{s}}^{*}}{\partial\boldsymbol{\rho}^{\mathrm{T}}}&\frac{\partial\mathbf{r}_{\mathrm{s}}^{*}}{\partial\sigma^{2}}\\ \frac{\partial\mathbf{r}_{\mathrm{s}}}{\partial\boldsymbol{\eta}^{\mathrm{T}}}&\frac{\partial\mathbf{r}_{\mathrm{s}}}{\partial\boldsymbol{\psi}^{\mathrm{T}}}&\frac{\partial\mathbf{r}_{\mathrm{s}}}{\partial\boldsymbol{\rho}^{\mathrm{T}}}&\frac{\partial\mathbf{r}_{\mathrm{s}}}{\partial\sigma^{2}}\\ \frac{\partial\mathbf{r}_{\mathrm{d}}^{*}}{\partial\boldsymbol{\eta}^{\mathrm{T}}}&\frac{\partial\mathbf{r}_{\mathrm{d}}^{*}}{\partial\boldsymbol{\psi}^{\mathrm{T}}}&\frac{\partial\mathbf{r}_{\mathrm{d}}^{*}}{\partial\boldsymbol{\rho}^{\mathrm{T}}}&\frac{\partial\mathbf{r}_{\mathrm{d}}^{*}}{\partial\sigma^{2}},\end{bmatrix},
\end{aligned} \tag{45}$$

with

$$\begin{aligned}
&\mathbb{O}=\left[\mathbf{I}_2\otimes\left(\textstyle\sum_{m=1}^{N+M}\sum_{j=1}^{2}O_{m,j}\otimes O_{m,j}^{\mathrm{T}}\right)\otimes\mathbf{I}_{N+M}\right]^{-1},\\
&\mathbf{R}_{\mathrm{dif}}=\mathrm{E}[\mathbf{z}(t)\mathbf{z}^{\mathrm{H}}(t)],\ \mathbf{R}_{\mathrm{sum}}=\mathrm{E}[\mathbf{z}(t)\mathbf{z}^{\mathrm{T}}(t)],\\
&\mathbf{r}_{\mathrm{d}}=\mathrm{vec}\left(\mathbf{R}_{\mathrm{dif}}\right)=\mathbf{T}_{\mathrm{d}}\left(\boldsymbol{\alpha},\boldsymbol{\beta}\right)\boldsymbol{\eta}+\sigma^2\,\mathrm{vec}\left(\mathbf{I}_{N+M}\right),\\
&\mathbf{r}_{\mathrm{s}}=\mathrm{vec}\left(\mathbf{R}_{\mathrm{sum}}\right)=\mathbf{T}_{\mathrm{s}}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right)\mathrm{diag}\{\boldsymbol{\eta}_{\mathrm{nc}}\}\,\mathrm{diag}\{\boldsymbol{\rho}\}\boldsymbol{\psi}_{\mathrm{e}},\\
&\mathbf{T}_{\mathrm{d}}\left(\boldsymbol{\alpha},\boldsymbol{\beta}\right)=\mathbf{A}^{*}\left(\boldsymbol{\alpha},\boldsymbol{\beta}\right)\odot\mathbf{A}\left(\boldsymbol{\alpha},\boldsymbol{\beta}\right),\\
&\mathbf{T}_{\mathrm{s}}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right)=\mathbf{A}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right)\odot\mathbf{A}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right),\\
&\boldsymbol{\psi}_{\mathrm{e}}=\left[e^{j\psi_1},e^{j\psi_2},\ldots,e^{j\psi_{K_{\mathrm{nc}}}}\right]^{\mathrm{T}},\\
&\boldsymbol{\eta}_{\mathrm{nc}}=[\eta_1,\eta_2,\cdots,\eta_{K_{\mathrm{nc}}}]^{\mathrm{T}},\\
&\boldsymbol{\alpha}_{\mathrm{nc}}=[\alpha_1,\alpha_2,\cdots,\alpha_{K_{\mathrm{nc}}}]^{\mathrm{T}},\\
&\boldsymbol{\beta}_{\mathrm{nc}}=[\beta_1,\beta_2,\cdots,\beta_{K_{\mathrm{nc}}}]^{\mathrm{T}},
\end{aligned} \tag{46}$$

where, $O_{h,l}$ is an $(N+M)\times 2$ matrix with one at the (h,l)th position and zeros elsewhere, and $\odot$ denotes the Khatri-Rao product (column-wise Kronecker product).

Using (46), we can calculate the derivatives in (45), and the results are listed below.

$$\begin{aligned}
&\frac{\partial \mathbf{r}_{\mathrm{d}}}{\partial \boldsymbol{\alpha}^{\mathrm{T}}} = \mathbf{T}'_{\mathrm{d}\alpha}(\boldsymbol{\alpha},\boldsymbol{\beta})\operatorname{diag}\{\boldsymbol{\eta}\},\quad \frac{\partial \mathbf{r}_{\mathrm{d}}}{\partial \boldsymbol{\beta}^{\mathrm{T}}} = \mathbf{T}'_{\mathrm{d}\beta}(\boldsymbol{\alpha},\boldsymbol{\beta})\operatorname{diag}\{\boldsymbol{\eta}\},\\
&\frac{\partial \mathbf{r}_{\mathrm{d}}}{\partial \boldsymbol{\eta}^{\mathrm{T}}} = \mathbf{T}_{\mathrm{d}}\left(\boldsymbol{\alpha},\boldsymbol{\beta}\right),\quad \frac{\partial \mathbf{r}_{\mathrm{d}}}{\partial \boldsymbol{\psi}^{\mathrm{T}}} = \mathbf{0}_{(N+M)^2\times K_{\mathrm{nc}}},\\
&\frac{\partial \mathbf{r}_{\mathrm{d}}}{\partial \boldsymbol{\rho}^{\mathrm{T}}} = \mathbf{0}_{(N+M)^2\times K_{\mathrm{nc}}},\quad \frac{\partial \mathbf{r}_{\mathrm{d}}}{\partial \sigma^2} = \operatorname{vec}(\mathbf{I}_{N+M}),\\
&\frac{\partial \mathbf{r}_{\mathrm{s}}}{\partial \boldsymbol{\alpha}^{\mathrm{T}}} =\\
&\left[\mathbf{T}'_{\mathrm{s}\alpha}(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}})\operatorname{diag}\{\boldsymbol{\eta}_{\mathrm{nc}}\}\operatorname{diag}\{\boldsymbol{\rho}\}\operatorname{diag}\{\boldsymbol{\psi}_{\mathrm{e}}\},\mathbf{0}_{(N+M)^2\times K_{\mathrm{c}}}\right],\\
&\frac{\partial \mathbf{r}_{\mathrm{s}}}{\partial \boldsymbol{\beta}^{\mathrm{T}}} =\\
&\left[\mathbf{T}'_{\mathrm{s}\beta}(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}})\operatorname{diag}\{\boldsymbol{\eta}_{\mathrm{nc}}\}\operatorname{diag}\{\boldsymbol{\rho}\}\operatorname{diag}\{\boldsymbol{\psi}_{\mathrm{e}}\},\mathbf{0}_{(N+M)^2\times K_{\mathrm{c}}}\right],\\
&\frac{\partial \mathbf{r}_{\mathrm{s}}}{\partial \boldsymbol{\eta}^{\mathrm{T}}} = \left[\mathbf{T}_{\mathrm{s}}(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}})\operatorname{diag}\{\boldsymbol{\rho}\}\operatorname{diag}\{\boldsymbol{\psi}_{\mathrm{e}}\},\mathbf{0}_{(N+M)^2\times K_{\mathrm{c}}}\right],\\
&\frac{\partial \mathbf{r}_{\mathrm{s}}}{\partial \boldsymbol{\psi}^{\mathrm{T}}} = j\mathbf{T}_{\mathrm{s}}(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}})\operatorname{diag}\{\boldsymbol{\eta}_{\mathrm{nc}}\}\operatorname{diag}\{\boldsymbol{\rho}\}\operatorname{diag}\{\boldsymbol{\psi}_{\mathrm{e}}\},\\
&\frac{\partial \mathbf{r}_{\mathrm{s}}}{\partial \boldsymbol{\rho}^{\mathrm{T}}} = \mathbf{T}_{\mathrm{s}}(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}})\operatorname{diag}\{\boldsymbol{\eta}_{\mathrm{nc}}\}\operatorname{diag}\{\boldsymbol{\psi}_{\mathrm{e}}\},\frac{\partial \mathbf{r}_{\mathrm{s}}}{\partial \sigma^2} = \mathbf{0}_{(N+M)^2\times 1},
\end{aligned}\tag{47}$$

with

$$\begin{aligned}
&\mathbf{T}'_{\mathrm{d}\alpha}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \mathbf{A}'^{*}_{\alpha}(\boldsymbol{\alpha},\boldsymbol{\beta})\odot\mathbf{A}(\boldsymbol{\alpha},\boldsymbol{\beta})+\\
&\qquad\qquad \mathbf{A}^{*}(\boldsymbol{\alpha},\boldsymbol{\beta})\odot\mathbf{A}'_{\alpha}(\boldsymbol{\alpha},\boldsymbol{\beta}),\\
&\mathbf{T}'_{\mathrm{d}\beta}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \mathbf{A}'^{*}_{\beta}(\boldsymbol{\alpha},\boldsymbol{\beta})\odot\mathbf{A}(\boldsymbol{\alpha},\boldsymbol{\beta})+\mathbf{A}^{*}(\boldsymbol{\alpha},\boldsymbol{\beta})\odot\mathbf{A}'_{\beta}(\boldsymbol{\alpha},\boldsymbol{\beta}),\\
&\mathbf{T}'_{\mathrm{s}\alpha}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right) = \mathbf{A}'_{\alpha}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right)\odot\mathbf{A}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right)+\\
&\qquad\qquad \mathbf{A}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right)\odot\mathbf{A}'_{\alpha}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right),\\
&\mathbf{T}'_{\mathrm{s}\beta}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right) = \mathbf{A}'_{\beta}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right)\odot\mathbf{A}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right)+\\
&\qquad\qquad \mathbf{A}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right)\odot\mathbf{A}'_{\beta}\left(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}\right),\\
&\mathbf{A}'_{\alpha}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \left[\frac{\partial \mathbf{a}\left(\alpha_1,\beta_1\right)}{\partial \alpha_1},\frac{\partial \mathbf{a}\left(\alpha_2,\beta_2\right)}{\partial \alpha_2},\cdots,\frac{\partial \mathbf{a}\left(\alpha_K,\beta_K\right)}{\partial \alpha_K}\right],\\
&\mathbf{A}'_{\beta}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \left[\frac{\partial \mathbf{a}\left(\alpha_1,\beta_1\right)}{\partial \beta_1},\frac{\partial \mathbf{a}\left(\alpha_2,\beta_2\right)}{\partial \beta_2}\cdots,\frac{\partial \mathbf{a}\left(\alpha_K,\beta_K\right)}{\partial \beta_K}\right],\\
&\mathbf{A}'_{\alpha}(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}) =\\
&\qquad \left[\frac{\partial \mathbf{a}\left(\alpha_1,\beta_1\right)}{\partial \alpha_1},\frac{\partial \mathbf{a}\left(\alpha_2,\beta_2\right)}{\partial \alpha_2},\cdots,\frac{\partial \mathbf{a}\left(\alpha_{K_{\mathrm{nc}}},\beta_{K_{\mathrm{nc}}}\right)}{\partial \alpha_{K_{\mathrm{nc}}}}\right],\\
&\mathbf{A}'_{\beta}(\boldsymbol{\alpha}_{\mathrm{nc}},\boldsymbol{\beta}_{\mathrm{nc}}) =\\
&\qquad \left[\frac{\partial \mathbf{a}\left(\alpha_1,\beta_1\right)}{\partial \beta_1},\frac{\partial \mathbf{a}\left(\alpha_2,\beta_2\right)}{\partial \beta_2}\cdots,\frac{\partial \mathbf{a}\left(\alpha_{K_{\mathrm{nc}}},\beta_{K_{\mathrm{nc}}}\right)}{\partial \beta_{K_{\mathrm{nc}}}}\right].
\end{aligned}\tag{48}$$

References

[1] J. Gu, W. Zhu, and M. N. S. Swamy, “Joint 2-D DOA estimation via sparse L-shaped array,” *IEEE Transactions on Signal Processing*, vol. 63, no. 5, pp. 1171–1182, Mar. 2015.

[2] H. Chen, Y. Liu, Q. Wang, W. Liu, and G. Wang, “Two-dimensional angular parameter estimation for noncircular incoherently distributed sources based on an L-shaped array,” *IEEE Sensors Journal*, vol. 20, pp. 13 704–13 715, Jul. 2020.

[3] B. Yya, B. Xma, B. Yha, and B. Gja, “2-D DOA estimation via correlation matrix reconstruction for nested L-shaped array,” *Digital Signal Processing*, vol. 98, 2020.

[4] J. Li and X. Zhang, “Combined real-valued subspace based two dimensional angle estimation using L-shaped array,” *Digital Signal Processing*, vol. 83, pp. 157–164, Dec. 2018.

[5] S. Kikuchi, H. Tsuji, and A. Sano, “Pair-matching method for estimating 2-D angle of arrival with a cross-correlation matrix,” *IEEE Antennas and Wireless Propagation Letters*, vol. 5, pp. 35–40, 2006.

[6] G. Wang, J. Xin, N. Zheng, and A. Sano, “Computationally efficient subspace-based method for two-dimensional direction estimation with L-shaped array,” *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3197–3212, Jul. 2011.

[7] N. Xi and L. P. Li, “A computationally efficient subspace algorithm for 2-D DOA estimation with L-shaped array,” *IEEE Signal Processing Letters*, vol. 21, no. 8, pp. 971–974, Aug. 2014.

[8] S. Ta, H. Wang, and H. Chen, “Two-dimensional direction-of-arrival estimation and pairing using L-shaped arrays,” *Signal, Image Video Process*, vol. 10, no. 8, pp. 1511–1518, 2016.

[9] Z. Li, X. Zhang, and J. Shen, “2D-DOA estimation of strictly noncircular sources utilizing connection-matrix for L-shaped array,” *IEEE Wireless Communication Letters*, pp. 1–1, Oct. 2020.

[10] Z. Cheng, Y. Zhao, Y. Zhu, P. Shui, and H. Li, “Sparse representation based two-dimensional direction-of-arrival estimation method with L-shaped array,” *IET Radar, Sonar and Navigation*, vol. 10, no. 5, pp. 976–982, 2016.

[11] Z. Li and X. Zhang, “Monostatic MIMO radar with nested L-shaped array: Configuration design, DOF and DOA estimation,” *Digital Signal Processing*, vol. 108, Jan. 2018.

[12] W. Chang, X. Li, and J. Wang, “Stokes parameters and 2-D DOAs estimation of polarized sources with an L-shaped coprime array,” *Digital Signal Processing*, vol. 78, Jul. 2018.

[13] J. Gu and P. Wei, “Joint SVD of two cross-correlation matrices to achieve automatic pairing in 2-D angle estimation problems,” *IEEE Antennas and Wireless Propagation Letters*, vol. 6, pp. 553–556, 2007.

[14] D. Liu and J. L. Liang, “L-shaped array-based 2-D DOA estimation using parallel factor analysis,” in *the 8th World Congress on Intelligent Control and Automation*, Jinan, 2010, pp. 6949–6952.

[15] S. O. Al-Jazzar, D. McLernon, and M. A. Smadi, “SVD-based joint azimuth/elevation estimation with automatic pairing,” *Signal Processing*, vol. 90, no. 5, pp. 1669–1675, 2010.

[16] J. Liang and D. Liu, “Joint elevation and azimuth direction finding using L-shaped array,” *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 6, pp. 2136–2141, Jun. 2010.

[17] Y. Yang, X. Mao, Y. Hou, and G. Jiang, “2-D DOA estimation via correlation matrix reconstruction for nested L-shaped array,” *Digital Signal Processing*, vol. 98, Mar. 2020.

[18] N. Tayem, K. Majeed, and A. A. Hussain, “Two-dimensional DOA estimation using cross-correlation matrix with L-shaped array,” *IEEE Antennas and Wireless Propagation Letters*, vol. 15, pp. 1077–1080, Dec. 2016.

[19] Y. Dong, C. Dong, J. Xu, and G. Zhao, “Computationally efficient 2-D DOA estimation for L-shaped array with automatic pairing,” *IEEE Antennas and Wireless Propagation Letters*, vol. 15, pp. 1669–1672, 2016.

[20] Y. Dong, C. Dong, W. Liu, H. Chen, and G. Zhao, “2-D DOA estimation for L-shaped array with array aperture and snapshots extension techniques,” *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 495–499, Apr. 2017.

[21] X. Zhang, L. Xu, L. Xu, and D. Xu, “Direction of departure (DOD) and direction of arrival (DOA) estimation in MIMO radar with reduced-dimension MUSIC,” *IEEE Communications Letters*, vol. 14, no. 12, pp. 1161–1163, Dec. 2010.

[22] J. Cai, W. Liu, R.Zong, and B. Wu, “Sparse array extension for non-circular signals with subspace and compressive sensing based DOA estimation methods,” *Signal Processing*, vol. 145, pp. 59–67, 2018.

[23] F. Gao, A. Nallanathan, and Y. Wang, “Improved MUSIC under the coexistence of both circular and noncircular sources,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, 2008.

[24] J. Zhang, T. Qiu, S. Luan, and H. Li, “Bounded non-linear covariance based ESPRIT method for noncircular signals in presence of impulsive noise,” *Digital Signal Processing*, vol. 87, pp. 104–111, Apr. 2019.

[25] J. Cai, W. Liu, R. Zong, and Y. Y. Dong, “A MUSIC-type DOA estimation method based on sparse arrays for a mixture of circular and non-circular signals,” in *IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Macau, China, Aug. 2020, pp. 21–23.

[26] J. Cai, H. Zhang, W. Liu, F. Tan, and Y. Y. Dong, “A derivative-based MUSIC algorithm for two-dimensional angle estimation employing an L-shaped array,” in *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Louisville, Kentucky, USA, Dec. 2020, pp. 9–11.

[27] H. Chen, C. Hou, W. P. Zhu, W. Liu, Y. Y. Dong, Z. Peng, and Q. Wang, “ESPRIT-like two-dimensional direction finding for mixed circular and strictly noncircular sources based on joint diagonalization,” *Signal Processing*, vol. 141, Dec. 2017.

[28] L. Zhang, W. Lv, X. Zhang, and S. Li, “2D-DOA estimation of noncircular signals for uniform rectangular array via NC-PARAFAC method,” *International Journal of Electronics*, vol. 103, pp. 1839–1856, Feb. 2016.

[29] J. Cai, Y. Liang, W. Liu, and Y. Y. Dong, “Stochastic Cramer-Rao bound for DOA estimation with a mixture of circular and non-circular signals,” *IEEE Access*, vol. 8, pp. 226 370–226 379, Nov. 2020.

[30] Z. Yang and J. Tang, “Source resolvability of spatial-smoothing-based subspace methods: a Hadamard product perspective,” *IEEE Transactions on Signal Processing*, vol. 67, pp. 2543–2553, May 2019.

[31] X. Nie and P. Wei, “Array aperture extension algorithm for 2-D DOA estimation with L-shaped array,” *Progress in Electromagnetics Research Letters*, vol. 52, pp. 63–69, 2015.