

Article

An Improved Two-Dimensional Direction-Of-Arrival Estimation Algorithm for L-Shaped Nested Arrays with Small Sample Sizes

Xiaofeng Gao, Xinhong Hao , Ping Li and Guolin Li

Science and Technology on Electromechanical Dynamic Control Laboratory, School of Mechatronical Engineering, Beijing Institute of Technology, 5 South Zhongguancun Street, Haidian District, Beijing 100081, China; nmbtgxf@bit.edu.cn (X.G.); liping85@bit.edu.cn (P.L.); 7920161015@bit.edu.cn (G.L.)

* Correspondence: haoxinhong@bit.edu.cn; Tel.: +86-010-6891-4850

Received: 27 March 2019; Accepted: 8 May 2019; Published: 10 May 2019



Abstract: In this paper, an improved two-dimensional (2-D) direction of arrival (DOA) estimation algorithm for L-shaped nested arrays is proposed. Unlike the approach for a classical nested array, which use the auto-correlation matrix (ACM) to increase the degrees of freedom (DOF), we utilize the cross-correlation matrix (CCM) of different sub-arrays to generate two long consecutive virtual arrays. These acquire a large number of DOF without redundant elements and eliminate noise effects. Furthermore, we reconstruct the CCM based on the singular value decomposition (SVD) operation in order to reduce the perturbation of noise for small numbers of samples. To cope with the matrix rank deficiency of the virtual arrays, we construct the full rank equivalent covariance matrices by using the output and its conjugate vector of virtual arrays. The unitary estimation of signal parameters via rotational invariance technique (ESPRIT) is then performed on the covariance matrices to obtain the DOA of incident signals with low computational complexity. Finally, angle pairing is achieved by deriving the equivalent signal vector of the virtual arrays using the estimated angles. Numerical simulation results show that the proposed algorithm not only provides more accurate 2-D DOA estimation performance with low complexity, but also achieves angle estimation for small numbers of samples compared to existing similar methods.

Keywords: 2-D DOA estimation; L-shaped nested arrays; small numbers of samples; cross-correlation matrix

1. Introduction

As an important field of array signal processing, direction of arrival (DOA) estimation has been applied in a wide range of applications such as wireless communications, sonar, radar, and acoustic localization [1–3]. Furthermore, signal sources location and identification are essential for monitoring the electromagnetic environment, and several different technologies based on parameter estimation have been applied to it [4–7]. The key problem of parameter estimation is estimating the azimuth and elevation angles. To obtain these angles, several DOA estimation methods have been proposed, such as multiple signal classification (MUSIC) [8], Time-Reversal MUSIC (TR-MUSIC) [9–11], Root-MUSIC [12], estimation of signal parameters via rotational invariance technique (ESPRIT) [13], and so on. Compared with 1-D DOA estimation, the study of 2-D DOA estimation, which can obtain the azimuth and elevation angles simultaneously, has attracted much attention. Numerous high-resolution 2-D DOA estimation methods with different structured arrays have been proposed, including the two-parallel arrays [14,15], circular arrays [16,17], uniform rectangular arrays [18–20], and L-shaped arrays [21–30]. Due to the wider coverage area and lower Cram'er-Rao boundary (CRB) [31], great efforts have been focused on 2-D DOA estimation for L-shaped arrays.

Due to the geometric configuration of L-shaped arrays, the elevation and azimuth angles can be estimated independently from each linear arrays. As the classical subspace method, MUSIC can be directly employed to the received data matrix of L-shaped arrays [21], but 2-D pseudo-spectrum searching of MUSIC requires a large number of calculations. An algorithm based on the propagator method (PM) has been proposed to estimate the angles with two L-shaped arrays [22], but the algorithm needs an extra pair-matching method. Consequently, a pair-matching algorithm has been proposed by utilizing the diagonal factors of the cross-correlation matrix (CCM) [23]. To avoid pair-matching of azimuth and elevation angles, an estimation algorithm based on joint singular value decomposition (JSVD) has been proposed to obtain the auto-matched azimuth and elevation angles [24]. However, the beamforming-like operation of the algorithm requires heavy calculations. By employing the shift-invariance property of the CCM, an algorithm based on ESPRIT has been proposed [25], which performs eigenvalue decomposition (EVD) on improved CCMs to estimate the angles. Considering the properties of several CCMs with the same signal subspace, a subspace-based algorithm without singular value decomposition (SVD) or EVD has been proposed to obtain the azimuth and elevation angles [26]. Although this algorithm can acquire the signal subspace without SVD or EVD, the algorithm needs to minimize a cost function, which requires heavy calculations. In order to expand the array aperture and snapshots, the researchers have proposed an estimation algorithm based on the conjugate symmetry property of the uniform array's manifold matrix [27]. However, the expanded snapshots only benefit the azimuth or elevation angles. Nie, X. [28] has presented a 2-D DOA estimation algorithm for closely spaced sources, which performs SVD on two matrices constructed by CCMs to estimate the azimuth and elevation angles. Pair-matching is achieved by using the conjugate property of the non-zero eigenvalues of two matrices. To reduce the complexity of 2-D DOA estimation, a novel auto-matched PM for L-shaped array has been presented [29]. Unfortunately, the signal subspace is obtained directly from CCM, and thus the performance of the method is affected by the residual correlations of noise vectors. Liang, J. [30] has proposed a joint 2-D DOA estimation algorithm, which constructs a new angle with the elevation and azimuth angle and divides the arrays into two sub-arrays. The ESPRIT algorithm is then performed to obtain the azimuth and elevation angles without additional pair-matching.

All of the algorithms mentioned above are applied to the L-shaped uniform arrays, which confine the aperture and the maximum number of incident signal sources to be estimated. In contrast to the uniform arrays, sparse arrays with more degrees of freedom (DOF) and larger apertures have aroused the interest of scholars and researchers, such as minimum redundancy arrays [32], co-prime arrays [33], and nested arrays [34]. Co-prime arrays have been proposed for increasing the DOFs and improving the estimation resolution. However, the co-arrays of co-prime arrays are usually not a filled uniform, which may lead to phase ambiguity. To make the utmost of the co-arrays, an off-grid method has been proposed [35], which interpolates the additional sensors to the generated co-arrays to convert the non-uniform arrays into uniform linear arrays (ULA). Zhou, C.W. [36] has proposed an estimation method based on virtual arrays interpolation to reconstruct the covariance matrix of co-arrays, which can utilize all the derived co-arrays. Compared with other sparse arrays, nested arrays with N physical sensors can generate $O(N^2)$ DOF with a closed-form expression. Therefore, several algorithms based on sparse L-shaped arrays have been proposed for 2-D DOA estimation. Jian-F. G. [37] has proposed a joint 2-D DOA estimation algorithm for the L-shaped sparse arrays, that are composed of one uniform linear arrays and one sparse linear arrays (SLA). In order to obtain the auto-matching angles, the elevation angle is estimated first, and the azimuth angle is estimated using the estimated elevation angle. The resolution of the elevation angle estimated by the ULA is not as high as the azimuth angle estimated by the SLA. To make use of the advantages of nested arrays, an interlaced double precision 2-D DOA estimation algorithm using L-shaped nested arrays has been presented [38]. However, the azimuth or elevation angle is achieved based on solving a least-squares problem with another estimated angle. Influenced by noise perturbations, high-precision estimation can only be obtained for either the azimuth or elevation angle. By utilizing the spatial-temporal property of signals, Dong, Y.Y. [39] has proposed an algorithm for L-shaped nested arrays, which constructs several CCMs with

different time lags and performs a signal subspace joint diagonalization technique (SSJD) to estimate the azimuth and elevation angles simultaneously. However, the number of valid snapshots is reduced by dividing output dates into several frames, and the SSJD technique requires a loop iterative operation which requires heavy calculations.

In this paper, we propose an improved algorithm for L-shaped nested arrays in order to estimate the azimuth and elevation angles with a small number of snapshots. The array configuration is composed of two orthogonal nested arrays in the X-Y plane. Rather than using the auto-correlation matrix (ACM) of nested arrays, we calculate the CCM using the output of different sub-arrays, which removes the noise vector. Considering the perturbation of the sample cross-correlation matrix under a small number of snapshots, we reconstruct the CCM by using SVD. The sub-matrices of the reconstructed CCM are used to generate two large consecutive virtual arrays without an overlap element or residual noise. Since the equivalent received signals of virtual arrays behave like coherent sources, the output of the virtual arrays and their conjugate vectors are used to construct the full rank Toeplitz matrix as the equivalent covariance matrix. This process overcomes the rank deficiency of the virtual arrays. To reduce the computational complexity of the algorithm, a unitary ESPRIT method is performed on the equivalent covariance matrix to estimate the azimuth and elevation angles, separately. Since the azimuth and elevation angles are obtained, we achieve the pair-matched angles by deriving the equivalent source vector of the virtual arrays with the estimated angles. Our procedure can also estimate the power of the incident signals. Detailed analysis and numerical simulation are provided to demonstrate the lower complexity and better performance of the proposed algorithm.

The rest of the paper is organized as follows. The array configuration and signal model are illustrated in Section 2. A description of the proposed algorithm is introduced in Section 3. The simulation results and analysis are shown in Section 4, and the conclusion is presented in Section 5.

The following notations will be used throughout the paper. Vectors and matrices are denoted by using lower-case and capital bold letters, respectively. The superscripts $(\cdot)^*$, $(\cdot)^T$, $(\cdot)^+$, and $(\cdot)^H$ represent the conjugate, transpose, pseudo-inverse, and conjugate transpose operations, respectively. The symbol \odot denotes the Khatri-Rao product between two matrices, and $\text{vec}(\cdot)$ denotes the column vectorization operator, which turns a matrix into a column vector. The symbol $E(\cdot)$ represents the expectation operator. Finally, the notation Π_N denotes an $N \times N$ exchange matrix with ones on its anti-diagonal and zeros elsewhere.

2. Array Configuration and Signal Model

The array geometric configuration is illustrated in Figure 1; the L-shaped nested arrays are composed of two orthogonal nested arrays in the X-Y plane. Each nested arrays consists of two ULAs, where the inner ULA contains N sensors with inter-sensor spacing d and the outer ULA contains $N - 1$ sensors with inter-sensor spacing Nd , where $d = 1/2\lambda$, and λ is the wavelength of the incident signal. The sensor at the origin is seen as the reference sensor. All sensors are assumed to be identical, omnidirectional, and isotropic. Suppose that there are K far-field, uncorrelated, narrowband signals $s_i(t)$ ($i = 1, \dots, K$), impinging from distinct directions. Note that there are two ways to define the incident angles of sources. The first way is in terms of the azimuth angle θ_i , measured between the projection of i -th incident signal in the X-Y plane and the X-axis, and the elevation angle ϕ_i , measured between the i -th incident signal and the Z-axis. The second set of angles included the angle α_i , which is measured between the i -th incident signal and the X-axis, and the angle β_i , which is measured between the i -th incident signal and the Y-axis. It is easy to verify that a relationship between the angles (α_i, β_i) and the angles (θ_i, ϕ_i) exists, as shown in:

$$\begin{aligned}\cos \alpha_i &= \cos \theta_i \sin \phi_i \\ \cos \beta_i &= \sin \theta_i \sin \phi_i\end{aligned}\quad (1)$$

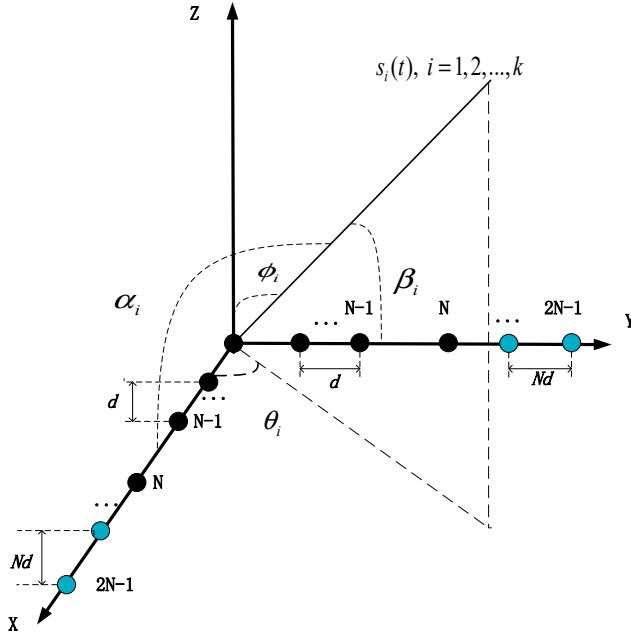


Figure 1. The geometry of the L-shaped nested array. Black dots indicate sensors of inner ULA with inter-spacing d , while blue dots show sensors of outer ULA with inter-spacing Nd .

The array output vectors at the sub-array along the X-axis and Y-axis are written in matrix form as follows:

$$\begin{aligned}\mathbf{X}(t) &= [x_1(t), x_2(t), \dots, x_{2N-1}(t)]^T \\ &= \mathbf{A}_x(\alpha)\mathbf{S}(t) + \mathbf{N}_x(t)\end{aligned}\quad (2)$$

$$\begin{aligned}\mathbf{Y}(t) &= [y_1(t), y_2(t), \dots, y_{2N-1}(t)]^T \\ &= \mathbf{A}_y(\beta)\mathbf{S}(t) + \mathbf{N}_y(t)\end{aligned}\quad (3)$$

where $\mathbf{A}_x(\alpha) = [a_x(\alpha_1), a_x(\alpha_2), \dots, a_x(\alpha_k)]$, $\mathbf{A}_y(\beta) = [a_y(\beta_1), a_y(\beta_2), \dots, a_y(\beta_k)]$ are the $(2N-1) \times K$ array manifold matrices of the X-axis and Y-axis, respectively, and the i -th manifold vectors $a_x(\alpha_i)$ and $a_y(\beta_i)$ are expressed as follows:

$$\begin{aligned}a_x(\alpha_i) &= [1, e^{-j\pi \cos \alpha_i}, \dots, e^{-j\pi(N-1) \cos \alpha_i}, \\ &\quad e^{-j\pi(2N-1) \cos \alpha_i}, \dots, e^{-j\pi(N^2-1) \cos \alpha_i}]^T\end{aligned}\quad (4)$$

$$\begin{aligned}a_y(\beta_i) &= [1, e^{-j\pi \cos \beta_i}, \dots, e^{-j\pi(N-1) \cos \beta_i}, \\ &\quad e^{-j\pi(2N-1) \cos \beta_i}, \dots, e^{-j\pi(N^2-1) \cos \beta_i}]^T\end{aligned}\quad (5)$$

Additionally, $\mathbf{S}(t) = [s_1(t), s_2(t), \dots, s_k(t)]^T$ is the source signal vector, $\mathbf{N}_x(t) = [n_{x1}(t), n_{x2}(t), \dots, n_{x2N-1}(t)]^T$ and $\mathbf{N}_y(t) = [n_{y1}(t), n_{y2}(t), \dots, n_{y2N-1}(t)]^T$ are the additive noise vectors. In this paper, we suppose the incident signals with variances $[\sigma_1^2, \sigma_2^2, \dots, \sigma_k^2]$ are uncorrelated to each other, the additive noise is temporally and spatially complex white Gaussian noise with zero mean and variance σ_n^2 , which are statistically independent of the signals. The number of incident signals k has already been estimated by a number detection technique. These properties of the source signal vector and additive noise vectors can be written as:

$$E[\mathbf{S}(t)\mathbf{S}^H(t)] = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_k^2) \quad (6)$$

$$E[\mathbf{S}(t)\mathbf{N}_i^H(t)] = \mathbf{0} \quad (7)$$

$$E[\mathbf{N}_i(t)\mathbf{N}_j^H(t)] = \begin{cases} \sigma_n^2\mathbf{I} & i = j \\ \mathbf{0} & i \neq j \end{cases}, \quad (8)$$

where \mathbf{I} and $\mathbf{0}$ indicate identity and null matrices, respectively.

3. The Proposed Method

3.1. Constructing the Cross-Correlation Matrix for a Small Number of Snapshots

We divide the nested arrays of the X-axis into two subarrays, as shown in Figure 2. Since the arrays along the X-axis and Y-axis have similar structures, the sub-array division of Y-axis is analogously divided. The nested array along the X-axis is composed of $2N-1$ sensors, there are N sensors in each sub-array, where the N -th sensor is shared by both sub-arrays. Sub-array 1 is a uniform linear array containing N sensors with spacing $d_1 = \lambda/2$, and sub-array 2 is a uniform linear arrays contained N sensors with spacing $d_2 = N\lambda/2$. The output vectors of sub-array 1 and sub-array 2 are given respectively by:

$$\begin{aligned} \mathbf{X}_1(t) &= [x_1(t), x_2(t), \dots, x_N(t)]^T \\ &= \mathbf{A}_{x1}(\alpha)\mathbf{S}(t) + \mathbf{N}_{x1}(t) \end{aligned} \quad (9)$$

$$\begin{aligned} \mathbf{X}_2(t) &= [x_N(t), x_{N+1}(t), \dots, x_{2N-1}(t)]^T \\ &= \mathbf{A}_{x2}(\alpha)\mathbf{S}(t) + \mathbf{N}_{x2}(t) \end{aligned} \quad (10)$$

where $\mathbf{N}_{x1}(t)$ and $\mathbf{N}_{x2}(t)$ are the noise vectors of sub-array 1 and sub-array 2, respectively. In addition, $\mathbf{A}_{x1}(\alpha) = [a_{x1}(\alpha_1), a_{x1}(\alpha_2), \dots, a_{x1}(\alpha_k)]$ and $\mathbf{A}_{x2}(\alpha) = [a_{x2}(\alpha_1), a_{x2}(\alpha_2), \dots, a_{x2}(\alpha_k)]$ denote the manifold matrices of sub-array 1 and sub-array 2, respectively, where the i -th column of these two manifold matrices are written as

$$a_{x1}(\alpha_i) = [1, e^{-j\pi \cos \alpha_i}, \dots, e^{-j(N-1)\pi \cos \alpha_i}]^T \quad (11)$$

$$a_{x2}(\alpha_i) = [e^{-j(N-1)\pi \cos \alpha_i}, \dots, e^{-j(N^2-1)\pi \cos \alpha_i}]^T \quad (12)$$

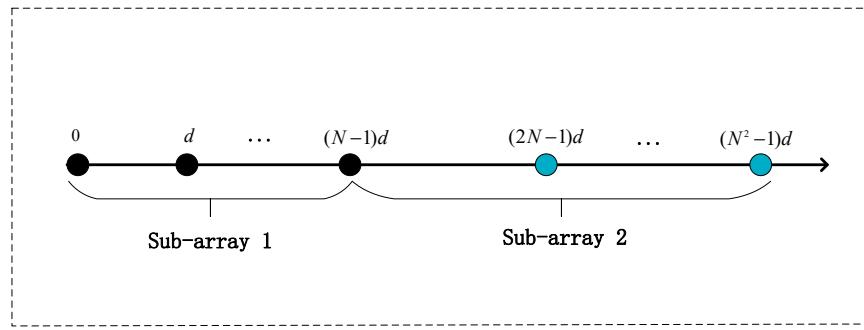


Figure 2. Subarray selection from the nested arrays of the X-axis. Black dots indicate sensors of sub-array 1 with inter-spacing d , while blue dots show sensors of sub-array 2 with inter-spacing Nd , the N -th sensor is shared by both sub-arrays, symbol denotes the location of sensor.

Similarly, sub-array 3 along the Y-axis is a uniform linear array containing N sensors with spacing $d_1 = \lambda/2$, and the sub-array 4 along the Y-axis is a uniform linear array containing N sensors with spacing $d_2 = N\lambda/2$, the output vector can be written as:

$$\begin{aligned} \mathbf{Y}_1(t) &= [y_1(t), y_2(t), \dots, y_N(t)]^T \\ &= \mathbf{A}_{y1}(\beta)\mathbf{S}(t) + \mathbf{N}_{y1}(t) \end{aligned} \quad (13)$$

$$\begin{aligned}\mathbf{Y}_2(t) &= [y_N(t), y_{N+1}(t), \dots, y_{2N-1}(t)]^T \\ &= \mathbf{A}_{y2}(\beta)\mathbf{S}(t) + \mathbf{N}_{y2}(t)\end{aligned}\quad (14)$$

where $\mathbf{N}_{y1}(t)$ and $\mathbf{N}_{y2}(t)$ are the noise vectors of sub-arrays 3 and 4, respectively, $\mathbf{A}_{y1}(\beta) = [a_{y1}(\beta_1), a_{y1}(\beta_2), \dots, a_{y1}(\beta_k)]$ and $\mathbf{A}_{y2}(\beta) = [a_{y2}(\beta_1), a_{y2}(\beta_2), \dots, a_{y2}(\beta_k)]$ are the manifold matrices of sub-array 3 and 4, respectively, where the i -th column of these two manifold matrices are written as:

$$a_{y1}(\beta_i) = [1, e^{-j\pi \cos \beta_i}, \dots, e^{-j\pi(N-1) \cos \beta_i}]^T \quad (15)$$

$$a_{y2}(\beta_i) = [e^{-j(N-1)\pi \cos \beta_i}, \dots, e^{-j(N^2-1)\pi \cos \beta_i}]^T. \quad (16)$$

By constructing the vectors with the outputs of the subarrays $\mathbf{Z}_1(t) = [\mathbf{X}_1(t), \mathbf{Y}_1(t)]^T$ and $\mathbf{Z}_2(t) = [\mathbf{X}_2(t), \mathbf{Y}_2(t)]^T$, the CCM of the two vectors can be computed as:

$$\mathbf{R}_z = E[\mathbf{Z}_1(t)\mathbf{Z}_2^H(t)] = \left[\begin{array}{c|c} \mathbf{X}_1(t) & \mathbf{X}_2(t) \\ \hline \mathbf{Y}_1(t) & \mathbf{Y}_2(t) \end{array} \right]^H = \left[\begin{array}{cc} \mathbf{R}_{x1x2} & \mathbf{R}_{x1y2} \\ \mathbf{R}_{y1x2} & \mathbf{R}_{y1y2} \end{array} \right]. \quad (17)$$

According to the Equation (17), the submatrices of \mathbf{R}_z can be expressed as:

$$\mathbf{R}_{x1x2} = E[\mathbf{X}_1(t)\mathbf{X}_2^H(t)] = \mathbf{A}_{x1}(\alpha)\mathbf{R}_s\mathbf{A}_{x2}^H(\alpha) + \sigma_n^2\mathbf{R}_{n0} \approx \mathbf{A}_{x1}(\alpha)\mathbf{R}_s\mathbf{A}_{x2}^H(\alpha) \quad (18)$$

$$\mathbf{R}_{x1y2} = E[\mathbf{X}_1(t)\mathbf{Y}_2^H(t)] = \mathbf{A}_{x1}(\alpha)\mathbf{R}_s\mathbf{A}_{y2}^H(\beta) \quad (19)$$

$$\mathbf{R}_{y1x2} = E[\mathbf{Y}_1(t)\mathbf{X}_2^H(t)] = \mathbf{A}_{y1}(\beta)\mathbf{R}_s\mathbf{A}_{x2}^H(\alpha) \quad (20)$$

$$\mathbf{R}_{y1y2} = E[\mathbf{Y}_1(t)\mathbf{Y}_2^H(t)] = \mathbf{A}_{y1}(\beta)\mathbf{R}_s\mathbf{A}_{y2}^H(\beta) + \sigma_n^2\mathbf{R}_{n0} \approx \mathbf{A}_{y1}(\beta)\mathbf{R}_s\mathbf{A}_{y2}^H(\beta), \quad (21)$$

where $\mathbf{R}_s = E[\mathbf{S}(t)\mathbf{S}^H(t)] = \text{diag}[\sigma_1^2, \sigma_2^2, \dots, \sigma_K^2]$, and σ_k^2 denotes the power of the k -th signal source. Since the noise vectors of different sensors are spatially independent and uncorrelated, the elements of \mathbf{R}_{n0} are all zeros except for the element $\mathbf{R}_{n0}(N, 1) = 1$. Considering that the matrix $\sigma_n^2\mathbf{R}_{n0}$ is an extremely sparse matrix, with only one non-zero element, it has little effect on the cross variance matrix \mathbf{R}_z , so we omit it here and in the coming analysis. The noise term $\sigma_n^2\mathbf{R}_{n0}$ is omitted only for simplifying analysis. The effect of omitting noise term $\sigma_n^2\mathbf{R}_{n0}$ on DOA estimation is examined in Section 4. From Equations (18)–(21), the noise vector is removed by the cross-correlation operation.

We perform SVD on \mathbf{R}_z , which can be expressed as:

$$\mathbf{R}_z = [\mathbf{U}_S \mathbf{U}_N] \left[\begin{array}{cc} \mathbf{\Lambda}_S & 0 \\ 0 & \mathbf{\Lambda}_N \end{array} \right] [\mathbf{V}_S \mathbf{V}_N]^H. \quad (22)$$

According to the second-order statistical properties of array signals, $\mathbf{\Lambda}_S$ and $\mathbf{\Lambda}_N$ are rectangular diagonal matrices with the K large singular values and other small singular values on the diagonal; \mathbf{U}_S , \mathbf{U}_N , \mathbf{V}_S , and \mathbf{V}_N are unitary matrices whose columns are left-singular vectors and right-singular vectors corresponding to $\mathbf{\Lambda}_S$ and $\mathbf{\Lambda}_N$. Considering the noise vectors are removed by the cross-correlation operation, $\mathbf{\Lambda}_N$ should be the null matrix.

In practice, the CCM \mathbf{R}_z is replaced by the sample CCM $\widehat{\mathbf{R}}_z$. Assuming that L snapshots are available, the sample CCM matrix $\widehat{\mathbf{R}}_z$ is:

$$\widehat{\mathbf{R}}_z = \frac{1}{L} \sum_{t=1}^L \mathbf{Z}_1(t)\mathbf{Z}_2^H(t). \quad (23)$$

Performing SVD on $\widehat{\mathbf{R}}_z$ can be thus expressed as:

$$\widehat{\mathbf{R}}_z = [\widehat{\mathbf{U}}_S \ \widehat{\mathbf{U}}_N] \begin{bmatrix} \widehat{\Lambda}_S & 0 \\ 0 & \widehat{\Lambda}_N \end{bmatrix} [\widehat{\mathbf{V}}_S \ \widehat{\mathbf{V}}_N]^H. \quad (24)$$

For small numbers of samples, the additive noise vectors does not satisfy the statistical assumptions in Equation (8). As mentioned in Ref. [40], if the number of samples is high, the singular values of the sample CCM will converge to the singular values of the CCM. On the contrary, the eigenvalues of the sample CCM will diverge in a large cluster. Therefore, $\widehat{\Lambda}_N$ may not equal the null matrix for small numbers of samples.

To reduce the perturbation of additive noise in the sample CCM $\widehat{\mathbf{R}}_z$ with small numbers of samples, we make the matrix $\widehat{\Lambda}_N = \mathbf{0}$ to reconstruct the modified CCM \mathbf{R}_{z1} :

$$\mathbf{R}_{z1} = [\widehat{\mathbf{U}}_S \ \widehat{\mathbf{U}}_N] \begin{bmatrix} \widehat{\Lambda}_S & 0 \\ 0 & \mathbf{0} \end{bmatrix} [\widehat{\mathbf{V}}_S \ \widehat{\mathbf{V}}_N]^H = \widehat{\mathbf{U}}_S \widehat{\Lambda}_S \widehat{\mathbf{V}}_S^H. \quad (25)$$

Comparing Equation (25) with Equation (24), it can be concluded that using the matrix \mathbf{R}_{z1} instead of $\widehat{\mathbf{R}}_z$ means that the sub-matrix $\widehat{\mathbf{U}}_N \widehat{\Lambda}_N \widehat{\mathbf{V}}_N^H$ is omitted.

3.2. Virtual Arrays Generation Based on the CCM for Nested Arrays

In this section, we first review the virtual arrays of the original nested arrays. As mentioned in Ref. [34], the nested arrays were proposed to obtain $O(M^2)$ DOF from M sensors by vectorizing the ACM of the received signals as follows:

$$z = \text{vec}(\mathbf{R}_{xx}) = (\mathbf{A}^* \odot \mathbf{A})p + \sigma_n^2 \vec{\mathbf{l}}_n, \quad (26)$$

where $\mathbf{R}_{xx} = E[\mathbf{X}(t)\mathbf{X}^H(t)] = \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \sigma_n^2 \mathbf{I}$ is the ACM of the received signals, $\mathbf{X}(t)$ is the output of the physical array, $p = [\sigma_1^2, \sigma_2^2, \dots, \sigma_k^2]^T$ denotes the equivalent signal vector, $\vec{\mathbf{l}}_n = [e_1^T \ e_2^T \ \dots \ e_N^T]^T$ is the noise vector of the virtual arrays, e_i is a column vector of all zeros except a 1 at the i -th position, and $\mathbf{A}^* \odot \mathbf{A}$ can be seen as the manifold matrix of the virtual arrays.

The locations of virtual arrays can be expressed by the set $D = \left\{ \vec{d}_i - \vec{d}_j, 1 \leq i, j \leq N \right\}$, where \vec{d}_i denote the position vector of the i -th sensor. According to the analysis in Ref. [33], the set D contains the self-differences set D_s and the cross-differences set D_c , defined as:

$$D_s = \{(N_1 + 1)(k_1 - k_2)\} \cup \{(l_1 - l_2)\}, 1 < k_1, k_2 < N_2, 1 < l_1, l_2 < N_1 \quad (27)$$

$$D_c = \{(N_1 + 1)k - l\} \cup \{l - (N_1 + 1)k\}, 1 < k < N_2, 1 < l < N_1, \quad (28)$$

where l denotes the position vector of first level subarray with spacing $d_1 = \lambda/2$, k denotes the position vector of second level sub-array with spacing $d_2 = (N_1 + 1)\lambda/2$, N_1 is the number of sensors in the first level subarray, and N_2 is the number of sensors in the second level subarray. From Equations (27) and (28), the cross-differences set D_c contributes the majority of sensor locations, the self-differences set D_s contributes the remaining sensor locations, which contain some overlapping ones. In addition, the output of the virtual arrays contains noise $\vec{\mathbf{l}}_n$, which need an extra operation to remove. In order to obtain the virtual arrays without the overlapping element and noise, we propose an improved L-shaped nested arrays based on CCM, in order to take full advantage of the cross-differences property.

We divide the modified CCM \mathbf{R}_{z1} into four sub-matrices as shown in Equation (17), and then reverse the matrices \mathbf{R}_{x1x2} and \mathbf{R}_{y1y2} according to the columns, which can be expressed as:

$$\begin{aligned}\mathbf{R}_x &= \prod_N \mathbf{R}_{x1x2} \\ &= \sum_{i=1}^k \sigma_i^2 \begin{bmatrix} 1 & e^{jN\pi \cos \alpha_i} & \dots & e^{j(N^2-N)\pi \cos \alpha_i} \\ e^{j\pi \cos \alpha_i} & e^{j(N+1)\pi \cos \alpha_i} & \dots & e^{j(N^2-N+1)\pi \cos \alpha_i} \\ \vdots & \vdots & \vdots & \vdots \\ e^{j(N-1)\pi \cos \alpha_i} & e^{j(2N-1)\pi \cos \alpha_i} & \dots & e^{j(N^2-1)\pi \cos \alpha_i} \end{bmatrix} \quad (29)\end{aligned}$$

$$\begin{aligned}\mathbf{R}_y &= \prod_N \mathbf{R}_{y1y2} \\ &= \sum_{i=1}^k \sigma_i^2 \begin{bmatrix} 1 & e^{jN\pi \cos \beta_i} & \dots & e^{j(N^2-N)\pi \cos \beta_i} \\ e^{j\pi \cos \beta_i} & e^{j(N+1)\pi \cos \beta_i} & \dots & e^{j(N^2-N+1)\pi \cos \beta_i} \\ \vdots & \vdots & \vdots & \vdots \\ e^{j(N-1)\pi \cos \beta_i} & e^{j(2N-1)\pi \cos \beta_i} & \dots & e^{j(N^2-1)\pi \cos \beta_i} \end{bmatrix} \quad (30)\end{aligned}$$

We utilize the matrices \mathbf{R}_x and \mathbf{R}_y to calculate the following vectors:

$$\begin{aligned}r_x &= \text{vec}(\mathbf{R}_x) = \text{vec}(\prod_N \mathbf{A}_{x1}(\alpha) \mathbf{R}_s \mathbf{A}_{x2}^H(\alpha)) \\ &= (\mathbf{A}_{x2}^*(\alpha) \odot \mathbf{A}_{x1z}(\alpha)) p \\ &= \overline{\mathbf{A}_x}(\alpha) p \quad (31)\end{aligned}$$

$$\begin{aligned}r_y &= \text{vec}(\mathbf{R}_y) = \text{vec}(\prod_N \mathbf{A}_{y1}(\beta) \mathbf{R}_s \mathbf{A}_{y2}^H(\beta)) \\ &= (\mathbf{A}_{y2}^*(\beta) \odot \mathbf{A}_{y1z}(\beta)) p \\ &= \overline{\mathbf{A}_y}(\beta) p \quad (32)\end{aligned}$$

where $\mathbf{A}_{x1z}(\alpha) = \prod_N \mathbf{A}_{x1}(\alpha)$, $\mathbf{A}_{y1z}(\alpha) = \prod_N \mathbf{A}_{y1}(\beta)$, and the equivalent signal vector is expressed by $p = [\sigma_1^2, \sigma_2^2, \dots, \sigma_k^2]^T$. Additionally, r_x can be seen as the output of the virtual arrays along the X-axis, whose manifold is given by $\overline{\mathbf{A}_x} = [\overline{a}_x(\alpha_1), \overline{a}_x(\alpha_2), \dots, \overline{a}_x(\alpha_k)]$; r_y can be seen as the output of the virtual arrays along the Y-axis, whose manifold is given by $\overline{\mathbf{A}_y} = [\overline{a}_y(\beta_1), \overline{a}_y(\beta_2), \dots, \overline{a}_y(\beta_k)]$; and the i -th column of matrix $\overline{\mathbf{A}_x}$ and $\overline{\mathbf{A}_y}$ are respectively written as:

$$\overline{a}_x(\alpha_i) = [1, e^{j\pi \cos \alpha_i}, \dots, e^{j\pi(N^2-1) \cos \alpha_i}] \in \mathbb{C}^{N^2 \times 1} \quad (33)$$

$$\overline{a}_y(\beta_i) = [1, e^{j\pi \cos \beta_i}, \dots, e^{j\pi(N^2-1) \cos \beta_i}] \in \mathbb{C}^{N^2 \times 1}. \quad (34)$$

Defining $r_x = [\overline{x}_1, \overline{x}_2, \dots, \overline{x}_{N^2}]^T$ and $r_y = [\overline{y}_1, \overline{y}_2, \dots, \overline{y}_{N^2}]^T$ as the outputs of the virtual arrays, where \overline{x}_i and \overline{y}_i denote the output of i -th virtual array along the X-axis and Y-axis, respectively. The conjugate vectors of r_x and r_y are expressed as follows:

$$\begin{aligned}r_x^* &= [\overline{x}_1^*, \overline{x}_2^*, \dots, \overline{x}_{N^2}^*]^T \\ &= (\overline{\mathbf{A}_x}(\alpha)p)^* \\ &= \overline{\mathbf{A}_x}^*(\alpha)p \quad (35)\end{aligned}$$

$$\begin{aligned}r_y^* &= [\overline{y}_1^*, \overline{y}_2^*, \dots, \overline{y}_{N^2}^*]^T \\ &= (\overline{\mathbf{A}_y}(\beta)p)^* \\ &= \overline{\mathbf{A}_y}^*(\beta)p \quad (36)\end{aligned}$$

where \overline{x}_i^* and \overline{y}_i^* denote the conjugate vector of the i -th virtual array's output along the X-axis and Y-axis, $p^* = p = [\sigma_1^2, \sigma_2^2, \dots, \sigma_k^2]^T$, $\overline{\mathbf{A}}_x^* = [\overline{a}_x^*(\alpha_1), \overline{a}_x^*(\alpha_2), \dots, \overline{a}_x^*(\alpha_k)]$, and $\overline{\mathbf{A}}_y^* = [\overline{a}_y^*(\beta_1), \overline{a}_y^*(\beta_2), \dots, \overline{a}_y^*(\beta_k)]$. The i -th column of the matrices of $\overline{\mathbf{A}}_x^*$ and $\overline{\mathbf{A}}_y^*$ are written as:

$$\overline{a}_x^*(\alpha_i) = [1, e^{-j\pi \cos \alpha_i}, \dots, e^{-j(N^2-1)\pi \cos \alpha_i}] \in \mathbb{C}^{N^2 \times 1} \quad (37)$$

$$\overline{a}_y^*(\beta_i) = [1, e^{-j\pi \cos \beta_i}, \dots, e^{-j\pi(N^2-1) \cos \beta_i}] \in \mathbb{C}^{N^2 \times 1}. \quad (38)$$

According to the Equations (35) and (36), r_x^* and r_y^* can be seen as the outputs of virtual arrays whose manifold are given by $\overline{\mathbf{A}}_x^*$ and $\overline{\mathbf{A}}_y^*$, respectively. Defining $\overline{\mathbf{X}} = [\overline{x}_1^*, \dots, \overline{x}_2^*, \overline{x}_1, \dots, \overline{x}_{N^2}]^T$, which behaves like the output of a uniform linear array along the X-axis containing $2N^2 - 1$ elements with spacing $d_1 = \lambda/2$, and $\overline{\mathbf{Y}} = [\overline{y}_1^*, \dots, \overline{y}_2^*, \overline{y}_1, \dots, \overline{y}_{N^2}]^T$, which behaves like the output of a uniform linear array along the Y-axis containing $2N^2 - 1$ elements with spacing $d_1 = \lambda/2$. Thus, we obtain the virtual arrays of $4N^2 - 2$ elements in the X-Y plane using only $4N - 2$ physical sensors, which dramatically increases the DOF.

3.3. Improved Unitary ESPRIT Algorithm for Received Signals of Virtual Arrays

According to the outputs of the virtual arrays in Equations (31), (32), (35), and (36), the equivalent signal vector p behaves like coherent signal sources that lead to a rank deficiency of the covariance matrix. Hence, the conventional high-resolution DOA estimation algorithm such as MUSIC or ESPRIT can't be directly used for the virtual arrays. In this subsection, the equivalent covariance matrix is constructed to estimate the DOA of the virtual arrays.

Since the covariance matrix is a Toeplitz matrix, we can use $\overline{\mathbf{X}}$ to construct the equivalent covariance matrix \mathbf{R}_X :

$$\mathbf{R}_X = \begin{bmatrix} \overline{x}_1 & \overline{x}_2^* & \overline{x}_3^* & \dots & \overline{x}_{N^2-1}^* \\ \overline{x}_2 & \overline{x}_1 & \overline{x}_2^* & \dots & \overline{x}_{N^2-1}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \overline{x}_{N^2} & \overline{x}_{N^2-1} & \overline{x}_{N^2-2} & \dots & \overline{x}_1 \end{bmatrix} = \overline{\mathbf{A}}_x \begin{pmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_k^2 \end{pmatrix} \overline{\mathbf{A}}_x^H. \quad (39)$$

From Equation (33), $\overline{\mathbf{A}}_x$ is Vandermonde matrix that satisfies the property of rotational invariance. After performing the EVD of the matrix \mathbf{R}_X , the rotational invariance property can be expressed as:

$$\mathbf{J}_1 \mathbf{U}_s \Phi_x = \mathbf{J}_2 \mathbf{U}_s, \quad (40)$$

where \mathbf{U}_s is the matrix composed of eigenvector corresponding to the K largest eigenvalues, $\mathbf{J}_1 = [I_{N^2-1}, \mathbf{0}_{(N^2-1) \times 1}]$, $\mathbf{J}_2 = [\mathbf{0}_{(N^2-1) \times 1}, I_{N^2-1}]$, and $\Phi_x = \text{diag}[e^{j\pi \cos \alpha_1}, e^{j\pi \cos \alpha_2}, \dots, e^{j\pi \cos \alpha_k}]$.

The ESPRIT algorithm needs to perform SVD or EVD on the complex covariance matrix which means leads to a high computational burden. To reduce the computational complexity of the algorithm, the unitary transformation is used to transform the complex covariance matrix into a real-valued matrix [41]. We construct the Centro-Hermitian matrix \mathbf{R}_{X1} :

$$\mathbf{R}_{X1} = \frac{1}{2} (\mathbf{R}_X + \prod_{N^2} \mathbf{R}_X^* \prod_{N^2}). \quad (41)$$

Additionally, we define the unitary matrix:

$$\mathbf{Q}_n = \begin{cases} \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_m & j\mathbf{I}_m \\ \Pi_m & -j\Pi_m \end{bmatrix} & n=2m \\ \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{(m \times 1)} & j\mathbf{I}_m \\ \mathbf{0}_{(1 \times m)} & \sqrt{2} & \mathbf{0}_{(1 \times m)} \\ \Pi_m & \mathbf{0}_{(m \times 1)} & -j\Pi_m \end{bmatrix} & n=2m+1 \end{cases}. \quad (42)$$

We use the unitary matrix \mathbf{Q}_n to transform the complex matrix \mathbf{R}_{X1} into real matrix \mathbf{R}_{X2} via:

$$\begin{aligned}\mathbf{R}_{X2} &= \frac{1}{2} \mathbf{Q}_{N^2}^H (\mathbf{R}_X + \Pi_{N^2} \mathbf{R}_X^* \Pi_{N^2}) \mathbf{Q}_{N^2} \\ &= \frac{1}{2} (\mathbf{Q}_{N^2}^H \mathbf{R}_X \mathbf{Q}_{N^2} + \mathbf{Q}_{N^2}^H \Pi_{N^2} \mathbf{R}_X^* \Pi_{N^2} \mathbf{Q}_{N^2}) \\ &= \frac{1}{2} (\mathbf{Q}_{N^2}^H \mathbf{R}_X \mathbf{Q}_{N^2} + \mathbf{Q}_{N^2}^T \mathbf{R}_X^* \mathbf{Q}_{N^2}^*) \\ &= \text{Re}(\mathbf{Q}_{N^2}^H \mathbf{R}_X \mathbf{Q}_{N^2})\end{aligned}\quad (43)$$

After the unitary transformation, the complex rotational invariance property of Equation (40) becomes the real-valued rotation; invariance property as follows:

$$\mathbf{K}_1 \mathbf{E}_{sx} \Psi_x = \mathbf{K}_2 \mathbf{E}_{sx}, \quad (44)$$

where $\mathbf{K}_1 = \mathbf{Q}_{N^2-1}^H (\mathbf{J}_1 + \mathbf{J}_2) \mathbf{Q}_{N^2}$, $\mathbf{K}_2 = \mathbf{Q}_{N^2-1}^H j(\mathbf{J}_1 - \mathbf{J}_2) \mathbf{Q}_{N^2}$, and $\Psi_x = \text{diag}[\tan(\pi \cos \alpha_1/2), \dots, \tan(\pi \cos \alpha_k/2)]$.

From the Equation (44), the Ψ_x can be obtained by the least square method:

$$\Psi_x = (\mathbf{K}_1 \mathbf{E}_{sx})^+ \mathbf{K}_2 \mathbf{E}_{sx}. \quad (45)$$

By performing EVD on Ψ_x , the angle $\widehat{\alpha}_i$ can be obtained from the i -th eigenvalue a_i of Ψ_x as follow:

$$\widehat{\alpha}_i = \arccos(2 \arctan(a_i)/\pi). \quad (46)$$

Similarly, we use \mathbf{r}_y and \mathbf{r}_y^* to construct the equivalent covariance matrix \mathbf{R}_Y :

$$\mathbf{R}_Y = \begin{bmatrix} \overline{y_1} & \overline{y_2}^* & \overline{y_3}^* & \cdots & \overline{y_{N^2}}^* \\ \overline{y_2} & \overline{y_1} & \overline{y_2}^* & \cdots & \overline{y_{N^2-1}}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \overline{y_{N^2}} & \overline{y_{N^2-1}} & \overline{y_{N^2-2}} & \cdots & \overline{y_1} \end{bmatrix} = \overline{\mathbf{A}_y} \begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k^2 \end{pmatrix} \overline{\mathbf{A}_y}^H. \quad (47)$$

We then use the unitary matrix \mathbf{Q}_n to transform \mathbf{R}_Y into real matrix \mathbf{R}_{Y1} as follow:

$$\begin{aligned}\mathbf{R}_{Y1} &= \frac{1}{2} \mathbf{Q}_{N^2}^H (\mathbf{R}_Y + \Pi_{N^2} \mathbf{R}_Y^* \Pi_{N^2}) \mathbf{Q}_{N^2} \\ &= \text{Re}(\mathbf{Q}_{N^2}^H \mathbf{R}_Y \mathbf{Q}_{N^2})\end{aligned}\quad (48)$$

After the unitary transformation, the real-valued rotation invariant property is written as:

$$\mathbf{K}_1 \mathbf{E}_{sy} \Psi_y = \mathbf{K}_2 \mathbf{E}_{sy} \quad (49)$$

where $\Psi_y = \text{diag}[\tan(\pi \cos \beta_1/2), \dots, \tan(\pi \cos \beta_k/2)]$.

Then Ψ_y can be obtained by the least square method. After performing EVD on Ψ_y , the angle $\widehat{\beta}_i$ can be obtained from the i -th eigenvalue b_i of the matrix Ψ_y as follows:

$$\Psi_y = (\mathbf{K}_1 \mathbf{E}_{sy})^+ \mathbf{K}_2 \mathbf{E}_{sy} \quad (50)$$

$$\widehat{\beta}_i = \arccos(2 \arctan(b_i)/\pi). \quad (51)$$

3.4. Pair Matching

Since the angles $\widehat{\alpha}$ and $\widehat{\beta}$ are separately obtained by 1-D DOA estimation. This may lead to a mismatch between angle $\widehat{\alpha}$ and angle $\widehat{\beta}$ with more than one incident signal. From Equations (31) and (32), the equivalent signal vector p of different virtual arrays is the same vector. The manifold matrix of

virtual arrays $\overline{\mathbf{A}_x}(\widehat{\alpha})$ can be calculated with the estimated angle $\widehat{\alpha}$. From Equation (31), the equivalent signal vector p_1 can be derived by the least square method as follows:

$$\widehat{p}_1 = \operatorname{argmin}_p \|r_x - \overline{\mathbf{A}_x}(\widehat{\alpha})\widehat{p}_1\|^2 \quad (52)$$

$$\widehat{p}_1 = \left[\overline{\mathbf{A}_x}^H(\widehat{\alpha}) \overline{\mathbf{A}_x}(\widehat{\alpha}) \right]^{-1} \overline{\mathbf{A}_x}^H(\widehat{\alpha}) r_x \quad (53)$$

Similarly, the manifold matirx $\overline{\mathbf{A}_y}(\beta)$ can be calculated with the estimated angle $\widehat{\beta}$. From Equation (32), the equivalent signal vector p_2 can be derived via:

$$\widehat{p}_2 = \operatorname{argmin}_p \|r_y - \overline{\mathbf{A}_y}(\widehat{\beta})\widehat{p}_2\|^2 \quad (54)$$

$$\widehat{p}_2 = \left[\overline{\mathbf{A}_y}^H(\widehat{\beta}) \overline{\mathbf{A}_y}(\widehat{\beta}) \right]^{-1} \overline{\mathbf{A}_y}^H(\widehat{\beta}) r_y. \quad (55)$$

The elements of the equivalent signal vector $p = [\sigma_1^2, \sigma_2^2, \dots, \sigma_k^2]^T$ denote the powers of the incident signals. Due to the different power of incident signals, the pair matching problem can be done by sorting the elements of \widehat{p}_1 and \widehat{p}_2 :

$$\widehat{\Pi} = \operatorname{argmin}_{\Pi} \|\widehat{p}_1 - \Pi \widehat{p}_2\|, \quad (56)$$

where the $\widehat{\Pi}$ is the sorting matrix. Then the result of pair matching is the following:

$$\widehat{\alpha} = \widehat{\Pi} \widehat{\beta}. \quad (57)$$

According to Equations (53) and (55), the powers of incident signals have also been estimated.

3.5. Algorithm Implementation and Complexity Analysis

The proposed algorithm can be summarized as follows:

Step 1: Divide the nested arrays of into four subarrays. Construct the vectors $\mathbf{Z}_1(t)$ and $\mathbf{Z}_2(t)$ with the outputs of the subarrays. Then calculate the CCM $\mathbf{R}_z = E[\mathbf{Z}_1(t)\mathbf{Z}_2^H(t)]$.

Step 2: Perform SVD on the matrix \mathbf{R}_z to reconstruct the modified CCM \mathbf{R}_{z1} via Equation (25).

Step 3: Divide the modified CCM \mathbf{R}_{z1} into four sub-matrices as in Equation (17), then use the submatrices \mathbf{R}_{x1x2} and \mathbf{R}_{y1y2} to obtain the output of the virtual arrays via Equations (29)–(32).

Step 4: Construct the equivalent covariance matrices \mathbf{R}_X and \mathbf{R}_Y with the output of the virtual arrays via Equations (39) and (47).

Step 5: Utilize the unitary matrix to transform the covariance matrix into a real-valued covariance matrix. Then conduct the ESPRIT algorithm on the real-valued covariance matrix to estimate the angles $\widehat{\alpha}$ and $\widehat{\beta}$.

Step 6: Derive the equivalent source signal vectors p_1 and p_2 with the estimated angles $\widehat{\alpha}$ and $\widehat{\beta}$ via Equations (52) and (54), respectively. Derive the sorting matrix $\widehat{\Pi}$, and pair the angles $\widehat{\alpha}$ and $\widehat{\beta}$ by using the sorting matrix $\widehat{\Pi}$.

As for the complexity of algorithm, the main computation of algorithm contains the reconstructing CCM, SVD operation, pseudo-inverse operation and the least square method. Considering that the virtual arrays extend the dimension of the equivalent covariance matrix, which need a lot of calculations, the unitary transformation is performed on equivalent covariance matrix to reduce the computational complexity of the proposed algorithm. The main computational complexity of our proposed algorithm is $O[(M+1)^2L + (M+1)^3 + (M+1)^6/128 + 5(M+1)^2k^2/8 + (M+1)^2k/8 + k^3/2]$, where M denotes the number of sensors, L denotes the number of snapshots, k denotes the number of incident

signals. The algorithm of joint singular value decomposition (JSVD) [24] involves a beamforming-like spectral search. By defining the search step of 0.1 degree, the computational complexity of JSVD is about $O[M^2L + 8M^3 + 1800M^2]$. On the other hand, the method of cross-correlation matrices propagator method (CCMs-PM) [29], which obtain the DOAs without EVD or SVD, cost approximately $O[M^2L + 2k^3 + (7M - 4)k^2 + k(M - 1)(2M - k)]$. The methods of JSVD and CCMs-PM are applied on L-shaped uniform arrays. The method of signal subspace joint diagonalization (SSJD) [39], performed on nested arrays, requires a complexity of $O[3M^2L(2N - 1) + 4((M^2 - 1)/2 + M^2 + M)^2(2N - 1) + 8((M^2 - 1)/2 + M^2 + M)^3 + 10k^3]$, where the N is frame number of the method. For the sake of clarity, the main complexity of the proposed algorithm, the JSVD algorithm, the SSJD algorithm, and the CCMs-PM algorithm are listed in Table 1.

Table 1. Comparison of the computational complexity for different algorithms.

Algorithm	Main Computational Complexity
Proposed algorithm	$O[(M + 1)^2L + (M + 1)^3 + (M + 1)^6/128 + 5(M + 1)^2k^2/8 + (M + 1)^2k/8 + k^3/2]$ $O[M^2L + 8M^3 + 1800M^2]$
JSVD	
SSJD	$O[3M^2L(2N - 1) + 4((M^2 - 1)/2 + M^2 + M)^2(2N - 1) + 8((M^2 - 1)/2 + M^2 + M)^3 + 10k^3]$
CCMs-PM	$O[M^2L + 2k^3 + (7M - 4)k^2 + k(M - 1)(2M - k)]$

Figure 3 shows the complexity comparison of algorithms versus the number of sensors, where the number of snapshots L is 100, the number of incident signals k is 3. Figure 4 shows the complexity comparison of algorithms versus the number of snapshots, where the number of sensors M is 9, the number of incident signals k is 3. From Figures 3 and 4, the complexity of the proposed algorithm is much lower than the JSVD and SSJD algorithms. The low computational complexity is obtained from the unitary transformation of the complex matrix into a real-valued matrix. Although the complexity of the proposed algorithm is a bit higher than the CCMs-PM, the computational complexity of the proposed algorithm is basically on the same order as the complexity of CCMs-PM.

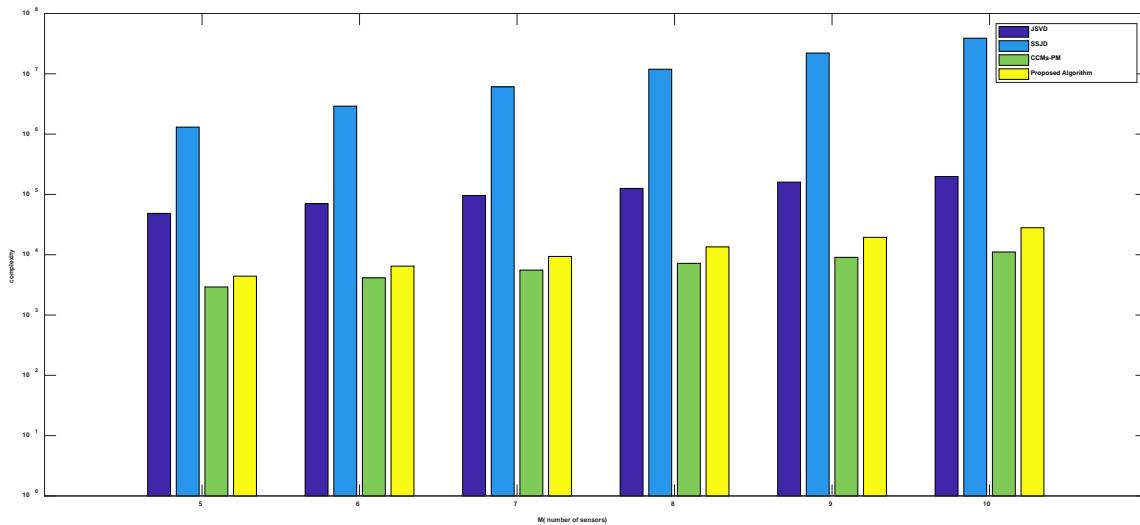


Figure 3. Comparison of complexity versus the number of sensors.

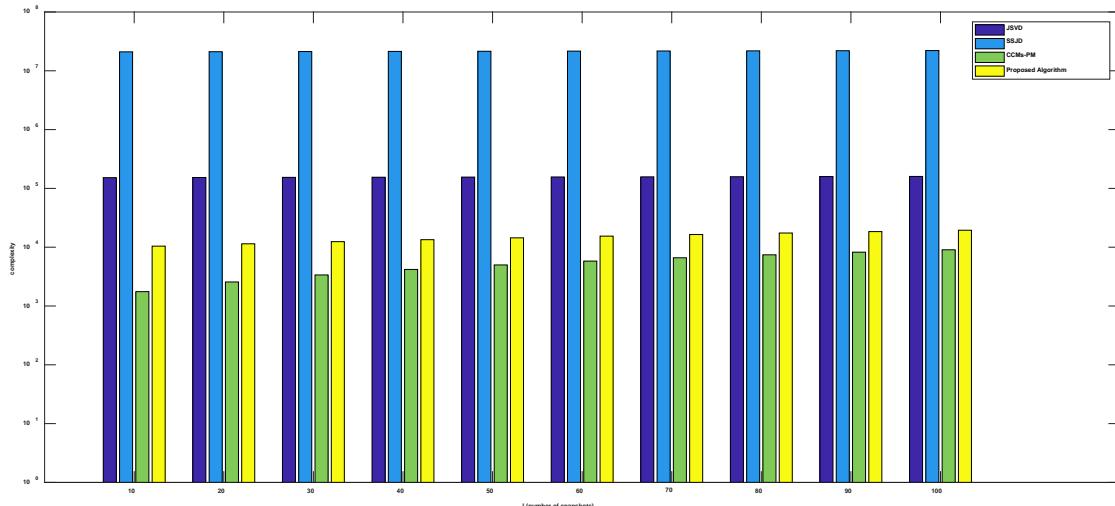


Figure 4. Comparison of complexity versus the number of snapshots.

4. Simulation Results and Performance Analysis

In this section, several simulation experiments are conducted to verify the performance of the proposed algorithm. Consider L-shaped nested arrays consisting of two orthogonal nested arrays, where each nested array contains nine sensors, of which the inner ULA contains five sensors with an inter-sensor spacing of $\lambda/2$ and the outer ULA contains four sensors with an inter-sensor spacing $5\lambda/2$, where λ is the wavelength of the incident waves. Consider that three signals are impinging from the locations $(\alpha_1, \beta_1) = (25^\circ, 30^\circ)$, $(\alpha_2, \beta_2) = (35^\circ, 40^\circ)$, and $(\alpha_3, \beta_3) = (45^\circ, 50^\circ)$. The power of signals are set as $\sigma_1^2 = 3$, $\sigma_2^2 = 4$, and $\sigma_3^2 = 5$. The input signal-to-noise-ratio (SNR) is defined as $SNR = 10 \log_{10}(\sigma_s^2 / \sigma_n^2)$, where σ_s^2 and σ_n^2 denote the power of signal and noise, respectively.

Two criteria for assessing the performance of the DOA estimation algorithm are the probability of resolution and the root mean square error (RMSE). The probability of resolution is defined as the probability that the angle difference between the estimated angle and the real angle is less than half of the beam width among several Monte-Carlo experiments, which can be expressed as follows:

$$P = P\left(\max\left\{\left|\widehat{\alpha}_{i,j} - \alpha_j\right|, \left|\widehat{\beta}_{i,j} - \beta_j\right|\right\} \leq \frac{BW}{2}\right) \quad (58)$$

where BW denotes the beam width.

The definition of RMSE is as follows:

$$RMSE = \sqrt{\frac{1}{nk} \sum_{i=1}^n \sum_{j=1}^k ((\widehat{\alpha}_{i,j} - \alpha_j)^2 + (\widehat{\beta}_{i,j} - \beta_j)^2)} \quad (59)$$

where $\widehat{\alpha}_{i,j}$ and $\widehat{\beta}_{i,j}$ are the estimated angles of j -th incident signal for i -th Monte-Carlo experiment. Likewise, α_j and β_j are the true angles of j -th incident signal. Additionally, n is the number of Monte-Carlo trials, and k denotes the number of incident signals.

In first experiment, we examine the effect of the noise term $\sigma_n^2 \mathbf{R}_{n0}$ in Equations (18) and (21) on DOA estimation. We compare the RMSE of the proposed algorithm with the denoising algorithm, which removed the noise term $\sigma_n^2 \mathbf{R}_{n0}$ from the CCM and the other steps are the same as the proposed algorithm. The RMSE of the proposed algorithm and denoising algorithm versus SNR under 1000 Monte-Carlo simulations is shown in Figure 5, where the number of snapshots L is 10, 20 and 100, respectively. From Figure 5, the RMSE of the proposed algorithm and denoising algorithm under different numbers of snapshots are almost overlapping in all cases, which verifies the rationality of omitting the noise term in Equations (18) and (21).

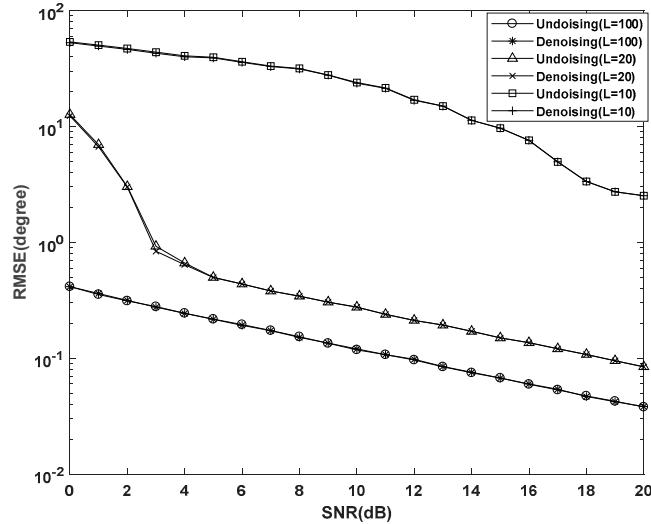


Figure 5. Effects of omitting the noise on root mean square error (RMSE) performance.

In the second experiment, we examine the DOA estimation performance of the proposed algorithm versus the SNR in terms of the root mean square error (RMSE). We also compare the proposed algorithm with the CRB, and the JSVD [24], CCMs-PM [29], SSJD algorithms [39]. The JSVD and CCMs-PM algorithms are performed on L-shaped uniform arrays, whose sensor spacing is $\lambda/2$. The SSJD algorithm is performed on the L-shaped nested arrays. The CRB of L-shaped nested arrays is derived in Appendix A. Figures 6 and 7 show the RMSE and the probability of resolution of the different algorithms versus the SNR for 1000 Monte-Carlo simulations, where the number of snapshots L is 100.

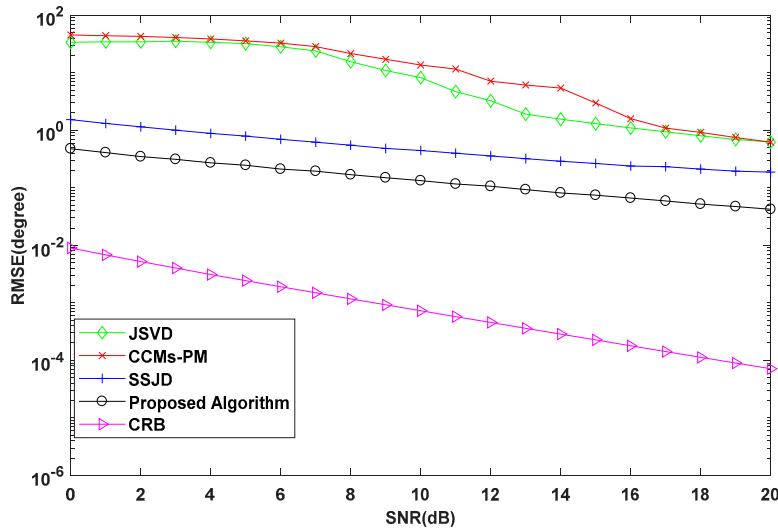


Figure 6. RMSE performance versus signal-to-noise-ratio (SNR) for simulations with 100 snapshots.

As shown in Figure 6, the RMSE of the proposed algorithm is lower than other compared algorithms for all SNRs. From Figure 7, we see the probability of resolution for all algorithms is improved with the increase of SNR, and the probability of resolution for the proposed algorithm is constant equal to 100%, which is obviously better than other compared algorithms. These results confirm that our proposed algorithm outperforms the JSVD algorithm and CCMs-PM algorithm due to the larger DOF obtained from nested arrays. Moreover, the computational complexity of the proposed algorithm is not higher than the JSVD algorithm and CCMs-PM algorithm. Compared with the SSJD algorithm, our algorithm also has better angle estimation performance by utilizing the reconstructed CCM instead of the ACM, which removes the noise vector.

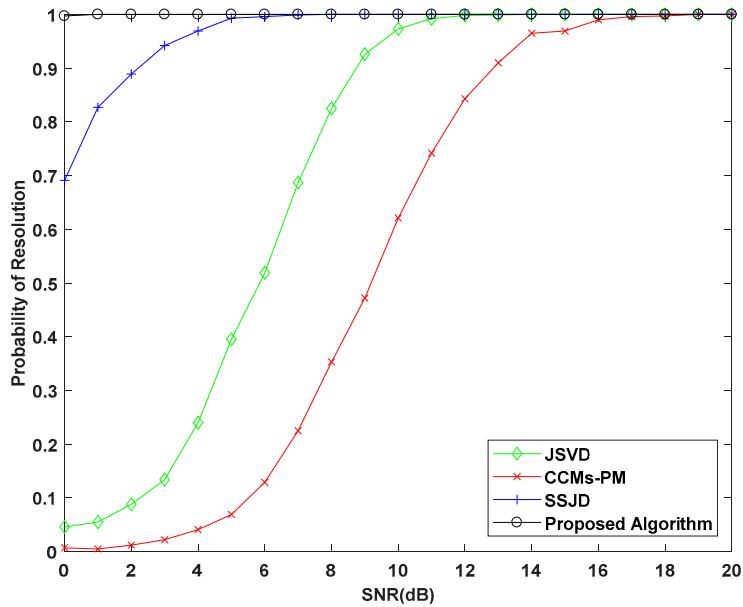


Figure 7. Probability of Resolution versus SNR for simulation with 100 snapshots.

In the third experiment, we evaluate the DOA estimation performance of the proposed algorithm versus the number of snapshots. The SNR is set to 10 dB and 20 dB. The number of snapshots varies from 20 to 200. Similarly, we compare the performance of the proposed algorithm with CRB, and the JSVD, CCMS-PM, and SSJD algorithms with 1000 Monte Carlo trials.

As shown in Figures 8 and 9, the RMSE and the probability of resolution depicts that the performance of the proposed algorithm still outperforms the contrasting algorithms for the studied numbers of snapshots. Since the JSVD and CCMS-PM are vulnerable to noise, the performance of these algorithms seriously deteriorates under a low SNR environment. From Figures 8 and 9, when SNR = 0 dB and the number of snapshots is less than 60, the RMSE of the proposed algorithm decreases rapidly with the increasing of snapshots, which is closed to the RMSE of SSJD. When the SNR is 0 dB and the number of snapshots is more than 60, the RMSE of the proposed algorithm decreases smoothly with the increasing number of snapshots, and the probability of resolution of the proposed algorithm is above 90%. When SNR = 10 dB, the RMSE of the proposed algorithm decrease smoothly with the increasing number of snapshots, and when the number of snapshots is more than 60, the proposed algorithm's probability of resolution is almost 100%. Figures 8 and 9 indicate that our proposed algorithm has a better performance under the conditions of low SNR and a small number of snapshots.

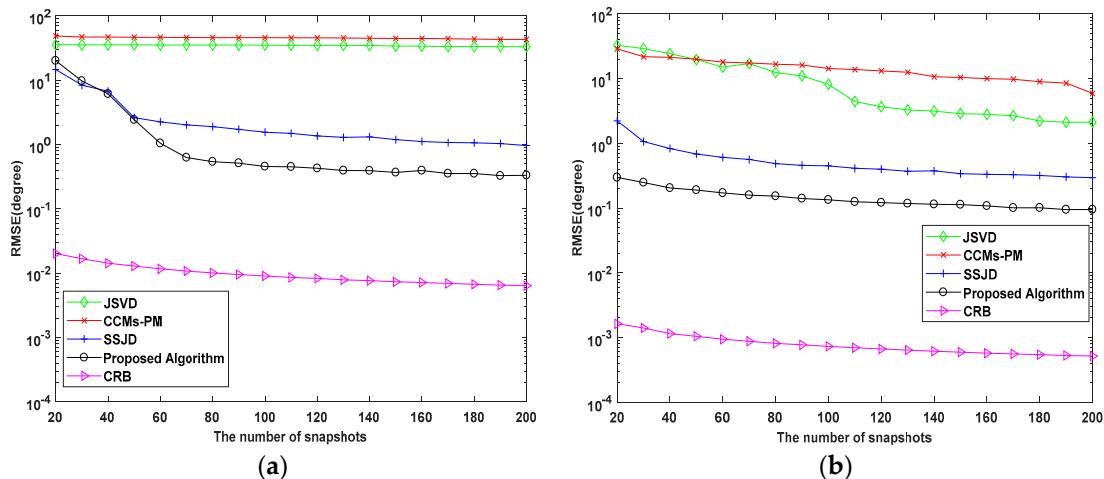


Figure 8. RMSE performance versus the number of snapshots for (a) SNR = 0 dB; (b) SNR = 10 dB.

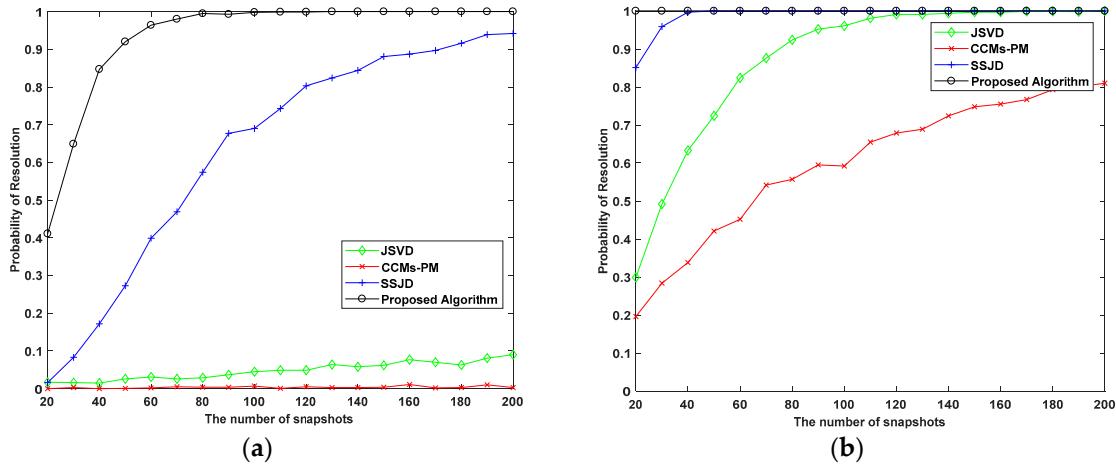


Figure 9. Probability of Resolution versus snapshots (a) SNR = 0 dB; (b) SNR = 10 dB.

In the fourth experiment, we examine the estimation performance of the proposed algorithm versus SNR for a small number of snapshots, where the SNR varies from 0 dB to 20 dB, and the number of snapshots are 10 and 20. Similarly, we compare the performance of the proposed algorithm with the CRB, and the JSVD, CCMs-PM, and SSJD algorithms with 1000 Monte Carlo trials. Figures 10 and 11 show the performance of the above algorithms versus SNR for a small number of snapshots.

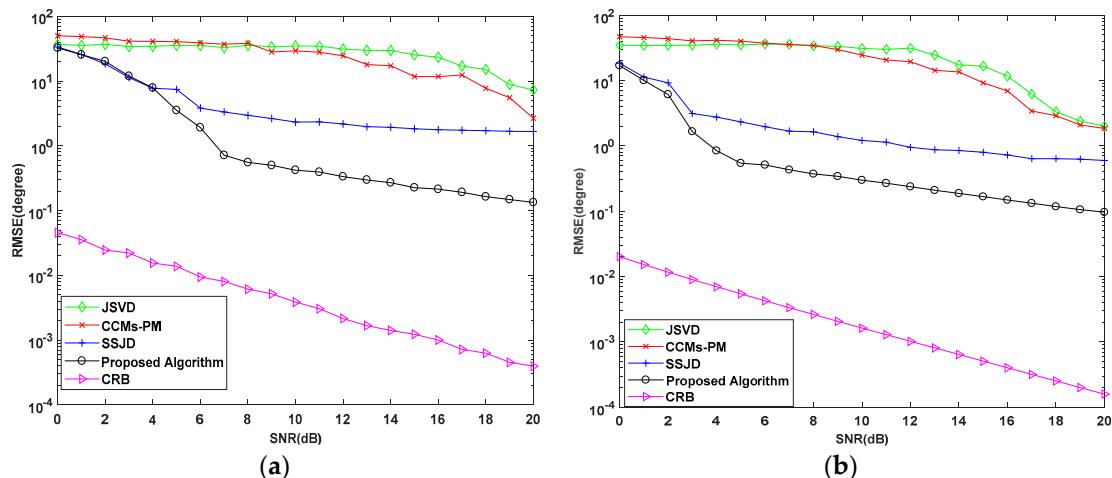


Figure 10. RMSE performance versus SNR for a small number of snapshots with (a) snapshots = 10; (b) snapshots = 20.

As shown in Figure 10, the proposed algorithm can obtain better estimation performance than the other algorithms. From Figure 10, the RMSE of the JSVD and CCMs-PM algorithms decline slowly at low SNR for a small number of snapshots. These results show that the effect of removing noise through the CCM is sensitive to the number of snapshots. The performance of SSJD is better than the JSVD and CCMs-PM algorithms, but inferior to the proposed algorithm. With an increase in the SNR, the estimation performance of the proposed algorithm dramatically decreased compared to the above algorithms, for a small number of snapshots. This is due to the reconstruction of the CCM based on SVD. When the SNR is greater than 10 dB, the RMSE of our algorithm is less than 1 degree with 10 snapshots, which verifies the effectiveness of the proposed algorithm under a small number of snapshots.

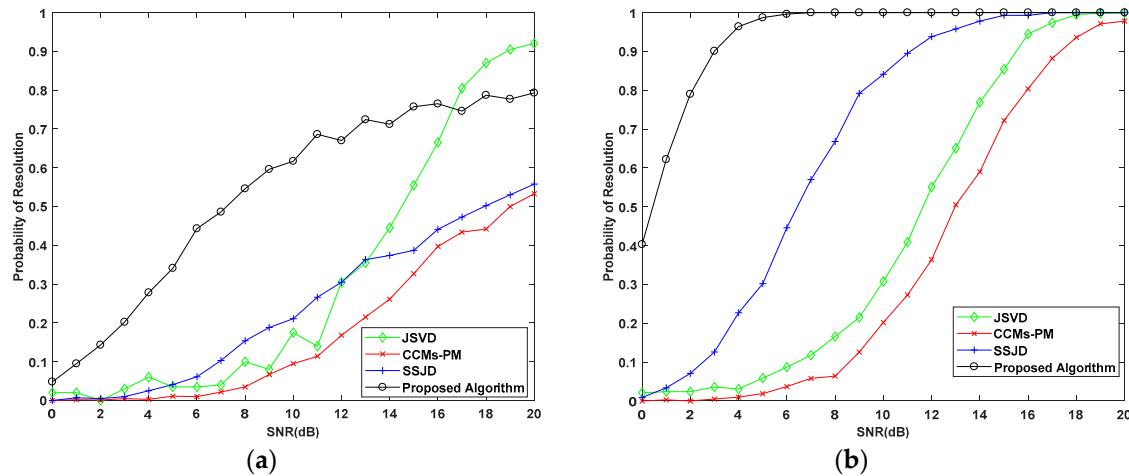


Figure 11. Probability of Resolution versus SNR for a small number of snapshots with (a) snapshots = 10; (b) snapshots = 20.

From Figure 11, when the number of snapshots is 10 and SNR is lower than 15 dB, the probability of the proposed algorithm is higher than other algorithms. When the number of snapshots is 10 and the SNR is higher than 15 dB, the probability of the JSVD algorithm, performed on the uniform arrays, is better than the proposed algorithm. The beam width is proportional to the array aperture. Therefore, the beam width for uniform L-shaped arrays is much higher than the nested L-shaped arrays, which make the probability of JSVD higher than the proposed algorithm in a high SNR environment. When the number of snapshots is 20, the probability of resolution for proposed algorithm is higher than the other compared algorithms for all studied SNRs. Furthermore, when SNR is higher than 10 dB, the probability of the proposed algorithm is 100%, which denotes the optimal performance of our algorithm.

For the sake of clarity, the RMSE of the proposed algorithm, the JSVD, the SSJD, and the CCMs-PM algorithms are listed in Table 2, where the number of snapshots are 10 and 20, the SNR are 0 dB, 5 dB, 10 dB.

Table 2. Comparison of estimation RMSE for different algorithms.

Algorithm	Snapshots	RMSE (0 dB)	RMSE (5 dB)	RMSE (10 dB)
JSVD	10	36.63°	35.52°	34.70°
CCMs-PM	10	49.84°	40.54°	29.28°
SSJD	10	33.44°	7.40°	2.33°
Proposed Algorithm	10	32.40°	3.52°	0.42°
JSVD	20	34.63°	35.12°	30.88°
CCMs-PM	20	46.68°	39.79°	24.43°
SSJD	20	18.28°	2.32°	1.21°
Proposed Algorithm	20	16.93°	0.5°	0.30°

As described in Table 2, the RMSE of the proposed algorithm is obviously lower than other compared algorithms, which indicates the best estimation performance among all considered algorithms for a small number of samples.

In the last experiment, the estimation performance of the proposed algorithm versus SNR for different sensor spacing. The SNR varies from 0 dB to 20 dB, and the number of snapshots is 20. The sensor spacing of inner ULA of nested arrays for simulation are 0.5λ and 0.4λ , respectively. The sensor spacing of outer ULA of nested arrays for simulation are 2.5λ and 2λ , respectively. Similarly, the sensor spacing of L-shaped uniform arrays for simulation are 0.5λ and 0.4λ , respectively. For ease of expression in the following analysis, we define the symbol d as the sensor spacing of L-shaped uniform arrays and the sensor spacing of inner ULA of nested arrays. We compare the performance of the

proposed algorithm with Cramér–Rao bound (CRB), JSVD, CCMs-PM, and SSJD algorithms with 1000 Monte Carlo trials.

As shown in Figure 12, as the increasing of SNR, the RMSE of the proposed algorithm is the lower than the other algorithms for different sensor spacing, which indicates the best performance of the proposed algorithm. From Figure 13, the probability of resolution of proposed algorithm is higher than the compared algorithms. When SNR is higher than 5 dB, the RMSE of the proposed algorithm is lower than 1 degree and the probability of resolution is almost 100% with both of the sensor spacing. Compare the RMSE curves and the probability of resolution curves with different sensor spacing, the RMSE for all of the algorithms with sensor spacing $d = 0.5\lambda$ are lower than these with sensor spacing $d = 0.4\lambda$. The probability of resolution with sensor spacing $d = 0.5\lambda$ of all the algorithms is higher than these with sensor spacing $d = 0.4\lambda$. This indicates that the estimation performance with bigger sensor spacing is better than the one with little sensor spacing. This is because that the sensor spacing is proportional to the effective array aperture, which determines the estimation accuracy of the estimation. However, the DOA estimation will lead to angle ambiguity when the sensor spacing of d is bigger than 0.5λ . Therefore, we choose the sensor spacing $d = \lambda/2$ as the array configuration of array sensor.

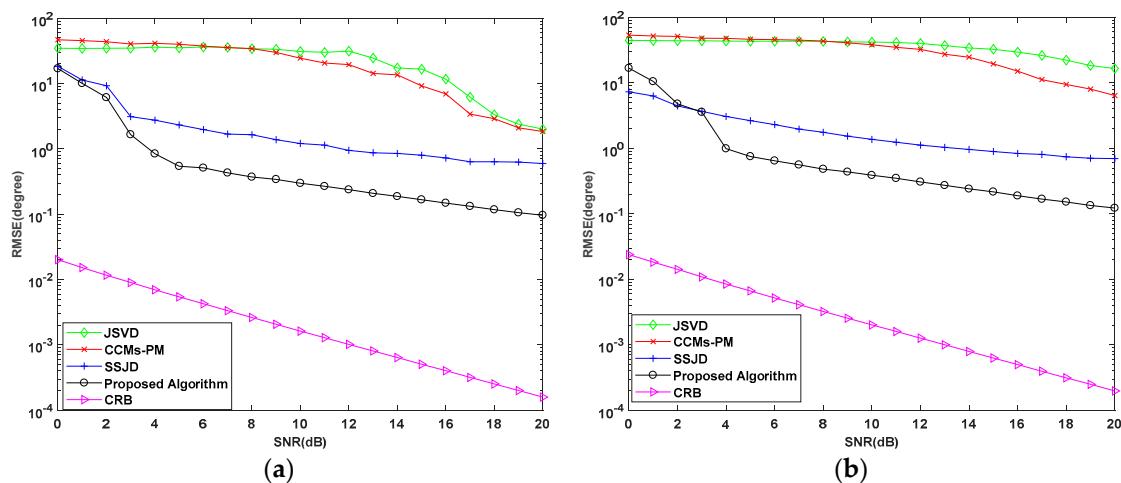


Figure 12. RMSE Performance versus SNR for different sensor spacing **(a)** $d = 0.5\lambda$; **(b)** $d = 0.4\lambda$.

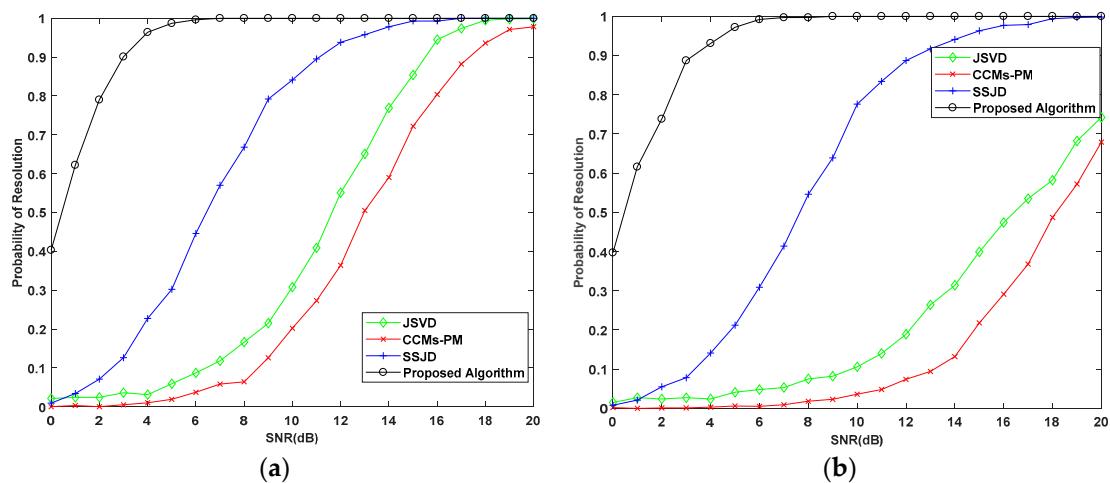


Figure 13. Probability of Resolution versus SNR for different sensor spacing **(a)** $d = 0.5\lambda$; **(b)** $d = 0.4\lambda$.

5. Conclusions

In this paper, we have presented an improved 2-D DOA estimation algorithm using an L-shaped nested array. To improve the estimation of CCM under small numbers of samples, a new CCM based on SVD was constructed. A novel virtual generation of nested array is considered by utilizing the

reconstructed CCM without the overlapping element and noise vector. After obtaining the virtual arrays, an improved unitary ESPRIT algorithm is performed on the full rank equivalent variance matrices, which are constructed by the output of the virtual arrays and their conjugate vectors, to estimate the azimuth, and elevation angles. The pair-matching method is presented by estimating the equivalent signal vector of the virtual arrays, which also estimate the power of incident signals. Several numerical experiments show a better estimation performance of the proposed algorithm in contrast to others, especially under a small number of snapshots. Considering the outperformance of the proposed algorithm under small samples, our algorithm can be directly used on mono-pulse angle measurement for radar and close-range directional detection.

Author Contributions: X.G. contributed to the main idea of methodology and array configuration design. X.H. participated in the methodology design and optimized the proposed algorithm. P.L. performed the simulation and analyzed the date. G.L. revised the manuscript and proposed constructive discussions for the articles. All authors have read and approved the final manuscript.

Funding: This research was supported by the National Science Foundation of China (NSFC) (Grant No. 61871414) and by the Innovation Fund (Grant No. 6141B012872).

Acknowledgments: The author would like to thank the Assistant Editor and the reviewers for their very helpful comments and suggestions that improved this paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

DOA	direction of arrival
CRB	Cram'er-Rao boundary
JSVD	joint singular value decomposition
ACM	autocorrelation matrix
CCM	cross correlation matrix
ESPRIT	estimation of signal parameters via rotational invariance technique
EVD	eigenvalue decomposition
PM	propagator method
ULA	uniform linear array
SLA	sparse linear arrays
SSJD	signal subspace joint diagonalization technique
SNR	signal-to-noise-ratio
SVD	singular value decomposition

Appendix A. CRB Analysis of L-Shaped Nested Arrays

The celebrated CRB expression which offers a theoretical lower bound on the unbiased DOA estimation for uniform arrays is not suitable for nested arrays. According to Refs. [42,43], we derive the CRB expression for L-shaped nested arrays.

From Equations (2) and (3), the combined received signal $\mathbf{Z}(t)$ is defined as:

$$\mathbf{Z}(t) = \begin{bmatrix} \mathbf{X}(t) \\ \mathbf{Y}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_x(\alpha) \\ \mathbf{A}_y(\beta) \end{bmatrix} \mathbf{S}(t) + \begin{bmatrix} \mathbf{N}_x(t) \\ \mathbf{N}_y(t) \end{bmatrix} = \mathbf{AS}(t) + \mathbf{N}(t), \quad (\text{A1})$$

where $\mathbf{A} = [\mathbf{A}_x^T(\alpha) \ \mathbf{A}_y^T(\beta)]^T$, and $\mathbf{N}(t) = [\mathbf{N}_x^T(t) \ \mathbf{N}_y^T(t)]^T$.

The covariance matrix is calculated via:

$$\mathbf{R} = E[\mathbf{Z}(t)\mathbf{Z}^H(t)] = \mathbf{AR}_s\mathbf{A}^H + \sigma^2\mathbf{I} \quad (\text{A2})$$

where $\mathbf{R}_s = \mathbf{S}(t)\mathbf{S}^H(t)/N = \text{diag}(p_1, p_2, \dots, p_k)$ denotes the source covariance matrix. Furthermore, define η as the parameter vector:

$$\eta = [\alpha_1, \dots, \alpha_k, \beta_1, \dots, \beta_k, p_1, p_2, \dots, p_k, \sigma^2]^T. \quad (\text{A3})$$

The (m, n) -th element of Fisher Information Matrix (FIM) is given by

$$\begin{aligned} \mathbf{F}_{(m,n)} &= N \text{tr} \left(\mathbf{R}^{-1} \frac{\partial \mathbf{R}}{\partial \eta_m} \mathbf{R}^{-1} \frac{\partial \mathbf{R}}{\partial \eta_n} \right) \\ &= N \left[\text{vec} \left(\frac{\partial \mathbf{R}}{\partial \eta_m} \right) \right]^H \left(\mathbf{R}^T \otimes \mathbf{R} \right)^{-1} \text{vec} \left(\frac{\partial \mathbf{R}}{\partial \eta_n} \right) \\ &= N \left(\frac{\partial r}{\partial \eta_m} \right)^H \left(\mathbf{R}^T \otimes \mathbf{R} \right)^{-1} \left(\frac{\partial r}{\partial \eta_n} \right) \end{aligned} \quad (\text{A4})$$

where $r = \text{vec}(\mathbf{R}) = (\mathbf{A}^* \odot \mathbf{A})p + \sigma^2 \mathbf{l}_n = \overline{\mathbf{A}}p + \sigma^2 \mathbf{l}_n$, and $p = [p_1, p_2, \dots, p_k]^T$.

From Equations (A2) and (A3),

$$\begin{aligned} \frac{\partial r}{\partial \eta} &= \left[\frac{\partial r}{\partial \alpha_1}, \dots, \frac{\partial r}{\partial \alpha_k}, \frac{\partial r}{\partial \beta_1}, \dots, \frac{\partial r}{\partial \beta_k}, \frac{\partial r}{\partial p_1}, \dots, \frac{\partial r}{\partial p_k}, \frac{\partial r}{\partial \sigma^2} \right] \\ &= \left[\left(\frac{\partial \mathbf{A}^*}{\partial \alpha} \odot \mathbf{A} + \mathbf{A}^* \odot \frac{\partial \mathbf{A}}{\partial \alpha} \right) \mathbf{P}, \left(\frac{\partial \mathbf{A}^*}{\partial \beta} \odot \mathbf{A} + \mathbf{A}^* \odot \frac{\partial \mathbf{A}}{\partial \beta} \right) \mathbf{P}, (\mathbf{A}^* \odot \mathbf{A}), \mathbf{l}_n \right] \end{aligned} \quad (\text{A5})$$

where $\mathbf{P} = \text{diag}(\mathbf{p}) = \text{diag}[p_1, p_2, \dots, p_k]$, $\theta = [\alpha_1, \dots, \alpha_k, \beta_1, \dots, \beta_k]$.

According to Equation (A4), the FIM can be expressed as:

$$\mathbf{F} = N \begin{bmatrix} \mathbf{M}_\theta^H \mathbf{M}_\theta \mathbf{M}_\theta^H \mathbf{M}_s \\ \mathbf{M}_s^H \mathbf{M}_\theta \mathbf{M}_\theta^H \mathbf{M}_s \end{bmatrix} \quad (\text{A6})$$

where $\mathbf{M}_\theta = (\mathbf{R}^T \otimes \mathbf{R})^{-1/2} \frac{\partial \overline{\mathbf{A}}}{\partial \theta} \mathbf{P}$, and $\mathbf{M}_s = (\mathbf{R}^T \otimes \mathbf{R})^{-1/2} [(\mathbf{A}^* \odot \mathbf{A}), \mathbf{l}_n]$. Thus, the CRB can be obtained by block-wise inversion:

$$\text{CRB}_\theta = \frac{1}{N} \left[\mathbf{M}_\theta^H \left(\mathbf{I} - \mathbf{M}_s \left(\mathbf{M}_s^H \mathbf{M}_s \right)^{-1} \mathbf{M}_s^H \right) \mathbf{M}_\theta \right]^{-1}. \quad (\text{A7})$$

References

- Chen, Z.; Xu, W. Joint Passive Detection and Tracking of Underwater Acoustic Target by Beamforming-Based Bernoulli Filter with Multiple Arrays. *Sensors* **2018**, *18*, 4022. [[CrossRef](#)] [[PubMed](#)]
- Zhou, H.; Hu, G.; Shi, J.; Feng, Z. Multi-Frequency Based Direction-of-Arrival Estimation for 2q-Level Nested Radar & Sonar Arrays. *Sensors* **2018**, *18*, 3385. [[CrossRef](#)]
- Xia, T.-Q. Joint diagonalization based 2D-DOD and 2D-DOA estimation for bistatic MIMO radar. *Signal Process.* **2015**, *116*, 7–12. [[CrossRef](#)]
- Lopatka, J.; Matuszewski, J.; Dikta, A. Emitter location errors in electronic recognition system. In Proceedings of the XI Conference on Reconnaissance and Electronic Warfare Systems, Oltarzew, Poland, 21–23 November 2016. [[CrossRef](#)]
- Dudczyk, J.; Kawalec, A.; Cyrek, J. Applying the distance and similarity functions to radar signals identification. In Proceedings of the International Radar Symposium, Wroclaw, Poland, 21–23 May 2008; pp. 1–4. [[CrossRef](#)]
- Dudczyk, J.; Kawalec, A.; Owczarek, R. An application of iterated function system attractor for specific radar source identification. In Proceedings of the International Conference on Microwaves Radar Wireless Communications, Wroclaw, Poland, 19–21 May 2008; pp. 1–4.
- Matuszewski, J. Identification of radar signals using discriminant vectors. In Proceedings of the International Conference on Microwaves Radar Wireless Communications, Wroclaw, Poland, 19–21 May 2008; pp. 1–4.
- Schmidt, R. Propagation. Multiple emitter location and signal parameter estimation. *IEEE Trans. Antennas Propag.* **1986**, *34*, 276–280. [[CrossRef](#)]

9. Ciuonzo, D. On Time-Reversal Imaging by Statistical Testing. *IEEE Signal Process. Lett.* **2017**, *24*, 1024–1028. [[CrossRef](#)]
10. Ciuonzo, D.; Rossi, P.S. Noncolocated Time-Reversal MUSIC: High-SNR Distribution of Null Spectrum. *IEEE Signal Process. Lett.* **2017**, *24*, 397–401. [[CrossRef](#)]
11. Ciuonzo, D.; Romano, G.; Solimene, R. Performance Analysis of Time-Reversal MUSIC. *IEEE Trans. Signal Process.* **2015**, *63*, 2650–2662. [[CrossRef](#)]
12. Barabell, A.J. Improving the resolution performance of eigenstructure-based direction-finding algorithms. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Boston, MA, USA, 14–16 April 1983; pp. 336–339. [[CrossRef](#)]
13. Paulraj, A.; Roy, R.; Kailath, T. Estimation of Signal Parameters via Rotational Invariance Techniques-Esprit. In Proceedings of the Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 6–8 November 1985; pp. 83–89. [[CrossRef](#)]
14. Xia, T.; Zheng, Y.; Wan, Q.; Wang, X. Decoupled Estimation of 2-D Angles of Arrival Using Two Parallel Uniform Linear Arrays. *IEEE Trans. Antennas Propag.* **2007**, *55*, 2627–2632. [[CrossRef](#)]
15. Zheng, Z.; Yang, Y.; Wang, W.-Q.; Zhang, S. Two-dimensional direction estimation of multiple signals using two parallel sparse linear arrays. *Signal Process.* **2018**, *143*, 112–121. [[CrossRef](#)]
16. Saravananakumar, N.; Sathyasekar, K. Efficient method for location and detection of partial discharge in transformer oil by DOA estimation of circular array of ultrasonic sensors. *J. Instrum.* **2018**, *13*, P01026. [[CrossRef](#)]
17. Jackson, B.R.; Rajan, S.; Liao, B.J.; Wang, S. Direction of Arrival Estimation Using Directive Antennas in Uniform Circular Arrays. *IEEE Trans. Antennas Propag.* **2015**, *63*, 736–747. [[CrossRef](#)]
18. Shi, J.; Hu, G.; Zhang, X.; Sun, F.; Xiao, Y. Computationally Efficient 2D DOA Estimation with Uniform Rectangular Array in Low-Grazing Angle. *Sensors* **2017**, *17*, 470. [[CrossRef](#)] [[PubMed](#)]
19. Shi, J.; Hu, G.; Sun, F.; Zong, B.; Wang, X. Improved Spatial Differencing Scheme for 2-D DOA Estimation of Coherent Signals with Uniform Rectangular Arrays. *Sensors* **2017**, *17*, 1956. [[CrossRef](#)] [[PubMed](#)]
20. Wisudawan, H.N.P.; Hidayat, R.; Ariananda, D.D. Two Dimensional Angle of Arrival Estimation Using Minimum Sparse Ruler Based Rectangular Array of Antennas. In Proceedings of the 2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE), Phuket, Thailand, 12–13 October 2017.
21. Porozantzidou, M.G.; Chryssomallis, M.T. Azimuth and elevation angles estimation using 2-D MUSIC algorithm with an L-shape antenna. In Proceedings of the IEEE Antennas and Propagation Society International Symposium, Toronto, ON, Canada, 11–17 July 2010; pp. 1–4. [[CrossRef](#)]
22. Tayem, N.; Kwon, H.M. L-shape 2-dimensional arrival angle estimation with propagator method. *IEEE Trans. Antennas Propag.* **2005**, *53*, 1622–1630. [[CrossRef](#)]
23. Kikuchi, S.; Tsuji, H.; Sano, A. Pair-Matching Method for Estimating 2-D Angle of Arrival with a Cross-Correlation Matrix. *IEEE Antennas Wirel. Propag. Lett.* **2006**, *5*, 35–40. [[CrossRef](#)]
24. Gu, J.-F.; Wei, P. Joint SVD of Two Cross-Correlation Matrices to Achieve Automatic Pairing in 2-D Angle Estimation Problems. *IEEE Antennas Wirel. Propag. Lett.* **2007**, *6*, 553–556. [[CrossRef](#)]
25. Zhang, X.; Gao, X.; Chen, W. Improved Blind 2d-Direction of Arrival Estimation with L-Shaped Array Using Shift Invariance Property. *J. Electromagnet. Wave* **2009**, *23*, 593–606. [[CrossRef](#)]
26. Wang, G.M.; Xin, J.M.; Zheng, N.N.; Sano, A. Computationally Efficient Subspace-Based Method for Two-Dimensional Direction Estimation with L-Shaped Array. *IEEE Trans. Signal Process.* **2011**, *59*, 3197–3212. [[CrossRef](#)]
27. Dong, Y.Y.; Dong, C.X.; Liu, W.; Chen, H.; Zhao, G.Q. 2-D DOA Estimation for L-Shaped Array with Array Aperture and Snapshots Extension Techniques. *IEEE Signal Process. Lett.* **2017**, *24*, 495–499. [[CrossRef](#)]
28. Nie, X.; Qian, G.B.; Xie, X.B.; Zeng, K.; Zhang, K.S. A 2-D DOA Estimation Algorithm for Closely Spaced Sources with L-Shaped Array. *Circ. Syst. Signal Process.* **2017**, *36*, 4498–4511. [[CrossRef](#)]
29. Wang, Q.; Yang, H.; Chen, H.; Dong, Y.Y.; Wang, L.H. A Low-Complexity Method for Two-Dimensional Direction-of-Arrival Estimation Using an L-Shaped Array. *Sensors* **2017**, *17*, 190. [[CrossRef](#)]
30. Liang, J.; Liu, D. Joint Elevation and Azimuth Direction Finding Using L-Shaped Array. *IEEE Trans. Antennas Propag.* **2010**, *58*, 2136–2141. [[CrossRef](#)]
31. Hua, Y.B.; Sarkar, T.K.; Weiner, D.D. An L-Shaped Array for Estimating 2-D Directions of Wave Arrival. *IEEE Trans. Antennas Propag.* **1991**, *39*, 143–146. [[CrossRef](#)]

32. Moffet, A.T. Minimum-Redundancy Linear Arrays. *IEEE Trans. Antennas Propag.* **1968**, *Ap16*, 172–175. [[CrossRef](#)]
33. Vaidyanathan, P.P.; Pal, P. Sparse Sensing With Co-Prime Samplers and Arrays. *IEEE Trans. Signal Process.* **2011**, *59*, 573–586. [[CrossRef](#)]
34. Pal, P.; Vaidyanathan, P.P. Nested Arrays: A Novel Approach to Array Processing With Enhanced Degrees of Freedom. *IEEE Trans. Signal Process.* **2010**, *58*, 4167–4181. [[CrossRef](#)]
35. Zhou, C.; Gu, Y.; Shi, Z.; Zhang, Y.D. Off-Grid Direction-of-Arrival Estimation Using Coprime Array Interpolation. *IEEE Signal Process. Lett.* **2018**, *25*, 1710–1714. [[CrossRef](#)]
36. Zhou, C.W.; Gu, Y.J.; Fan, X.; Shi, Z.G.; Mao, G.Q.; Zhang, Y.M.D. Direction-of-Arrival Estimation for Coprime Array via Virtual Array Interpolation. *IEEE Trans. Signal Process.* **2018**, *66*, 5956–5971. [[CrossRef](#)]
37. Gu, J.-F.; Zhu, W.-P.; Swamy, M.N.S. Joint 2-D DOA Estimation via Sparse L-shaped Array. *IEEE Trans. Signal Process.* **2015**, *63*, 1171–1182. [[CrossRef](#)]
38. Niu, C.; Zhang, Y.; Guo, J. Interlaced Double-Precision 2-D Angle Estimation Algorithm Using L-Shaped Nested Arrays. *IEEE Signal Process. Lett.* **2016**, *23*, 522–526. [[CrossRef](#)]
39. Dong, Y.-Y.; Dong, C.-X.; Zhu, Y.-T.; Zhao, G.-Q.; Liu, S.-Y. Two-dimensional DOA estimation for L-shaped array with nested subarrays without pair matching. *IET Signal Process.* **2016**, *10*, 1112–1117. [[CrossRef](#)]
40. Mestre, X.; Lagunas, M.Á. Modified Subspace Algorithms for DoA Estimation With Large Arrays. *IEEE Trans. Signal Process.* **2008**, *56*, 598–614. [[CrossRef](#)]
41. Haardt, M.; Nossek, J.A. Unitary Esprit—How to Obtain Increased Estimation Accuracy with a Reduced Computational Burden. *IEEE Trans. Signal Process.* **1995**, *43*, 1232–1242. [[CrossRef](#)]
42. Liu, C.-L.; Vaidyanathan, P.P. Cramér–Rao bounds for coprime and other sparse arrays, which find more sources than sensors. *Digit. Signal Process.* **2017**, *61*, 43–61. [[CrossRef](#)]
43. Wang, M.; Nehorai, A. Coarrays, MUSIC, and the Cramér–Rao Bound. *IEEE Trans. Signal Process.* **2017**, *65*, 933–946. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).