

# Vehicle Positioning With Auxiliary Improved Differential-Evolution-Based 2-D DOA Estimation for Orthogonal Distributed Arrays

Mo Chen<sup>ID</sup>, Xiangtian Meng<sup>ID</sup>, Member, IEEE, Runhu Liu<sup>ID</sup>, Bingxia Cao<sup>ID</sup>, Fenggang Yan<sup>ID</sup>, Member, IEEE, and Ming Jin<sup>ID</sup>

**Abstract**—Direction-of-arrival (DOA) estimation using distributed arrays has emerged as a promising technique for autonomous vehicle (AV) positioning in Internet of Vehicles (IoV) systems. This article proposes a novel DOA estimation method based on orthogonal distributed arrays for accurate AV localization. Specifically, the covariance matrix between orthogonal arrays is exploited to construct two univariate polynomials for DOA estimation. As the polynomial degree depends on the array aperture and the discontinuous sensor layout, the resulting polynomials are often high-order and lacunary. To efficiently solve these polynomials, an improved differential-evolution (DE) algorithm is developed, featuring an adaptive mutation strategy to reduce computational cost and a counter-based mechanism to escape local optima. In addition, a covariance-based cost function is designed for 2-D angle pairing. The array aperture is further extended to enable simultaneous DOA estimation of multiple vehicles when the orthogonal arrays share a common sensor. Simulation results demonstrate that, compared with existing methods, the proposed approach achieves higher estimation accuracy and lower computational complexity, offering a promising solution for vehicle positioning in IoV environments.

**Index Terms**—Adaptive mutation strategy, autonomous vehicle (AV) positioning, differential-evolution (DE), direction-of-arrival (DOA) estimation, distributed array, Internet of Vehicles (IoV), orthogonal array.

## I. INTRODUCTION

WITH the rapid development of intelligent transportation systems, connected and autonomous vehicles (AVs) in vehicular networks are placing increasingly stringent demands on positioning technologies in terms of both accuracy and reliability. Conventional single-vehicle localization approaches primarily rely on global navigation satellite systems (GNSSs), inertial navigation systems (INSs), and vision/LiDAR-based simultaneous localization and mapping (SLAM) techniques

Received 28 July 2025; revised 12 November 2025; accepted 23 November 2025. Date of publication 26 November 2025; date of current version 5 February 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62171150; in part by the Taishan Scholar Special Funding Project of Shandong Province under Grant tsqn202211087; in part by Shandong Provincial Natural Science Foundation under Grant ZR2023MF091, Grant ZR2024MF071, and Grant ZR2024QF068; and in part by the Aeronautical Science Foundation of China under Grant 2023Z037077002. (Corresponding author: Bingxia Cao.)

The authors are with the School of Information Science and Engineering, Harbin Institute of Technology at Weihai, Weihai 264209, China (e-mail: fund1biao@163.com; mxxtian@hit.edu.cn; hit\_liurunhu@163.com; cbxhit@163.com; yfglion@163.com; jinming0987@163.com).

Digital Object Identifier 10.1109/JIOT.2025.3637211

[1]. While GNSS demonstrates satisfactory positioning accuracy in open environments, its performance significantly deteriorates in urban canyons and tunnels due to severe signal blockage and multipath effects. Although INS provides high short-term precision, it suffers from inherent error accumulation over time [2]. SLAM techniques, on the other hand, are particularly vulnerable to environmental dynamics and adverse weather conditions [3]. These inherent limitations make it challenging for traditional methods to independently meet the demanding requirements of continuous, stable, and high-precision localization for autonomous driving in complex urban environments, thereby driving researchers to investigate cooperative positioning methods within vehicular networks.

Cooperative positioning enhances both accuracy and reliability through information sharing and collaborative efforts among vehicles or between vehicles and infrastructure [4]. Primary techniques include distance-based approaches such as time of arrival (TOA) [5] and time difference of arrival (TDOA) [6], distance-difference-based methods including received signal strength indicator (RSSI) [7], and relative-position-based solutions like visual feature matching. However, each of these methods exhibits notable limitations: distance-based techniques impose stringent requirements on clock synchronization while being particularly susceptible to errors in multipath-rich urban environments; RSSI-based approaches are highly vulnerable to environmental interference, resulting in compromised positioning accuracy; and vision-based feature matching suffers from excessive computational complexity and performance degradation under poor lighting conditions or in feature-deficient scenarios.

Against this backdrop, direction-of-arrival (DOA)-based cooperative positioning has emerged as a promising approach with distinctive advantages. This technique offers several superior characteristics, including strong resilience to multipath interference, relaxed synchronization requirements, and moderate computational complexity. By leveraging roadside base stations as reference nodes, DOA-based positioning can achieve stable and high-precision localization even in GNSS-denied environments [8]. Consequently, DOA estimation has become a focal research topic in cooperative positioning within vehicular networks [9], [10], [11], [12], [13], [14], [15]. Previous studies have explored a variety of DOA-based positioning schemes utilizing LTE base stations. Building upon

these advances, this work focuses on a DOA-based positioning architecture tailored for AVs.

DOA estimation has been extensively investigated for various array geometries, including uniform rectangular arrays [16], L-shaped arrays [17], [18], double parallel linear arrays [19], [20], and uniform circular arrays [21]. Numerous high-precision DOA estimation algorithms have been developed, among which subspace-based methods such as multiple signal classification (MUSIC) and estimation of signal parameters via rotational invariance techniques (ESPRITs) are capable of achieving super-resolution performance under ideal conditions [22], [23], [24]. In addition, sparse reconstruction techniques, including sparse Bayesian learning (SBL) and compressed sensing-based DOA (CS-DOA), effectively mitigate the dependence on array aperture. More recently, deep learning-based DOA estimation methods have enhanced robustness in complex environments through data-driven modeling [25]. It is also worth noting that 2-D DOA estimation techniques—such as MUSIC-like and SBL-based approaches—have been successfully applied to vehicle localization and tracking, enabling azimuth estimation via antenna arrays deployed at roadside base stations [26], [27], [28]. Nevertheless, in real-world Internet of Vehicles (IoV) scenarios, particularly in dense urban road environments, existing DOA estimation algorithms still encounter the following critical challenges when applied to practical vehicle localization tasks.

- 1) Most existing DOA algorithms are validated using small-scale arrays in simulations. In practical vehicle localization, physical size constraints lead to insufficient array aperture, degrading angular resolution. For instance, in dense traffic scenarios, small arrays struggle to distinguish between closely spaced vehicles, resulting in localization ambiguity and mismatches that severely compromise positioning accuracy.
- 2) Traditional spectral search-based algorithms, such as 2-D-MUSIC, require exhaustive scanning over a 2-D angular space, leading to exponentially increasing computational costs with finer search steps. Moreover, most DOA methods rely on eigenvalue decomposition (EVD) of the covariance matrix to extract signal subspaces, with a complexity of  $O(N^3)$ , where  $N$  is the number of array elements [29], [30], [31]. For base station processing centers, this imposes an impractical computational burden, hindering low-latency requirements [32], [33], [34].
- 3) Although sparse reconstruction methods like SBL and expectation-maximization (EM) algorithms theoretically reduce array element requirements [35], [36], their iterative optimization processes suffer from slow convergence and sensitivity to initial values, making them unsuitable for real-time applications in dynamic vehicular environments. In addition, while deep learning-based DOA estimation avoids EVD [37], it demands extensive labeled training data, raising concerns about feasibility in real-world vehicle localization tasks.

To address the aforementioned challenges, this article proposes a DOA estimation method utilizing orthogonal distributed arrays. Compared with existing DOA estimation methods applied to vehicle localization, our method establishes

a complete computational framework from received data to DOA estimation, achieving higher accuracy with significantly reduced computational complexity. The key contributions are as follows.

- 1) Instead of conventional compact arrays, we employ orthogonal distributed arrays that achieve extended array aperture while maintaining equivalent hardware costs, thereby improving angular resolution without additional sensor expenditure.
- 2) Departing from traditional EVD-based subspace methods, our work proposes a novel approach that reconstructs angle-dependent matrices through selective row/column element matrix manipulation of the cross-covariance matrix. Following the root-MUSIC paradigm, we formulate two univariate polynomials based on array geometry, whose roots directly yield 2-D DOA estimates, eliminating the need for both EVD and exhaustive spectral search while maintaining low computational complexity.
- 3) The discontinuous sensor distribution characteristic of distributed arrays generates high-order lacunary polynomials in our method. To address this challenging root-finding problem, the differential-evolution (DE) algorithm is pioneered for solving root-MUSIC polynomials in this study, with its practical feasibility for real-time vehicle localization applications being thoroughly demonstrated.
- 4) Beyond conventional DE implementation, our work develops a phase-adaptive mutation strategy selection, and introduces a counter mechanism to mitigate local optima convergence. These innovations collectively improve root-finding efficiency and ensure real-time performance in practical localization scenarios.

The remainder of this article is organized as follows. Section II establishes the system architecture of vehicle localization using orthogonal distributed arrays for 2-D DOA estimation. Section III elaborates on the proposed DOA estimation method based on the DE algorithm. Simulation results and corresponding analyses are presented in Section IV, followed by concluding remarks in Section V.

## II. POSITIONING SYSTEM AND DATA MODEL

The architecture of the adopted vehicle positioning system is illustrated in Fig. 1(a), which primarily consists of two key components: 1) vehicle terminals equipped with wireless signal transmitters and 2) base stations deployed with an orthogonally distributed antenna array. In the practical implementation of vehicle positioning, moving vehicles first transmit positioning signals via their onboard transmitters. The base station arrays then receive these signals and perform local DOA estimation. Finally, multiple neighboring base stations transmit their local DOA estimates to a cloud center, where triangulation positioning is performed using selected base stations' estimation results to determine the vehicle's location.

The key to this architecture lies in an accurate and robust DOA estimation method. Consider an orthogonal distributed array along the  $x$ - and  $z$ -axes with several subarrays per

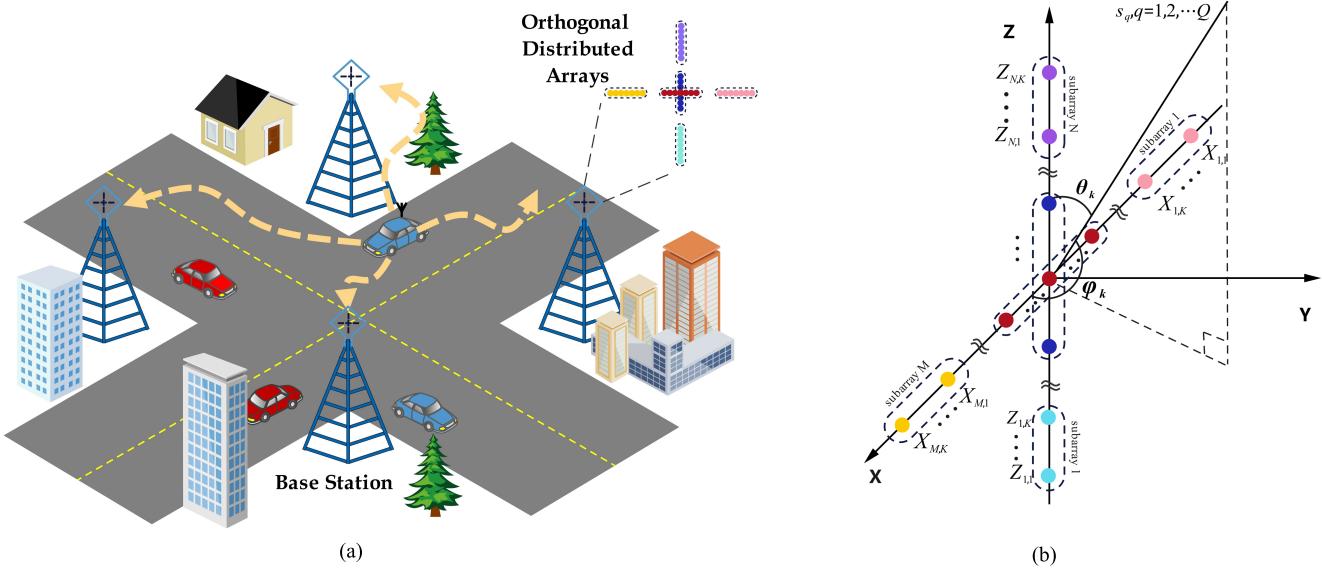


Fig. 1. (a) Vehicle positioning system utilizing multiple collaborative BSs. (b) Orthogonal distributed arrays adopted for DOA estimation.

axis—the cross-shaped array or L-shaped array, etc. There are  $M$  subarrays on the  $x$ -axis and  $N$  subarrays on the  $z$ -axis, with each subarray being a uniform linear array composed of  $K$  sensor elements. For sensors in an array, the following assumptions hold.

- A1) The sensors use a common clock for sampling.
- A2) The sensors within each subarray are uniformly distributed with a spacing of half a wavelength.
- A3) The position of each subarray is known, and the distance between subarrays is an integer multiple of half the wavelength.

As shown in Fig. 1(b), taking an orthogonal distributed cross-shaped array as an example. It is supposed that  $Q$  uncorrelated narrowband signals  $S_q$  impinge on the array from distinct directions with the unknown angles  $\{\varphi_q, \theta_q\}$ , where the elevation angle  $\theta_q$  and azimuth angle  $\varphi_q$  are measured clockwise relatively to the  $x$ - and  $z$ -axes.

The steering matrix of the arrays on the  $x$ - and  $z$ -axes can be given as

$$\mathbf{A}(\varphi) = [\mathbf{a}(\varphi_1), \mathbf{a}(\varphi_2), \dots, \mathbf{a}(\varphi_Q)] \quad (1)$$

$$\mathbf{A}(\theta) = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_Q)] \quad (2)$$

where  $\mathbf{a}(\varphi_q)$  and  $\mathbf{a}(\theta_q)$  are the steering vectors related to the  $q$ th signal, which can be partitioned according to the subarrays. The above steering vectors can be expressed as follows:

$$\begin{aligned} \mathbf{a}(\varphi_q) &= [\mathbf{a}_1(\varphi_q), \dots, \mathbf{a}_M(\varphi_q)]^T \\ \mathbf{a}(\theta_q) &= [\mathbf{a}_1(\theta_q), \dots, \mathbf{a}_N(\theta_q)]^T. \end{aligned} \quad (3)$$

The steering vector of the  $m$ th subarray on the  $x$ -axis is expressed as follows:

$$\begin{aligned} \mathbf{a}_m(\varphi_q) &= [e^{j\frac{2\pi}{\lambda}X_{m,1}(d)\cos\varphi_q}, \dots, e^{j\frac{2\pi}{\lambda}X_{m,K}d\cos\varphi_q}] \\ &= [\xi_q^{X_{m,1}}, \dots, \xi_q^{X_{m,K}}] \end{aligned} \quad (4)$$

where  $\xi_q = \exp(j(2\pi)/\lambda\cos\varphi_q)$  and  $d$  represents the interelement spacing within the subarray,  $X_{m,k}d$  is the position of the

$k$ th sensor in the  $m$ th subarray along the  $x$ -axis relative to the intersection of the orthogonal array.

The steering vector of the  $n$ th subarray on the  $z$ -axis is given by

$$\begin{aligned} \mathbf{a}_n(\theta_q) &= [e^{j\frac{2\pi}{\lambda}Z_{n,1}d\cos\theta_q}, \dots, e^{j\frac{2\pi}{\lambda}Z_{n,K}d\cos\theta_q}] \\ &= [\zeta_q^{Z_{n,1}}, \dots, \zeta_q^{Z_{n,K}}] \end{aligned} \quad (5)$$

where  $\zeta_q = \exp(j(2\pi)/\lambda\cos\theta_q)$  and  $Z_{n,k}d$  is the position of the  $k$ th sensor in the  $n$ th subarray along the  $z$ -axis relative to the intersection of the orthogonal array.

Then the received signals at the distributed cross-shaped array are given by

$$\mathbf{x}(t) = \mathbf{A}(\varphi)\mathbf{s}(t) + \mathbf{w}_x(t) \quad (6)$$

$$\mathbf{z}(t) = \mathbf{A}(\theta)\mathbf{s}(t) + \mathbf{w}_z(t) \quad (7)$$

where  $\mathbf{s}(t) = [s_1(t), \dots, s_Q(t)]^T$  is the snapshot data of source signal, and  $\mathbf{w}_x(t)$  and  $\mathbf{w}_z(t)$  are the additive Gaussian white noise with zero-mean and variance  $\sigma^2$  in the  $x$ - and  $z$ -axes, respectively, they are statistically independent of signal samples, i.e.,  $E\{\mathbf{w}_x(t)\mathbf{w}_z^H(t)\} = 0$ .

The cross-covariance matrix of data received by orthogonal arrays along the  $x$ - and  $z$ -axes, expressed as

$$\begin{aligned} \mathbf{R}_{xz} &= E\{\mathbf{x}(t)\mathbf{z}^H(t)\} \\ &= \mathbf{A}(\varphi)E\{\mathbf{s}(t)\mathbf{s}^H(t)\}\mathbf{A}^H(\theta) + E\{\mathbf{w}_x(t)\mathbf{w}_z^H(t)\} \\ &= \mathbf{A}(\varphi)\mathbf{R}_s\mathbf{A}^H(\theta) \end{aligned} \quad (8)$$

where the covariance matrix  $\mathbf{R}_s = \text{diag}\{s_1, s_2, \dots, s_Q\}$  of the signal sources is a diagonal matrix, and the diagonal elements  $s_q$  is the power of  $q$ th signal [38].

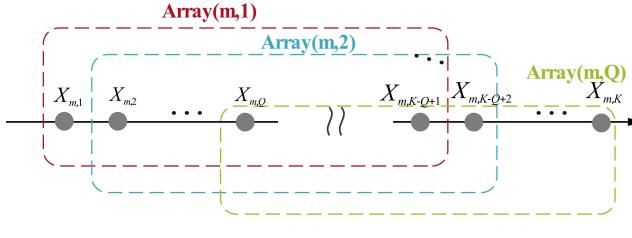


Fig. 2. Matrix reconstruction process for ULA.

### III. IMPROVED DE METHOD FOR 2-D DOA ESTIMATION IN THE DISTRIBUTED ARRAYS

#### A. Azimuth Estimation via Solving High-Degree Lacunary Polynomials

Regard the first column of  $\mathbf{R}_{xz}$  as  $\mathbf{c}$ ,  $\mathbf{c} = \mathbf{R}_{xz}(:, 1)$  is a linear combination of steering vectors about azimuth, according to (8), it can be decomposed into

$$\begin{aligned}\mathbf{c} &= \mathbf{A}(\varphi) \mathbf{R}_s \mathbf{A}_{(:,1)}^H(\theta) \\ &= \mathbf{A}(\varphi) \left[ s_1 \zeta_1^{Z_{1,1}}, \dots, s_Q \zeta_Q^{Z_{1,1}} \right]^H\end{aligned}\quad (9)$$

where  $\mathbf{A}_{(:,1)}^H(\theta)$  represents the first column of  $\mathbf{A}^H(\theta)$ . According to (9),  $\mathbf{c}$  can be treated as the single snapshot data received by the array along the  $x$ -axis from the signal source. Therefore,  $\mathbf{c}$  can be divided into the following form:

$$\mathbf{c} = [\mathbf{c}_1^T, \mathbf{c}_2^T, \dots, \mathbf{c}_M^T]^T \quad (10)$$

where column vector  $\mathbf{c}_m$  is the single snapshot data received by the  $m$ th subarray along the  $x$ -axis and  $\mathbf{c}_m$  can be expressed as

$$\mathbf{c}_m = \mathbf{A}^{(m)}(\varphi) \mathbf{R}_s \begin{bmatrix} \zeta_1^{Z_{m,1}} \\ \vdots \\ \zeta_Q^{Z_{m,1}} \end{bmatrix} \quad (11)$$

and  $\mathbf{A}^{(m)}(\varphi)$  is steering matrix for the  $m$ th subarray in  $x$ -axis, which is

$$\mathbf{A}^{(m)}(\varphi) = [\mathbf{a}_m(\varphi_1), \mathbf{a}_m(\varphi_2), \dots, \mathbf{a}_m(\varphi_Q)]. \quad (12)$$

Construct a matrix  $\Gamma_m$  from the elements of vector  $\mathbf{c}_m$ , which is

$$\begin{aligned}\Gamma_m &= \begin{bmatrix} \mathbf{c}_m(1) & \mathbf{c}_m(2) & \cdots & \mathbf{c}_m(Q) \\ \mathbf{c}_m(2) & \mathbf{c}_m(3) & \cdots & \mathbf{c}_m(Q+1) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_m(K-Q+1) & \cdots & \cdots & \mathbf{c}_m(K) \end{bmatrix} \\ &= [\mathbf{c}_{m,1}, \mathbf{c}_{m,2}, \dots, \mathbf{c}_{m,Q}]\end{aligned}\quad (13)$$

where  $\mathbf{c}_{m,q} \in \mathbb{C}^{(K-Q+1) \times 1}$  is the single snapshot data received by the  $Array(m, q)$  in the  $k$ th subarray in Fig. 2, and  $\mathbf{c}_{m,q}$  consists of elements from  $q$ th to  $(K-Q+q)$ th in  $\mathbf{c}_m$ .

Following the decomposition in (9),  $\Gamma_m$  can be expressed into the following form:

$$\Gamma_m = \mathbf{U}_x^{(m)} \mathbf{R}_s \text{diag}(\mathbf{A}_{(:,1)}^H(\theta)) \mathbf{V}_x \quad (14)$$

where  $\mathbf{U}_x^{(m)}$  is the steering vector matrix composed of the first  $(K-Q+1)$  sensors of the  $m$ th subarray in  $x$ -axis and  $\mathbf{V}_x$  is a Vandermonde matrices, which can be given by

$$\mathbf{U}_x^{(m)} = \begin{bmatrix} \xi_1^{X_{m,1}} & \xi_2^{X_{m,1}} & \cdots & \xi_Q^{X_{m,1}} \\ \vdots & \vdots & \ddots & \vdots \\ \xi_1^{X_{m,K-Q+1}} & \xi_2^{X_{m,K-Q+1}} & \cdots & \xi_Q^{X_{m,K-Q+1}} \end{bmatrix} \quad (15)$$

and

$$\mathbf{V}_x = \begin{bmatrix} 1 & \xi_1 & \cdots & \xi_1^{Q-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \xi_Q & \cdots & \xi_Q^{Q-1} \end{bmatrix}. \quad (16)$$

*Remark 1:*  $\Gamma_m$  is the matrix containing the azimuth angle. This is because  $\mathbf{R}_s$  and  $\text{diag}(\mathbf{A}_{(:,1)}^H(\theta))$  are full-rank diagonal matrices, and  $\mathbf{V}_x$  is a column-full-rank Vandermonde. This will result in  $\mathbf{H}_x = \mathbf{R}_s \text{diag}(\mathbf{A}_{(:,1)}^H(\theta)) \mathbf{V}_x$  a column-full-rank matrix. Moreover, the columns of  $\Gamma_m$  are linear combinations of the columns of  $\mathbf{U}_x^{(m)}$ , we can get  $\text{span}(\Gamma_m) \subseteq \text{span}(\mathbf{U}_x^{(m)})$ . Since  $\mathbf{H}_x$  is a full-rank diagonal matrix, there is  $\mathbf{H}_x^{-1}$  such that  $\mathbf{U}_x^{(m)} = \Gamma_m \mathbf{H}_x^{-1}$ , which leads to  $\text{span}(\mathbf{U}_x^{(m)}) \subseteq \text{span}(\Gamma_m)$ . Therefore,  $\text{span}(\Gamma_m) = \text{span}(\mathbf{U}_x^{(m)})$ , and  $\Gamma_m$  is the matrix containing the azimuth angle  $\varphi$ .

For each subarray's single-snapshot data  $\mathbf{c}_m, m = 1, \dots, M$ , the reconstruction method shown in Fig. 2 is applied. All reconstruction results are combined by columns to form  $\Gamma$ , which is

$$\Gamma = \begin{bmatrix} \Gamma_1 \\ \vdots \\ \Gamma_M \end{bmatrix} = \begin{bmatrix} \mathbf{U}_x^{(1)} \\ \vdots \\ \mathbf{U}_x^{(M)} \end{bmatrix} \mathbf{R}_s \mathbf{H}_x = \mathbf{U}_x \mathbf{R}_s \mathbf{H}_x. \quad (17)$$

After obtaining the matrix  $\Gamma$  about azimuth angle from single-snapshot data  $\mathbf{c}$ , following the concept of root-MUSIC [39], a polynomial is defined as:

$$f(\mu) = \mathbf{p}^H(\mu) (\mathbf{I} - \Gamma_o \Gamma_o^H) \mathbf{p}(\mu) \quad (18)$$

where  $\Gamma_o$  is the matrix obtained by applying the Gram-Schmidt orthogonalization to  $\Gamma$ , and vector  $\mathbf{p}(\mu)$  is determined by the sensor positions and can be partitioned into subarrays as follows:

$$\mathbf{p}(\mu) = [\mathbf{p}_1(\mu), \mathbf{p}_2(\mu), \dots, \mathbf{p}_M(\mu)]^T \quad (19)$$

with  $\mathbf{p}_m(\mu) = [\mu^{X_{m,1}}, \dots, \mu^{X_{m,K-Q+1}}], m = 1, \dots, M$ . When  $\mu = \exp(j2\pi d \cos \varphi / \lambda)$ ,  $\mathbf{p}_m(\mu)$  is the steering vector of the first  $(K-Q+1)$  sensors in the  $m$ th subarray on the  $x$ -axis, and  $\mathbf{p}(\mu)$  is orthogonal to  $(\mathbf{I} - \Gamma_o \Gamma_o^H)$ . By extracting the roots of (18), the azimuth angles can be obtained.

For convenience in computation, (18) is modified as follows:

$$f(\mu) = \mu^{X_{M,K-Q+1}-X_{1,1}} \mathbf{p}^H(\mu) (\mathbf{I} - \Gamma_o \Gamma_o^H) \mathbf{p}(\mu) \quad (20)$$

where the degree of (20) is  $X_{M,K-Q+1} - X_{1,1}$ .

*Theorem 1:* The matrix  $\Gamma_o$  contains the azimuth angle and can be used for azimuth angle estimation in root-MUSIC.

*Proof:* Assuming that the number of sensors in subarray,  $K$ , satisfies  $K - (Q-1) \geq Q$ , so  $\mathbf{U}_x^{(m)}$  is a Vandermonde matrix

with full column rank  $Q$  when the azimuth angles are distinct, expressed as

$$\mathbf{U}_x^{(m)} = \begin{bmatrix} \xi_1^{X_{m,1}} & \xi_2^{X_{m,1}} & \dots & \xi_Q^{X_{m,1}} \\ \xi_1^{X_{m,2}} & \xi_2^{X_{m,2}} & \dots & \xi_Q^{X_{m,2}} \\ \vdots & \vdots & & \vdots \\ \xi_1^{X_{m,K-Q+1}} & \xi_2^{X_{m,K-Q+1}} & \dots & \xi_Q^{X_{m,K-Q+1}} \end{bmatrix}. \quad (21)$$

The rank relationship between  $\mathbf{U}_x^{(m)}$  and  $\mathbf{U}_x$  in (17) is as follows:

$$Q = \text{Rank}(\mathbf{U}_x^{(m)}) \leq \text{Rank}(\mathbf{U}_x) = \text{Rank}\left(\begin{bmatrix} \vdots \\ \mathbf{U}_x^{(m)} \\ \vdots \end{bmatrix}\right) \leq Q. \quad (22)$$

Therefore, it can be concluded that  $\mathbf{U}_x$  is guaranteed to be full column rank with a rank of  $Q$ , and its column vectors are composed of the phase factors of the signals. Since these phase factors are determined by the azimuth angles of the signals,  $\mathbf{U}_x$  contains the azimuth angles. Moreover, since  $\mathbf{H}_x$  in (17) is of full rank, it follows that  $\text{span}(\mathbf{U}_x) = \text{span}(\mathbf{\Gamma}_o)$ , so  $\mathbf{\Gamma}_o$  contains the azimuth angles, and  $(\mathbf{I} - \mathbf{\Gamma}_o \mathbf{\Gamma}_o^H)$  can be used to obtain the noise subspace that does not contain azimuth.

*Theorem 2:* For a subarray composed of  $K$  array sensors, when using matrix  $\mathbf{\Gamma}_o$  for angle estimation, the maximum number of estimable sources is  $Q = \lfloor (K + 1)/2 \rfloor$ , where  $\lfloor x \rfloor$  denotes the floor of  $x$ .

*Proof:* In (13), when the snapshot data received by each subarray is reconstructed into  $\mathbf{\Gamma}_m$ ,  $K - Q + 1$  elements from  $\mathbf{c}_k$  are taken as the column vectors of the new matrix  $\mathbf{\Gamma}_m$ . Therefore,  $(K - Q + 1)$  must be at least equal to 1, implying  $K \geq Q$ . In (20), inspired by the concept of root-MUSIC, we utilize  $(\mathbf{I} - \mathbf{\Gamma}_o \mathbf{\Gamma}_o^H)$  to obtain the noise subspace that does not contain azimuth, where  $\mathbf{\Gamma}_o \mathbf{\Gamma}_o^H$  is a square matrix of size  $M(K - Q + 1)$ . To successfully separate the space containing azimuth from the noise subspace, the following condition must be satisfied:  $M(K - Q + 1) > Q$ . In (22), to ensure that matrix  $\mathbf{U}_x^{(m)}$  has rank  $Q$ ,  $K$  must satisfy  $K - Q + 1 \geq Q$ . In summary, when the subarray consists of  $K$  elements, the maximum number of estimable sources is  $Q = \lfloor (K + 1)/2 \rfloor$ .

By solving (20),  $(X_{M,K-Q+1} - X_{1,1})$  pairs of complex roots will be generated. The phase of the  $Q$  roots  $\mu_1, \dots, \mu_Q$  with the largest amplitude inside the unit circle is selected to solve the DOA estimation of the azimuth  $\hat{\varphi}_q$ , i.e.,

$$\hat{\varphi}_q = \arccos \left[ \frac{\lambda}{2\pi d} \arg(\mu_q) \right], \quad (q) = 1, \dots, Q. \quad (23)$$

*Remark 2:* The exponents in (19) are determined by the sensor positions. With half-wavelength spacing between sensors within each subarray, the exponents in  $\mathbf{p}_m, m = 1, \dots, M$  form a consecutive sequence. However, since the distance between subarrays typically exceeds several half-wavelengths, this results in nonconsecutive exponents in the univariate polynomial of (20). Furthermore, the high-degree term in (20) is determined by the distance between the first and last sensor elements along the coordinate axis, given as  $(X_{M,K-Q+1} - X_{1,1})$ . When either the number of elements per subarray is large

or the intersubarray spacing is significant, this degree can potentially reach the order of hundreds. Consequently, the univariate polynomial in (20) represents a high-degree, lacunary polynomial with complex coefficients. Given the complexity of this root-extracting problem, further discussion will be provided in Section III-C.

### B. Elevation Estimation via Solving High-Degree Lacunary Polynomials

Represent the conjugate transpose of the first row of  $\mathbf{R}_{xz}$  in (8) as  $\mathbf{r} = (\mathbf{R}_{xz}(1,:))^H$  is a linear combination of elevation angle steering vectors, it can be written as

$$\begin{aligned} \mathbf{r} &= \mathbf{A}(\theta) \mathbf{R}_s (\mathbf{A}_{(1,:)}(\varphi))^H \\ &= \mathbf{A}(\theta) [s_1 \xi_Q^{X_{1,1}}, \dots, s_Q \xi_Q^{X_{1,1}}]^H \end{aligned} \quad (24)$$

where  $\mathbf{A}_{(1,:)}(\varphi)$  defines the first row of  $\mathbf{A}(\varphi)$ , and  $\mathbf{r}$  can be regarded as the single snapshot data received by the array along the  $z$ -axis. Similarly as (10),  $\mathbf{r}$  can be written in the following form:

$$\mathbf{r} = [\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N]^T \quad (25)$$

where  $\mathbf{r}_n$  can be treated as the single snapshot data received by the  $n$ th subarray along the  $z$ -axis.

As shown in Fig. 3, the matrix reconstruction is applied to the  $k$ th subarray for the received data  $\mathbf{r}_k$

$$\begin{aligned} \mathbf{Y}_n &= \begin{bmatrix} \mathbf{r}_n(1) & \mathbf{r}_n(2) & \dots & \mathbf{r}_n(Q) \\ \mathbf{r}_n(2) & \mathbf{r}_n(3) & \dots & \mathbf{r}_n(Q+1) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{r}_n(K-Q+1) & \dots & \dots & \mathbf{r}_n(K) \end{bmatrix} \\ &= [\mathbf{r}_{n,1}, \mathbf{r}_{n,2}, \dots, \mathbf{r}_{n,Q}] \end{aligned} \quad (26)$$

where  $\mathbf{r}_{n,q} \in \mathbb{C}^{(K-Q+1) \times 1}$  consists of elements from  $q$ th to  $(K - Q + q)$ th in  $\mathbf{r}_n$ .

Therefore,  $\mathbf{Y}_n$  can be decomposed into the following form:

$$\mathbf{Y}_n = \mathbf{U}_z^{(n)} \mathbf{R}_s \text{diag} \left( (\mathbf{A}_{(1,:)}(\varphi))^H \right) \mathbf{V}_z \quad (27)$$

where  $\mathbf{U}_z^{(n)}$  denotes the steering vector of the first  $(K - Q + 1)$  sensors of the  $k$ th subarray along the  $z$ -axis. Both  $\mathbf{U}_z^{(n)}$  and  $\mathbf{V}_z$  are Vandermonde matrix, which can be expressed as

$$\mathbf{U}_z^{(n)} = \begin{bmatrix} \zeta_1^{Z_{n,1}} & \zeta_2^{Z_{n,1}} & \dots & \zeta_Q^{Z_{n,1}} \\ \vdots & \vdots & & \vdots \\ \zeta_1^{Z_{n,K-Q+1}} & \zeta_2^{Z_{n,K-Q+1}} & \dots & \zeta_Q^{Z_{n,K-Q+1}} \end{bmatrix} \quad (28)$$

and

$$\mathbf{V}_z = \begin{bmatrix} 1 & \zeta_1 & \dots & \zeta_1^{Q-1} \\ \vdots & \vdots & \dots & \vdots \\ 1 & \zeta_Q & \dots & \zeta_Q^{Q-1} \end{bmatrix}. \quad (29)$$

The reconstructed matrices  $\mathbf{Y}_n, n = 1, \dots, N$  are assembled into  $\mathbf{Y}$ , which is

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_N \end{bmatrix} = \begin{bmatrix} \mathbf{U}_z^{(1)} \\ \mathbf{U}_z^{(2)} \\ \vdots \\ \mathbf{U}_z^{(N)} \end{bmatrix} \mathbf{R}_s \text{diag} \left( (\mathbf{A}_{(1,:)}(\varphi))^H \right) \mathbf{V}_z. \quad (30)$$

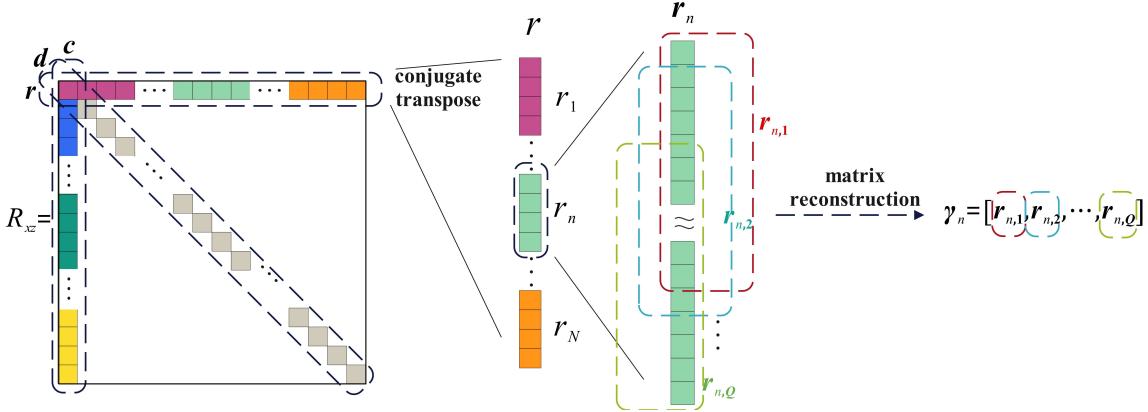


Fig. 3. Matrix reconstruction process.

Similar to (17),  $\Upsilon$  is a matrix containing all elevation angles  $\theta$ .

Define a polynomial, which is

$$f(v) = v^{Z_{N,K-Q+1}-Z_{1,1}} \mathbf{p}^H(v) (\mathbf{I} - \Upsilon_o \Upsilon_o^H) \mathbf{p}(v) \quad (31)$$

where  $\Upsilon_o$  is the result of the Schmidt orthogonalization of  $\Upsilon$ , and  $\mathbf{p}(v)$  is determined by the sensors positions along the  $z$ -axis and can be represented according to subarray division as

$$\mathbf{p}(v) = [\mathbf{p}_1(v), \mathbf{p}_2(v), \dots, \mathbf{p}_N(v)] \quad (32)$$

with  $\mathbf{p}_n(v) = [v^{Z_{n,1}}, \dots, v^{Z_{n,K-Q+1}}], n = 1, \dots, N$ , when  $v = \exp(j2\pi d \cos \theta / \lambda)$ ,  $\mathbf{p}_n(v)$  is the steering vector of the first  $(K - Q + 1)$  sensors in the  $n$ th subarray on the  $z$ -axis, thus  $\mathbf{p}(v)$  is orthogonal to  $(\mathbf{I} - \Upsilon_o \Upsilon_o^H)$ , where the degree of (31) is  $Z_{N,K-Q+1} - Z_{1,1}$ .

Solving (31) will yield  $(Z_{N,K-Q+1} - Z_{1,1})$  pairs of complex roots, the  $Q$  roots with the maximum magnitude inside the unit circle are selected, and extract their phases to estimate the elevation angle, i.e.,

$$\hat{\theta}_q = \arccos \left[ \frac{\lambda}{2\pi d} \arg(v_q) \right], \quad q = 1, \dots, Q. \quad (33)$$

Like (20), (31) is also a high-degree and lacunary univariate polynomial with complex coefficients. The efficient root finding for such a univariate polynomial is further presented in Section III-C.

The steps for constructing the univariate polynomials used to estimate azimuth and elevation angles are presented in Algorithm 1.

### C. Efficient Root Finding for High-Degree Lacunary Polynomials Using Improved DE

The univariate polynomials in (20) and (31) are high-degree, lacunary, and with complex coefficients, exhibiting significant complexity [40]. Galois theory demonstrates that polynomials beyond the fifth degree lack analytical solutions, requiring root approximation through numerical methods. However, the lacunary property may introduce multiple extrema, which can significantly reduce the convergence rate of gradient-based methods such as Newton's method.

---

### Algorithm 1 Polynomials Construction via Cross-Covariance Matrix for DOA Estimation

---

**Input** : cross-covariance matrix  $\mathbf{R}_{xz}$ , number of subarrays along  $x$ - and  $z$ - axes  $M$  and  $N$ , number of sensors in per subarray  $K$ , number of signal sources  $Q$

**Output**: Two univariate polynomials  $f(\mu)$  and  $f(v)$

- 1 Extract the first column  $\mathbf{c} = \mathbf{R}_{xz}(:, 1)$ ;
  - 2 Extract the conjugate transpose of the first row  
 $\mathbf{r} = (\mathbf{R}_{xz}(1, :))^H$ ;
  - 3 **for**  $m = 1$  **to**  $M$  **do**
    - 4 | Form the azimuth-dependent matrix  $\Gamma_m$  by (13).
  - 5 **end**
  - 6 Combine the matrices  $\Gamma_m, m = 1, \dots, M$  by (17);
  - 7 **for**  $n = 1$  **to**  $N$  **do**
    - 8 | Form the azimuth-dependent matrix  $\Upsilon_n$  by (26).
  - 9 **end**
  - 10 Combine the matrices  $\Upsilon_n, n = 1, \dots, N$  by (30);
  - 11 Construct two univariate polynomials for DOA estimation using (20) and (31).
- 

Optimization algorithms, with their inherent advantages such as high convergence and robustness, are naturally suited for the complex root-finding problem above. The accuracy of root finding is directly tied to the estimation precision of 2-D angles. Thus, to ensure root-finding accuracy, the optimization algorithm is employed to compute one root at a time instead of solving all roots simultaneously. The identified roots are subsequently removed via polynomial long division, and this process is iterated until all roots are fully found.

This process requires the algorithm to have strong global search ability, stable convergence performance, and high local search accuracy. There are already many mature optimization algorithms available today, such as DE, genetic algorithm (GA), and particle swarm optimization (PSO) [41]. For the root-finding problem above, DE's mutation has directionality and adaptability, enabling it to perform directed search. Moreover, DE offers multiple mutation strategies with different search ranges and convergence speeds, and its parameters are simple and easy to adjust [42]. Finally, DE's population-based

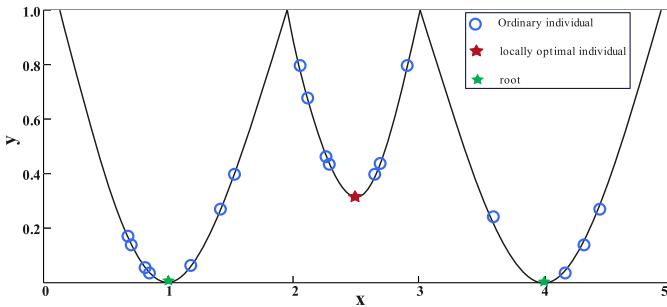


Fig. 4. Local optimum trap.

search does not rely on gradient information and is less likely to get stuck in local minima when the function changes. In contrast, GA's crossover and mutation are random, resulting in lower search efficiency; while PSO converges faster, it easily falls into premature convergence and struggles to find all roots [43].

Therefore, the DE algorithm is selected to solve the roots of the univariate polynomial in this article. However, for the standard DE algorithm, its convergence rate is influenced by the mutation strategy and local optimum traps, as shown in Fig. 4, resulting in the waste of computational resources [44].

This section proposes an improved DE algorithm to extract all roots with high precision of polynomials in (20) and (31). Compared to the standard DE algorithm, the improved DE algorithm uses a hybrid mutation strategy and applies an effective way to deal with local optimal traps to accelerate convergence and reduce computational resources.

The root-finding problem of polynomials is transformed into a single-objective optimization function, defined as

$$\min |f(x)|, \quad x = \mu, v. \quad (34)$$

The improved DE algorithm is used to search for only one root  $x_i$  at a time. Due to the symmetry of the coefficients in the univariate polynomials in (20) and (31), the reciprocal conjugate  $\bar{x}_i = (x^*)^{-1}$  is also a root [21]. Before extracting the next pair of roots, the following equation is applied to eliminate the roots already found, which is expressed as:

$$\bar{f}(x) = f(x)/[(x - x_i)(x - \bar{x}_i)]. \quad (35)$$

The function in (34) initially has many roots clustered around the unit circle in complex plane, with a complex objective function curve that contains multiple local optima, making it increasing the likelihood of getting trapped in local optima, as shown in Fig. 4. As roots are removed, the objective function curve becomes simpler, reducing the probability of local optimal traps. A slow convergence mutation strategy at this stage may lead to unnecessary computational inefficiency. Based on these characteristics, the selection of the mutation strategy is dynamically adjusted based on the number of roots extracted.

In the initial stage of extracting polynomial roots using the improved DE algorithm, due to the larger number of roots to be found and the complex objective function landscape, the mutation strategy DE/rand/1/bin with slow convergence

is employed to facilitate a broad exploration of the solution space, expressed as

$$v_i = x_{r1} + F(x_{r2} - x_{r3}) \quad (36)$$

where  $x_{r1}$ ,  $x_{r2}$ , and  $x_{r3}$  are individuals randomly selected from the population and  $F$  is mutation factor. Preventing premature convergence to local optima. This mutation strategy enhances population diversity, enabling more exhaustive exploration for optimal solutions.

As the known roots are removed, the curve of the objective function becomes simpler; thus, the mutation strategy DE/rand-to-best/1 with faster convergence than DE/rand/1/bin is adopted to improve the search efficiency of the algorithm and locate the remaining roots more quickly, represented as

$$v_i = x_{r1} + F \cdot (x_{\text{best}} - x_{r1}) + F \cdot (x_{r2} - x_{r3}) \quad (37)$$

where  $x_{r1}$ ,  $x_{r2}$ , and  $x_{r3}$  are individuals randomly selected from the population and  $x_{\text{best}}$  is the individual with the best fitness in the current population.

When fewer roots remain to be solved, and the objective function curve is simpler, the local optimum trap is sparser. Therefore, the mutation strategy DE/best/1/bin with even faster convergence is used to efficiently complete the remaining root-solving process, expressed as

$$v_i = x_{\text{best}} + F(x_{r1} - x_{r2}) \quad (38)$$

where  $x_{r1}$  and  $x_{r2}$  are individuals randomly selected from the population, while  $x_{\text{best}}$  is the individual with the best fitness in the current population.

Although dynamically adjusting the mutation strategy has reduced computational resource consumption and the probability of converging to local optima, it remains necessary to detect and handle individuals trapped in local optima. Therefore, a counter is introduced to detect individuals trapped in local optima. Individuals in the population are bound to a counter with a maximum value of  $T$ , recording the number of times the individual has not been replaced by its offspring. The counter is then used to classify individuals into one of three types, i.e.,

- T1) Ordinary individual—where the counter has not reached to the maximum  $T$ . For an ordinary individual, the steps of mutation, crossover, and selection are executed.
- T2) Individual trapped in local optima—where the counter has reached  $T$  but the fitness value  $|f(x_i)|$  has not reached to the precision threshold. For this kind of individual, reinitialization is performed to escape the trap, and the corresponding counter is reset to zero.
- T3) Target individual—where the counter has reached  $T$  and the fitness value has reached the precision threshold. The target individual is output as the root.

The specific steps of the improved DE algorithm are shown in Algorithm 2, and the evolution of individuals in generation  $g$  is shown in Fig. 5.

#### D. Pair Matching

This article estimates 2-D DOAs separately, requiring angle pairing of the obtained estimates  $\hat{\varphi}_q, \hat{\theta}_q, q = 1, \dots, Q$ . Since the number of subarrays on the  $x$ - and  $z$ -axes is different,  $\mathbf{R}_{xz}$

**Algorithm 2** Improved DE Algorithm

---

**Input** : mutation factor  $F$ , crossover probability  $CR$ , maximum generations  $G_{max}$ , accuracy threshold  $\varepsilon$ , maximum count  $T$ ,  $K_1$ ,  $K_2$ ,  $K_3$ , cost function  $f(x)$  and its order  $K$

**Output:** roots  $x_i$  and  $\bar{x}_i$ ,  $i = 1, \dots, K$

- 1 **Initialize**  $N$  population randomly;
- 2 **repeat**
- 3   **repeat**
- 4     **for**  $i = 1$  to  $N$  **do**
- 5       **if**  $counter(i) \leq T$  **then**
- 6         **if**  $K \geq K_1$  **then**
- 7           | Execute (36) mutation strategy
- 8         **else if**  $K_2 \leq K < K_1$  **then**
- 9           | Execute (37) mutation strategy
- 10         **else if**  $0 \leq K < K_2$  **then**
- 11           | Execute (38) mutation strategy
- 12         **end**
- 13         Exchange partial elements with probability  $C_r$  to generate the trial vector  $u_i$ ;
- 14         **if**  $f(u_i^g) \leq f(x_i^g)$  **then**
- 15           |  $x_i^{g+1} = u_i^g$ ;
- 16           |  $counter(i) = counter(i) + 1$ ;
- 17         **else**
- 18           |  $x_i^{g+1} = x_i^g$ ;
- 19         **end**
- 20         **else if**  $counter(i) > T$  and  $f(x_i) > \varepsilon$  **then**
- 21           | Reinitialize the individual  $x_i$  and set  $counter(i) = 0$ ;
- 22         **else if**  $counter(i) > T$  and  $f(x_i) \leq \varepsilon$  **then**
- 23           | Output the best solution  $x_i$ ;
- 24         **end**
- 25          $g \leftarrow (g + 1)$
- 26     **end**
- 27   **until**  $g = G_{max}$  or Output the best solution  $x_i$ ;
- 28   Save the best solution found  $x_i$  and its reciprocal conjugate  $\bar{x}_i$
- 29   **update** the cost function by (35),  $K \leftarrow (K - 2)$ ;
- 30 **until**  $K = 0$ ;

---

is a nonsquare matrix. Extract the main diagonal elements of  $\mathbf{R}_{xz}$  and form column vector  $\mathbf{d}$ , which is

$$\mathbf{d} = \begin{bmatrix} \mathbf{R}_{xz}(1, 1) \\ \mathbf{R}_{xz}(2, 2) \\ \vdots \\ \mathbf{R}_{xz}(L, L) \end{bmatrix}, \quad L = \min(M, N) \quad (39)$$

where  $\mathbf{R}_{xz}(a, b)$  is the element at the  $a$ th row and  $b$ th column of matrix  $\mathbf{R}_{xz}$ .

According to (8),  $\mathbf{d}$  can be decomposed into

$$\mathbf{d} = \begin{cases} \overline{\mathbf{A}}(\varphi) \odot \mathbf{A}^*(\theta) \mathbf{R}_s \vec{\mathbf{1}}, & M \geq N \\ \mathbf{A}(\varphi) \odot \overline{\mathbf{A}}^*(\theta) \mathbf{R}_s \vec{\mathbf{1}}, & M < N \end{cases} \quad (40)$$

where  $\overline{\mathbf{A}}(\varphi)$  consists of the first  $NK$  rows of the steering vector matrix  $\mathbf{A}(\varphi)$ ,  $\overline{\mathbf{A}}(\theta)$  consists of the first  $MK$  rows of the

steering vector matrix  $\mathbf{A}(\theta)$  and  $\mathbf{A}^*(\theta)$  denotes the element-wise complex conjugate of  $\mathbf{A}(\theta)$ .

For clarity, assume  $M \geq N$ . The vector  $\mathbf{d}$  can be partitioned according to subarrays as  $\mathbf{d} = [\mathbf{d}_1^T, \mathbf{d}_2^T, \dots, \mathbf{d}_M^T]^T$ . For each vector  $\mathbf{d}_m, m = 1, \dots, M$ , reconstruct matrix  $\mathbf{P}_m$  using the method similar to (13), which can be expressed as

$$\mathbf{P}_m = \begin{bmatrix} \mathbf{d}_m(1) & \mathbf{d}_m(2) & \cdots & \mathbf{d}_m(Q) \\ \mathbf{d}_m(2) & \mathbf{d}_m(3) & \cdots & \mathbf{d}_m(Q+1) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{d}_m(M-Q+1) & \cdots & \cdots & \mathbf{d}_m(M) \end{bmatrix} \quad (41)$$

where  $\mathbf{d}_m(k)$  denotes the  $k$ th element of  $\mathbf{d}_m$ .

$\mathbf{P}_m$  can be decomposed into the following form:

$$\mathbf{P}_m = \mathbf{U}_p^{(m)} \mathbf{R}_s \mathbf{V}_p \quad (42)$$

where  $\mathbf{U}_p^{(m)}$  and  $\mathbf{V}_p$  can be written as

$$\mathbf{U}_p^{(m)} = \begin{bmatrix} \xi_1^{X_{m,1}} \zeta_1^{-Z_{m,1}} & \cdots & \xi_Q^{X_{m,1}} \zeta_Q^{-Z_{m,1}} \\ \vdots & & \vdots \\ \xi_1^{X_{m,K}} \zeta_1^{-Z_{m,K}} & \cdots & \xi_Q^{X_{m,K}} \zeta_Q^{-Z_{m,K}} \end{bmatrix} \quad (43)$$

and

$$\mathbf{V}_p = \begin{bmatrix} 1 & \cdots & (\xi_1 \zeta_1^{-1})^{Q-1} \\ \vdots & & \vdots \\ 1 & \cdots & (\xi_Q \zeta_Q^{-1})^{Q-1} \end{bmatrix}. \quad (44)$$

Combine the results of all matrix reconstructions, which is

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_M \end{bmatrix} = \begin{bmatrix} \mathbf{U}_p^{(1)} \\ \vdots \\ \mathbf{U}_p^{(M)} \end{bmatrix} \mathbf{R}_s \mathbf{V}_p. \quad (45)$$

Matrix  $\mathbf{P}$  contains the azimuth and elevation angles; 2-D angles can be paired by minimizing the cost function, which is expressed as

$$f(\hat{\varphi}, \hat{\theta}) = \mathbf{a}^H(\hat{\varphi}, \hat{\theta})(\mathbf{I} - \mathbf{P}_o \mathbf{P}_o^H) \mathbf{a}(\hat{\varphi}, \hat{\theta}) \quad (46)$$

where  $\mathbf{P}_o$  is the result of the Schmidt orthogonalization of  $\mathbf{P}$ ,  $\mathbf{a}(\varphi, \theta) = [\mathbf{a}_1^T(\varphi, \theta), \dots, \mathbf{a}_M^T(\varphi, \theta)]^T$  and  $\mathbf{a}_m(\varphi, \theta)$  can be represented as

$$\mathbf{a}_m(\varphi, \theta) = \begin{bmatrix} e^{\frac{2\pi}{\lambda} d(X_{m,1} \cos \varphi - Z_{m,1} \cos \theta)} \\ \vdots \\ e^{\frac{2\pi}{\lambda} d(X_{m,K-Q+1} \cos \varphi - Z_{m,K-Q+1} \cos \theta)} \end{bmatrix}. \quad (47)$$

#### E. Aperture Expansion via Conjugate Transpose and Inversion Operations

Since constructing the angle-related matrix from specific rows or columns of the covariance matrix  $\mathbf{R}_{xz}$  in (13) and (26) leads to partial array aperture loss, this section proposes an aperture extension method for scenarios where the  $x$ - and  $z$ -axes arrays share common elements, thereby enhancing the algorithm's maximum number of estimable sources.

To clarify, suppose the common sensor is the  $\sigma$ th sensor in the first subarray on the  $x$ -axis and the  $\delta$ th sensor in the first subarray on the  $z$ -axis.<sup>1</sup> Represent the  $\delta$ th column of  $\mathbf{R}_{xz}$  as  $\mathbf{c}$ ,

<sup>1</sup>If the sensor is not in the first subarray, the method is still applicable.

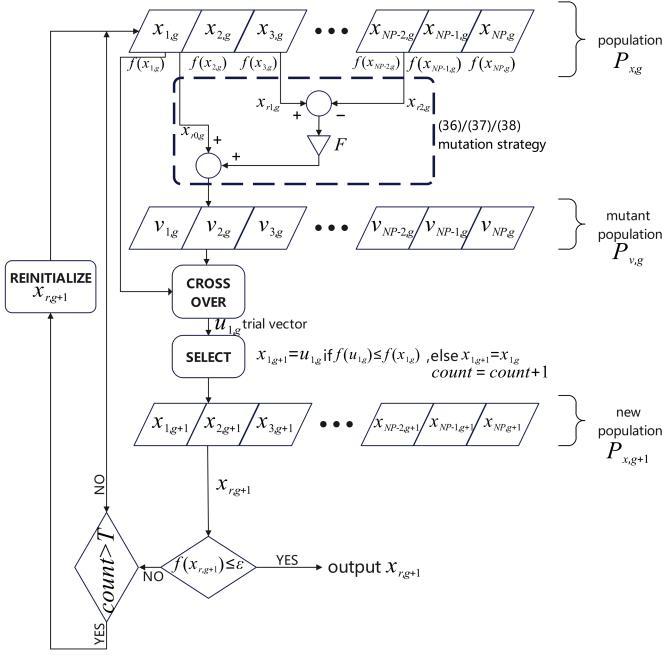
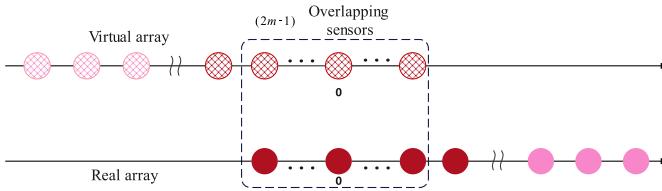
Fig. 5. Evolution of individuals in generation  $g$ .

Fig. 6. Process of virtual aperture expansion.

according to (8),  $\mathbf{c}$  can be expressed as  $\mathbf{c} = \mathbf{A}(\varphi) \mathbf{R}_s \vec{\mathbf{1}}$ , based on  $\mathbf{c}$ , a new column vector  $\bar{\mathbf{c}}$  is obtained, which is

$$\bar{\mathbf{c}} = \mathbf{J}_x(\mathbf{c})^* = \mathbf{J}_x \mathbf{A}^*(\varphi) \mathbf{R}_s \vec{\mathbf{1}} \quad (48)$$

where  $\mathbf{J}_x$  is the anti-diagonal identity matrix of size  $MK$ .

The vector  $\bar{\mathbf{c}}$  corresponds to the snapshot data received by the virtual array on the  $x$ -axis, which is symmetric around the common sensor. The expansion process is illustrated in Fig. 6.

The expansion results in the overlap of  $(2\sigma - 1)$  sensors, so remove the first  $(2\sigma - 1)$  rows of  $\mathbf{c}$  or the last  $(2\sigma - 1)$  rows of  $\bar{\mathbf{c}}$ , and combine  $\mathbf{c}$  and  $\bar{\mathbf{c}}$  to form a column vector  $\mathbf{c}_{\text{ex}}$ , which is expressed as

$$\mathbf{c}_{\text{ex}} = \begin{bmatrix} \bar{\mathbf{c}} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_x \mathbf{A}^*(\varphi) \\ \mathbf{A}(\varphi) \end{bmatrix} \mathbf{R}_s \vec{\mathbf{1}}. \quad (49)$$

$\mathbf{c}_{\text{ex}}$  can be partitioned according to different subarrays into

$$\mathbf{c}_{\text{ex}} = [\bar{\mathbf{c}}_M, \dots, \bar{\mathbf{c}}_2, \mathbf{c}_{1\text{new}}, \mathbf{c}_2, \dots, \mathbf{c}_M] \quad (50)$$

where  $\mathbf{c}_m$  and  $\bar{\mathbf{c}}_m, m = 2, \dots, M$  are the single snapshot data received by the real subarray and the virtual subarray on the  $x$ -axis, respectively, and  $\mathbf{c}_{1\text{new}}$  is the single snapshot data received by the new subarray, formed by removing the overlapping sensors shown in Fig. 6. Similar to (13), matrix reconstruction is performed on  $\mathbf{c}_{\text{ex}}$  to form  $\boldsymbol{\Gamma}$ , which can be

TABLE I  
COMPUTATIONAL COMPLEXITY

Algorithm	Computational Complexity
2D-MUSIC	$O(16M^2T + 64M^3 + \frac{180}{\Delta\theta}\frac{90}{\Delta\phi}(16M^2(4M - Q)))$
DS-ESPRIT	$O(8M^2T + 16M^3 + 4Q^2M)$
Root-MUSIC	$O(8M^2T + 16M^3 + 2(2M + D)^3)$
Proposed	$O(4M^2T + Q^2(2M - Q) + G * NP * (2M + D)^2)$

decomposed into  $\boldsymbol{\Gamma} = \mathbf{U}_x \mathbf{R}_s \mathbf{V}_x$ , with  $\mathbf{U}_x = [\mathbf{b}(\varphi_1), \dots, \mathbf{b}(\varphi_Q)]$ , where  $\mathbf{b}(\varphi_q) = [\bar{\mathbf{b}}_M(\varphi_q), \dots, \mathbf{b}_{1\text{new}}(\varphi_q), \dots, \mathbf{b}_M(\varphi_q)]^T$  with

$$\begin{cases} \bar{\mathbf{b}}_i(\varphi_q) = [\xi_q^{-X_{i,K}}, \dots, \xi_q^{-X_{i,Q}}], & i = 2, \dots, M \\ \mathbf{b}_{1\text{new}}(\varphi_q) = [\xi_q^{-X_{1,K}}, \dots, 1, \dots, \xi_q^{-X_{1,K-Q+1}}] \\ \mathbf{b}_i(\varphi_q) = [\xi_q^{X_{i,1}}, \dots, \xi_q^{X_{i,K-Q+1}}], & i = 2, \dots, M. \end{cases} \quad (51)$$

According to (51), define a polynomial  $f(\mu)$  and find the roots of it to acquire the estimation of the azimuth angle  $\varphi$ , which is

$$f(\mu) = \mu^{X_{M,K-Q+1} + X_{M,K}} \mathbf{p}^H(\mu) (I - \boldsymbol{\Gamma}_o \boldsymbol{\Gamma}_o^H) \mathbf{p}(\mu) \quad (52)$$

where  $\mathbf{p}(\mu) = [\bar{\mathbf{p}}_K(\mu), \dots, \mathbf{p}_{1\text{new}}(\mu), \dots, \mathbf{p}_K(\mu)]^T$ , with

$$\begin{cases} \bar{\mathbf{p}}_i(\mu) = [\mu^{-X_{i,K}}, \dots, \mu^{-X_{i,Q}}], & i = 2, \dots, M \\ \mathbf{p}_{1\text{new}}(\mu) = [\mu^{-X_{1,K}}, \dots, 1, \dots, \mu^{X_{1,K-Q+1}}] \\ \mathbf{p}_i(\mu) = [\mu^{X_{i,1}}, \dots, \mu^{X_{i,K-Q+1}}], & i = 2, \dots, M. \end{cases} \quad (53)$$

Represent the conjugate transpose of the  $\sigma$ th row of  $\mathbf{R}_{xz}$  as  $\mathbf{r}$ , the subsequent procedure of estimating the elevation angle is the same as that for estimating the azimuth angle in this section.

#### F. Computational Complexity Analysis

In real scenarios, using DOA estimation methods to position vehicles requires algorithms to execute as quickly as possible, so analyzing and comparing the computational complexity in terms of multiplication between existing classic DOA methods and our proposed algorithm is necessary. Consider an orthogonal distributed array where each subarray is a ULA with  $M$  sensors, subarray spacing is  $D$  times the half-wavelength, and there are two subarrays along the  $x$ -axis and two along the  $z$ -axis. On this array, the computational complexity of the 2-D-MUSIC method is  $O(16M^2T + 64M^3 + (180/\Delta\theta)(90/\Delta\varphi)(16M^2(4M - Q)))$ , where  $O(16TM^2)$  and  $O(64M^3)$  are the complexities of computing the covariance matrix and performing EVD, respectively. 2-D spectral peak search, has a complexity of  $O((180/\Delta\theta)(90/\Delta\varphi)(16M^2(4M - Q)))$ , where  $\Delta\theta$  and  $\Delta\varphi$  denotes the search step. For the 1-D DS-ESPRIT algorithm, its complexity is  $O(4M^2T + 8M^3 + 2Q^2M)$  with  $O(4M^2T + 8M^3)$  representing the complexity of computing the covariance matrix and EVD; since DS-ESPRIT estimates angles using closed-form solutions, this step further involves  $O(2Q^2M)$  complexity. As 2-D DS-ESPRIT estimates azimuth and elevation angles separately, its total computational

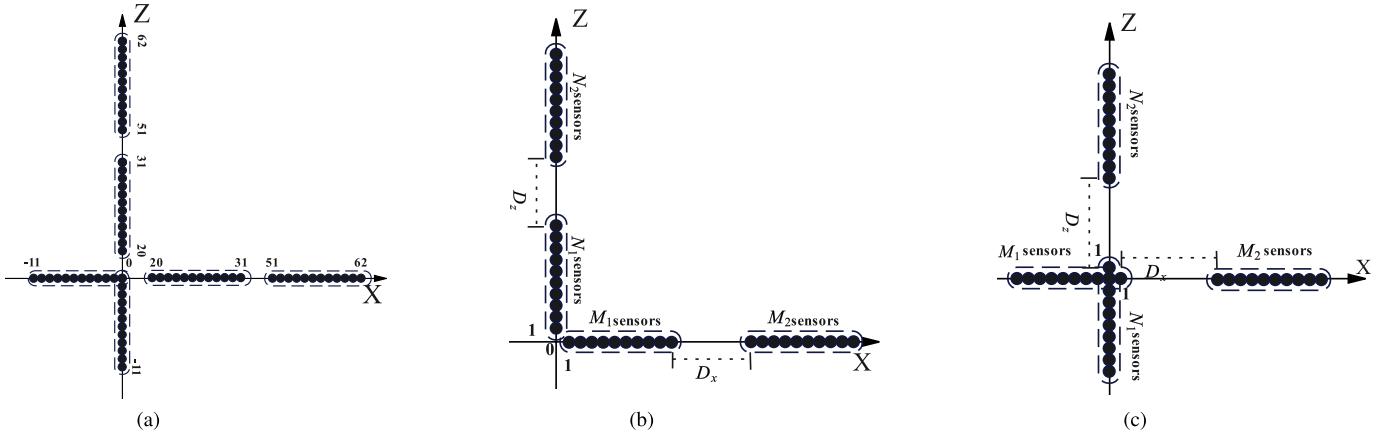


Fig. 7. Distributed array configuration. (a) configure1. (b) configure2. (c) configure3.

complexity becomes  $O(8M^2T + 16M^3 + 4Q^2M)$ . On the same array, the computational complexity of 1-D root-MUSIC is  $O(4M^2T + 8M^3 + (2M + D)^3)$ . The complexity of computing the covariance matrix and EVD is the same as that of DS-ESPRIT. Root-MUSIC finds angle estimates by calculating roots of a univariate polynomial of order  $(2M + D)$ , adding  $O((2M + D)^3)$  complexity for this step, making its total complexity  $O((2M + D)^3)$ . The overall computational complexity of 2-D root-MUSIC is  $O(8M^2T + 16M^3 + 2(2M + D)^3)$  [45]. In the proposed algorithm, we first compute the cross-covariance matrix of data received by sensors on the  $x$ -axis and  $z$ -axis, the complexity of this step is  $O(4M^2T)$ , then perform Schmidt orthogonalization on the matrix in (17), the complexity is  $O(Q^2(2M - Q))$ , and finally use the DE method to find roots of the univariate polynomial in (20). Each DE iteration costs  $O(NP * G)$ , so the total root-finding complexity is  $O(G * NP * (2M + D)^2)$ . Thus, the total computational complexity of our proposed algorithm is  $O(4M^2T + Q^2(2M - Q) + G * NP * (2M + D)^2)$ . The overall computational complexities of all algorithms are presented in Table I.

#### IV. SIMULATION RESULTS

In this section, the effectiveness of the proposed DOA estimation algorithm is demonstrated through several case studies. In addition, a performance comparison in some aspects is conducted between the proposed algorithm, the traditional root-MUSIC algorithm, and the DS-ESPRIT algorithm. The root-mean-square error (RMSE) for assessing the DOA estimation accuracy of azimuth and elevation angles is defined as

$$\text{RMSE} = \sqrt{\frac{1}{JQ} \sum_{j=1}^J \sum_{q=1}^Q \left( (\hat{\phi}_{q,j} - \phi_q)^2 \right)}, \quad \phi = \varphi, \theta \quad (54)$$

where  $J$  represents the number of Monte Carlo experiments and  $Q$  denotes the number of signal sources.  $(\varphi_q, \theta_q)$  represents the true azimuth and elevation angles of the signal, while  $(\hat{\varphi}_{q,j}, \hat{\theta}_{q,j})$  denotes the estimated azimuth and elevation angles of the  $q$ th source in the  $j$ th Monte Carlo trial. In our simulation, experiment 2–5 runs 500 Monte Carlo experiments. When

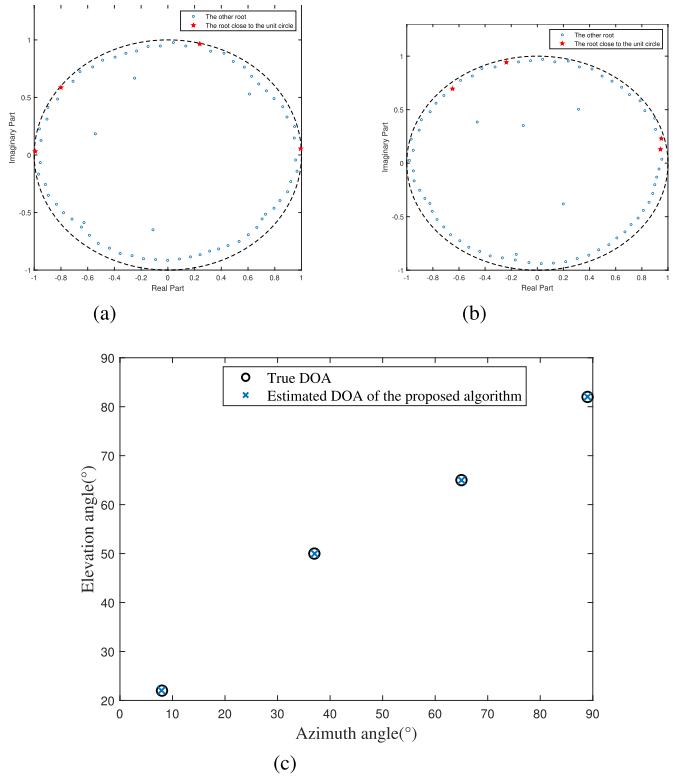


Fig. 8. Simulation result of the proposed algorithm. (a) Roots of the polynomial in 20. (b) Roots of the polynomial in 31. (c) 2-D DOA estimation results.

using DE for root-finding, the precision threshold for the roots is set to  $10^{-4}$ , and the population size is 50. The number of evolutionary generations is selected as 300.  $K_1$ ,  $K_2$ , and  $K_3$  in Algorithm 2 are set to  $30\%K$ ,  $30\%K$ , and  $40\%K$ , respectively.

*Experiment 1:* The first simulation experiment selects the array configuration shown in Fig. 7(a) to validate the effectiveness of the proposed algorithm: the arrays on the  $x$ - and  $z$ -axes are identical, each consisting of three subarrays, with each subarray having 12 sensors, and the distance between the subarrays is 20 times the half-wavelength. Four equi-power incoherent signal sources, with angles of  $[8^\circ, 22^\circ]$ ,  $[37^\circ 50^\circ]$ ,  $[65^\circ, 65^\circ]$ , and  $[89^\circ, 82^\circ]$ , respectively, are incident from

the far-field into the orthogonal distributed array. The signal wavelength is 0.1 m (3 GHz), and the number of snapshots is set to  $T = 500$ , with a signal-to-noise ratio (SNR) of 3 dB. The method proposed in this article is applied to the array's received cross-covariance matrix, and (20) and (31) yield two higher order polynomials with complex coefficients. Fig. 8(a) and (b) show the results of solving these polynomials in the complex plane using the improved DE algorithm. The four roots closest to the unit circle are selected from the two plots, and the 2-D DOA estimation results are obtained by solving them using (23) and (33) and the paired angle results are displayed in Fig. 8(c). This example demonstrates that the proposed algorithm can effectively estimate 2-D angles.

The performance of the proposed algorithm is evaluated under four key parameters: SNR, the number of snapshots, the distance of the subarray, and the number of sensors in each subarray. Consider an orthogonal array configuration as shown in Fig. 7(b), where the array on the  $x$ -axis and the array on the  $z$ -axis have the same configuration: each axis has two subarrays, and the distance between subarrays is  $D$  times the half-wavelength. The number of sensors in each subarray is the same. Three noncoherent signals with equal power and a wavelength of 1 m are incident on the array from directions [35°, 27°], [63°, 52°], and [88°, 81°]. In this section, the RMSE of the proposed DOA estimation method is plotted, along with a comparison to the RMSE of the DS-ESPRIT algorithm, the root-MUSIC algorithm, and CRB [46]. This comparison includes the RMSE of the proposed algorithm and the traditional root-MUSIC algorithm for both DA and ULA with the same aperture. Since the algorithm proposed in this article estimates the 2-D DOA separately, the RMSEs for estimating the azimuth and elevation angles are plotted separately to more effectively demonstrate the algorithm's performance.

**Experiment 2:** The second example tests the performance of the proposed method in terms of SNR. The SNR varies in the range from -4 to 20 dB, with 500 snapshots, a subarray spacing  $D$  of 30 times the half-wavelength, and ten sensors per subarray. As shown in Fig. 9, on the DA, the performance of the proposed algorithm outperforms the traditional root-MUSIC algorithm in the SNR range of -4 to 20 dB, thanks to the reduced additive noise from the cross-covariance matrix. At low SNR, angle ambiguity is more pronounced, but when the SNR exceeds 2 dB, both the proposed algorithm and root-MUSIC show no angle ambiguity. Moreover, the proposed algorithm outperforms traditional root-MUSIC in both DA and ULA, as well as providing better performance than the coarse estimate of DS-ESPRIT. On a ULA with the same aperture as DA, the proposed algorithm also outperforms the traditional root-MUSIC algorithm. Although the proposed algorithm is suboptimal in Fig. 9, it approaches the ESPRIT's estimation accuracy with merely a single row and column of the covariance matrix.

**Experiment 3:** The third experiment compares the DOA performance of the proposed algorithm and traditional algorithms under varying subarray distances. The range of subarray distances is from 50 to 120 times the half-wavelength, with the SNR set to 10 dB, and other parameters are the same

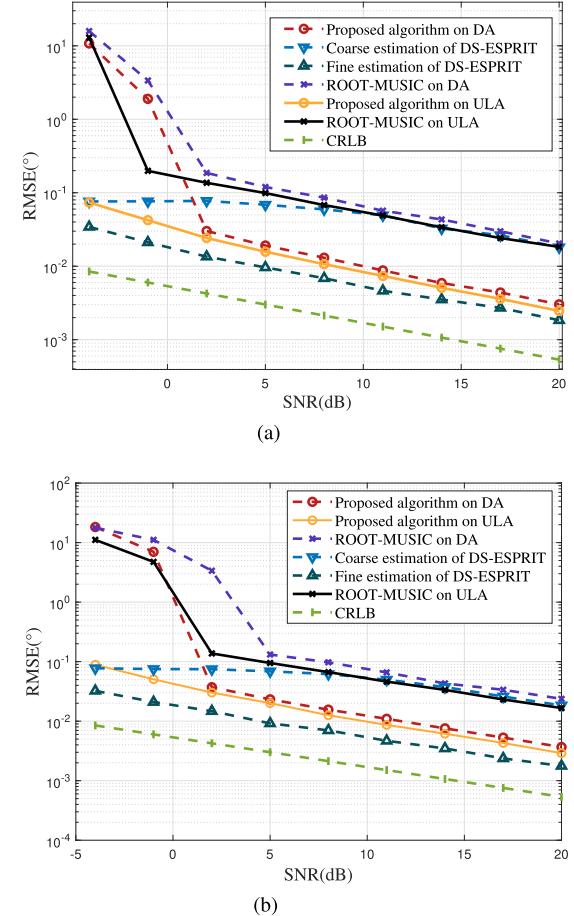
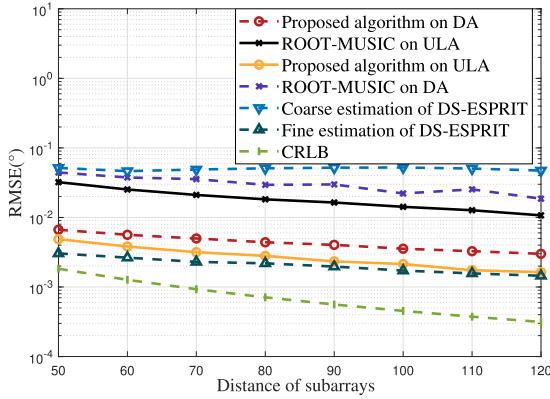


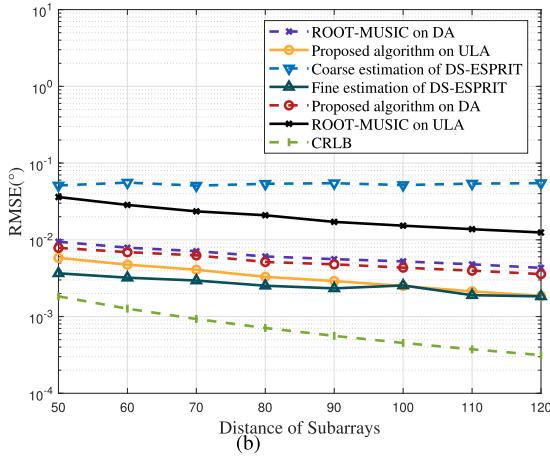
Fig. 9. Estimation performance versus SNR. (a) RMSE of azimuth angle. (b) RMSE of elevation angle.

as in the first example. As shown in Fig. 10, the accuracy of the proposed algorithm increases with the increase in array spacing. Furthermore, the performance of the proposed algorithm is superior to that of the traditional root-MUSIC and the coarse estimate of DS-ESPRIT, both in DA and ULA. In DA, the RMSE of the proposed algorithm is on the same order of magnitude as that of a fine estimate of DS-ESPRIT, which uses all data of the covariance matrix.

**Experiment 4:** In the fourth simulation experiment, the performance of the proposed algorithm is verified in terms of the number of snapshots. Consider the orthogonal array configuration shown in Fig. 7(b), with the SNR set to 10 dB and the number of snapshots varying from 200 to 1400. Other parameters are set the same as in the first example. As can be seen from the results in Fig. 11, at the same number of snapshots, the accuracy of root-MUSIC and coarse estimates exhibits comparable performance. However, the proposed algorithm significantly outperforms the traditional root-MUSIC and the coarse estimate of DS-ESPRIT at the same snapshot count. While the accuracy of the proposed algorithm is slightly lower than that of the precise estimate of DS-ESPRIT, it not only reduces the impact of additive noise but also uses only one row or column of the covariance matrix, leading to a much lower computational complexity compared to the DS-ESPRIT algorithm.



(a)

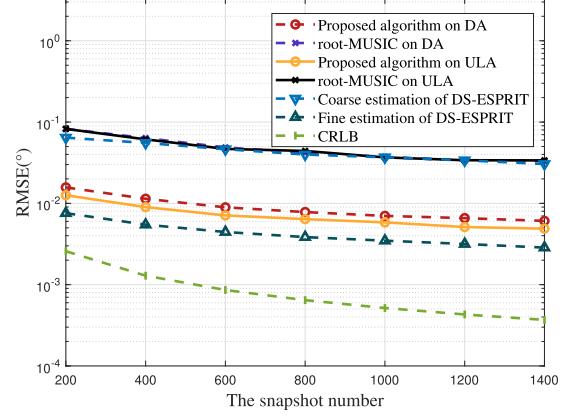


(b)

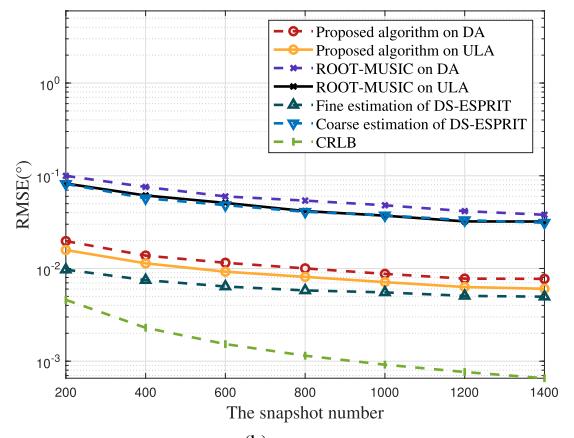
Fig. 10. Estimation performance versus the distance of subarrays. (a) RMSE of azimuth angle. (b) RMSE of elevation angle.

**Experiment 5:** The fifth experiment compares the DOA performance of the proposed algorithm and traditional algorithms under varying numbers of sensors within each subarray. Using the same configuration as in the second example, the number of sensors in each subarray is varied within the range of 10–30, while keeping the SNR fixed at 10 dB. The other parameters are the same as those in the second example. As shown in Fig. 12, when the number of sensors per subarray exceeds 18, the performance of the proposed algorithm on both the distributed array and the uniform linear array becomes very similar. This indicates that the distributed array can achieve the same estimation accuracy as the uniform linear array with fewer sensors. Moreover, the estimation accuracy of the proposed algorithm significantly outperforms that of the traditional root-MUSIC algorithm.

**Experiment 6:** The sixth experiment tests the performance of the proposed method and several classical methods under different angle separations. Assume an array configuration as shown in Fig. 7(b), where each subarray is a ULA with ten sensors. Parameters are set as  $D_x = D_z = 5$ , SNR is fixed at 10 dB, and the number of snapshots is 500. Two signals are incident on the array from  $[35^\circ, 27^\circ]$  and  $[35^\circ + \Delta, 27^\circ + \Delta]$ , respectively. The RMSE of the algorithm is shown in Fig. 13. When SNR is fixed at 10 dB, the estimation error of the algorithm gradually decreases as the angle



(a)



(b)

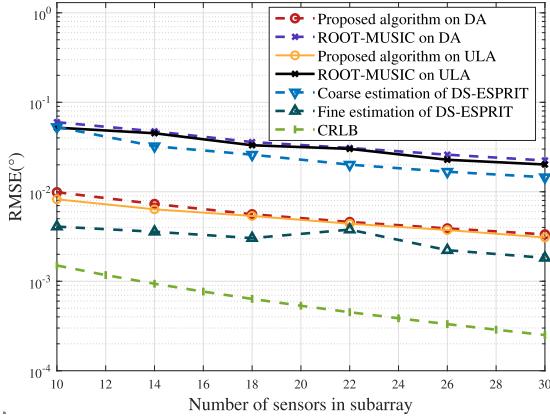
Fig. 11. Estimation performance versus the number of snapshots. (a) RMSE of azimuth angle. (b) RMSE of elevation angle.

separation between sources increases. The proposed algorithm achieves an approximate error  $1^\circ$  when the angle separation is  $3.6^\circ$ , and even reaches an estimation error  $0.1^\circ$  when the angle separation is  $6^\circ$ . DS-ESPRIT and root-MUSIC exhibit higher accuracy than our algorithm at small angle separations. This is because, compared to these two algorithms, our algorithm employs a low-complexity matrix reconstruction method to obtain the approximate signal subspace  $\mathbf{U}_a$ , instead of using EVD to derive the precise signal subspace  $\mathbf{U}_f$ . This matrix reconstruction approach suffers from greater information loss when the angle separation is small, leading to a degradation in estimation accuracy.

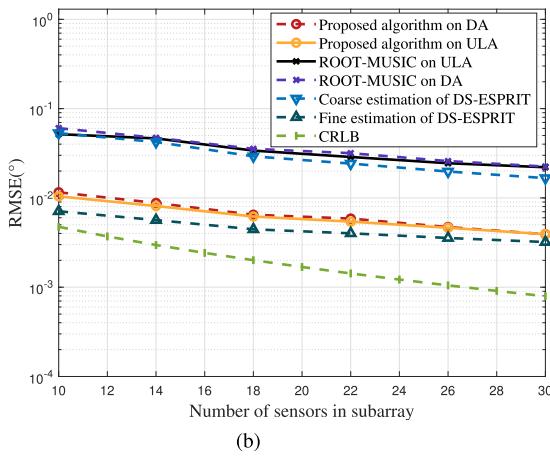
To evaluate the performance observed in Fig. 13, an experiment on subspace similarity is further conducted in this experiment. The maximum principal angle (MPA) is employed to represent the similarity between  $\mathbf{U}_f$  and  $\mathbf{U}_a$ , which measures the largest angle deviation between the least similar directions of the two subspaces [47]. Specifically, the matrix  $\mathbf{C} = \mathbf{U}_f^H \mathbf{U}_a$  is defined, and singular value decomposition (SVD) is performed on  $\mathbf{C}$ , expressed as

$$\mathbf{C} = Q \Sigma V^H, \quad \Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p). \quad (55)$$

The corresponding principal angles  $\theta_i = \arccos(\sigma_i)$  and  $\theta_{\max} = \max_i \theta_i$  can then be obtained. The variation of the MPA with respect to the angle separation is shown in Fig. 14.



(a)



(b)

Fig. 12. Estimation performance versus the number of sensors in the subarray.  
(a) RMSE of azimuth angle. (b) RMSE of elevation angle.

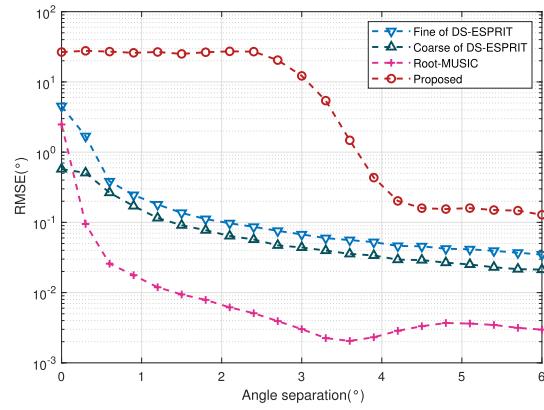


Fig. 13. Estimation performance versus angle separation.

As the angle separation increases, the subspaces  $\mathbf{U}_f$  and  $\mathbf{U}_a$  become more similar, which is consistent with the trend of the proposed algorithm and the root-MUSIC approach gradually observed in Fig. 13.

**Experiment 7:** The seventh simulation experiment compares the proposed algorithm with its extended version, demonstrating that the extension can simultaneously estimate the DOA of more signals. In this example, five equal-power far-field signals are incident from  $[10^\circ, 5^\circ]$ ,  $[30^\circ, 25^\circ]$ ,  $[50^\circ, 45^\circ]$ ,

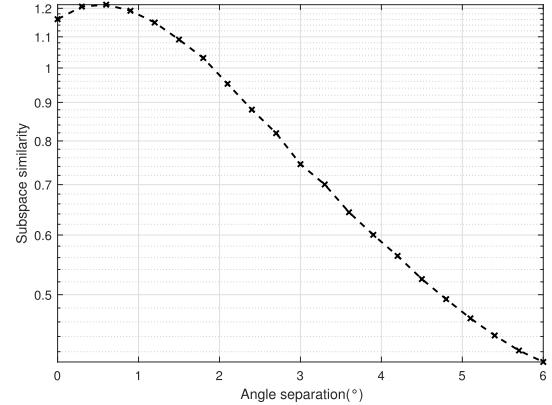


Fig. 14. MPA versus angle separation.

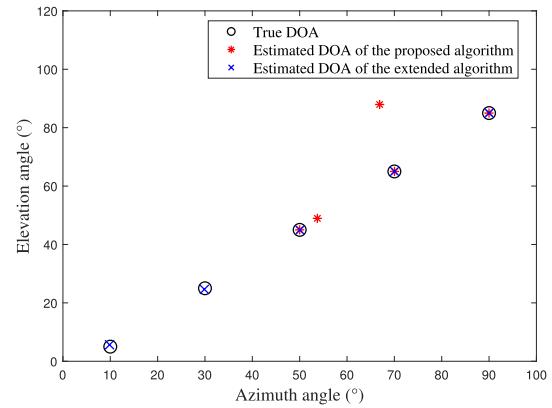


Fig. 15. Proposed algorithm versus the extend algorithm.

$[70^\circ, 65^\circ]$ , and  $[90^\circ, 85^\circ]$  to the array shown in the figure. The SNR is set to 10 dB, the wavelength is 0.1 m, and the array configuration is set as  $M_1 = N_1 = M_2 = N_2 = 10$ ,  $D = D_x = D_z = 30$ . From Section III-C, it can be seen that the 9th column of  $R_{xz}$  data can be approximated as the single snapshot data received by the array sensors located at positions  $[-8, \dots, 0, 1, 31, \dots, 40] * \lambda/2$  along the  $x$ -axis. After applying conjugate symmetry, it can be viewed as the data received by the array sensors located at positions  $[-40, \dots, -31, -1, 0, \dots, 8] * \lambda/2$ . After removing duplicate data from the array sensors, the data is concatenated in order of their positions. The  $\mathbf{c}_{ex}$  in (50) can thus be interpreted as the data received by the new array at positions  $[-40, \dots, -31, -8, \dots, 0, \dots, 8, 31, \dots, 40] * \lambda/2$ . Two univariate polynomials are set according to the positions of the new sensors, and the 2-D DOA is solved as shown in Fig. 15. Under the simulation conditions of this example, the proposed algorithm can estimate only three signals, while the extended algorithm can effectively estimate the DOA of all the signals. The array before the extension can estimate at most five signals under ideal conditions without noise, whereas after the extension, it can estimate up to eight signals.

**Experiment 8:** The eighth experiment compares the angle measurement performance of the algorithm before and after aperture extension. On the same array used in Experiment 6, the signal SNR ranges from  $-4$  to 20 dB. The RMSE results

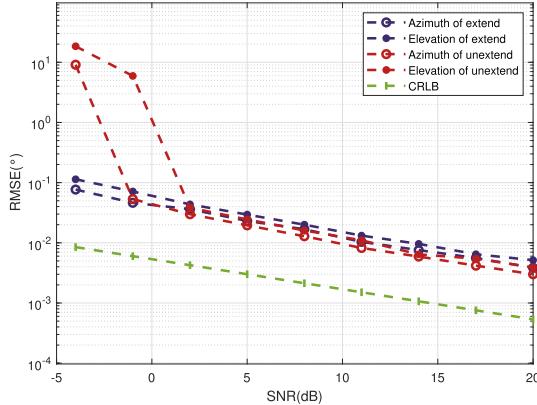


Fig. 16. Comparison before and after algorithm extension.

TABLE II  
COMPUTATION TIME COMPARISON

Algorithm	$D = 5$	$D = 10$	$D = 20$	$D = 30$
Root-MUSIC	0.3244s	0.3390s	0.3511s	0.3862s
DS-ESPRIT	0.0044s	0.0046s	0.0043s	0.0041s
2D-MUSIC	19.1755s	19.3803s	19.3038s	19.2170s
Proposed	0.1286s	0.1290s	0.1350s	0.1728s

are shown in Fig. 16. At low SNR, the extended algorithm has a smaller angular error than the unextended one—this is because it can estimate angles for more signals. However, at high SNR, the extended algorithm's accuracy is slightly lower than the unextended version. The reason is that the virtual array's data is derived from symmetric real sensors, which introduces small data errors.

*Experiment 9:* The ninth experiment compares the execution time of the proposed algorithm with existing classical algorithms. In fact, the computational complexities of the algorithms should first be clarified, as presented in Table I. The computer's CPU configuration is Intel<sup>2</sup> Core<sup>3</sup> i5-10400, and the array is the same as that in the third experiment. When the subarray distance increases from 5 to 30 times the half-wavelength, the computation times of algorithms are shown in Table II. The execution speed of the proposed algorithm is far superior to that of the 2-D-MUSIC algorithm and is three times faster than the root-MUSIC algorithm. Although its execution speed is slower than that of the DS-ESPRIT algorithm, the DS-ESPRIT algorithm is only applicable to two subarrays and cannot increase the number of subarrays further to achieve higher estimation accuracy.

## V. CONCLUSION

This article presents a novel vehicle positioning method based on 2-D DOA estimation using orthogonal distributed arrays, enhanced by an improved DE algorithm, which offers greater practical significance for vehicle positioning in IoV systems. Since applying the root-MUSIC algorithm

to distributed arrays results in high-order polynomials with missing terms, the proposed method employs an improved DE approach to efficiently solve these lacunary polynomials. Moreover, the signal subspace can be recovered from only a single row or column of the covariance matrix, thereby reducing computational complexity. For crossed arrays, an aperture extension technique is introduced to increase the number of detectable sources, supported by theoretical analysis. In addition, the proposed algorithm exhibits strong scalability and can be readily extended to multisubarray configurations, enabling flexible deployment in large-scale distributed array systems. Simulation results demonstrate that the proposed method accurately extracts polynomial roots and outperforms existing DOA estimation algorithms in terms of estimation accuracy and computational efficiency.

## REFERENCES

- [1] D. Kumar and N. Muhammad, "A survey on localization for autonomous vehicles," *IEEE Access*, vol. 11, pp. 115865–115883, 2023.
- [2] J. Li and S.-H. Hwang, "Improved GNSS positioning schemes in urban canyon environments," *IEEE Access*, vol. 13, pp. 112354–112367, 2025.
- [3] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular SLAM in dynamic environments," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2013, pp. 209–218.
- [4] T.-L. Kim, J.-S. Lee, and T.-H. Park, "Fusing LiDAR, radar, and camera using extended Kalman filter for estimating the forward position of vehicles," in *Proc. IEEE Int. Conf. Cybern. Intell. Syst. (CIS) IEEE Conf. Robot., Autom. Mechatronics (RAM)*, Nov. 2019, pp. 374–379.
- [5] X. Ma, B. Hao, H. Zhang, and P. Wan, "Semidefinite relaxation for source localization by TOA in unsynchronized networks," *IEEE Signal Process. Lett.*, vol. 29, pp. 622–626, 2022.
- [6] Y. Liu, C. Chen, Y. Wang, and C. Liu, "Range-independent TDOA localization using stepwise accuracy enhancement under speed uncertainty," *IEEE Signal Process. Lett.*, vol. 30, pp. 1372–1376, 2023.
- [7] R. W. Ouyang, A. K.-S. Wong, and C.-T. Lea, "Received signal strength-based wireless localization via semidefinite programming: Non-cooperative and cooperative schemes," *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1307–1318, Mar. 2010.
- [8] J. Fang, H. Chen, W. Liu, S. Yang, C. Yuen, and H. C. So, "Three-dimensional localization of mixed near-field and far-field sources based on a unified exact propagation model," *IEEE Trans. Signal Process.*, vol. 73, pp. 245–258, 2025.
- [9] H. Xu, W. Liu, M. Jin, and Y. Tian, "Positioning and contour extraction of autonomous vehicles based on enhanced DOA estimation by large-scale arrays," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11792–11803, Jul. 2023.
- [10] M. Ammous and S. Valaee, "Cooperative positioning in vehicular networks using angle of arrival estimation through mmWave," in *Proc. IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.
- [11] H. Chen et al., "Near-field target localization for EMVS-MIMO radar with arbitrary configuration," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 4, pp. 5406–5417, Aug. 2024.
- [12] M. Lin et al., "Situational awareness based resource allocation for multi-target tracking in distributed radar network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Hyderabad, India, Apr. 2025, pp. 1–5.
- [13] K. Shamaei and Z. M. Kassas, "A joint TOA and DOA acquisition and tracking approach for positioning with LTE signals," *IEEE Trans. Signal Process.*, vol. 69, pp. 2689–2705, 2021.
- [14] A.-A. Saucan, T. Chonavel, C. Sintes, and J.-M. Le Caillec, "CPHD-DOA tracking of multiple extended sonar targets in impulsive environments," *IEEE Trans. Signal Process.*, vol. 64, no. 5, pp. 1147–1160, Mar. 2016.
- [15] A. Kangas and T. Wigren, "Angle of arrival localization in LTE using MIMO pre-coder index feedback," *IEEE Commun. Lett.*, vol. 17, no. 8, pp. 1584–1587, Aug. 2013.
- [16] P. Heidenreich, A. M. Zoubir, and M. Rubsamen, "Joint 2-D DOA estimation and phase calibration for uniform rectangular arrays," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4683–4693, Sep. 2012.

<sup>2</sup>Registered trademark.<sup>3</sup>Trademarked.

- [17] N. Yilmazer and T. K. Sarkar, "2-D unitary matrix pencil method for efficient direction of arrival estimation," *Digit. Signal Process.*, vol. 16, no. 6, pp. 767–781, Nov. 2006.
- [18] N. Tayem and H. M. Kwon, "L-shape 2-dimensional arrival angle estimation with propagator method," *IEEE Trans. Antennas Propag.*, vol. 53, no. 5, pp. 1622–1630, May 2005.
- [19] H. Chen, J. Li, S. Yang, W. Liu, Y. C. Eldar, and C. Yuen, "Near-field source localization in 3-D using two parallel centrally symmetric unfold coprime array," *IEEE Trans. Wireless Commun.*, vol. 24, no. 6, pp. 4738–4749, Jun. 2025.
- [20] X. Dai, X. Zhang, and Y. Wang, "Extended DOA-matrix method for DOA estimation via two parallel linear arrays," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1981–1984, Nov. 2019.
- [21] Z. Zhang, W. Wang, Y. Huang, and S. Liu, "Decoupled 2-D direction of arrival estimation in L-shaped array," *IEEE Commun. Lett.*, vol. 21, no. 9, pp. 1989–1992, Sep. 2017.
- [22] Z. Zhang, Z. Shi, and Y. Gu, "Ziv-zakai bound for DOAs estimation," *IEEE Trans. Signal Process.*, vol. 71, pp. 136–149, 2023.
- [23] Z. Zhang, Z. Shi, C. Shao, J. Chen, M. S. Greco, and F. Gini, "Ziv-Zakai bound for 2D-DOAs estimation," *IEEE Trans. Signal Process.*, vol. 72, pp. 2483–2497, 2024.
- [24] Z. Jing, B. Cao, X. Meng, X. Li, F. Yan, and M. Jin, "Noniterative DOA estimation in time-modulated array under unidirectional phase center motion," *IEEE Antennas Wireless Propag. Lett.*, vol. 23, pp. 2416–2420, 2024.
- [25] P. Chen, Z. Chen, L. Liu, Y. Chen, and X. Wang, "SDOA-Net: An efficient deep-learning-based DOA estimation network for imperfect array," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–12, 2024.
- [26] F. Wen, J. Wang, J. Shi, and G. Gui, "Auxiliary vehicle positioning based on robust DOA estimation with unknown mutual coupling," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5521–5532, Jun. 2020.
- [27] Y. Jin, D. He, S. Wei, and W. Yu, "Off-grid DOA estimation method based on sparse Bayesian learning with clustered structural-aware prior information," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 5469–5483, Apr. 2024.
- [28] M. Lin et al., "Multi-dimensional resource management optimization strategy for high-resolution multi-target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, early access, Jul. 14, 2025, doi: [10.1109/TAES.2025.3588819](https://doi.org/10.1109/TAES.2025.3588819).
- [29] M. Haardt, M. D. Zoltowski, C. P. Mathews, and J. Nossek, "2D unitary ESPRIT for efficient 2D parameter estimation," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 3, 1995, pp. 2096–2099.
- [30] B. Chandna and A. Mahadar, "2D beamspace ESPRIT with spatial smoothing," in *IEEE Antennas Propag. Soc. Int. Symp. Dig. Antennas, Gateways Global Netw.*, vol. 1, Jul. 1998, pp. 219–222.
- [31] J. Li, P. Ma, X. Zhang, and G. Zhao, "Improved DFT algorithm for 2D DOA estimation based on 1D nested array motion," *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 1953–1956, Sep. 2020.
- [32] W. Suleiman, P. Parvazi, M. Pesavento, and A. M. Zoubir, "Non-coherent direction-of-arrival estimation using partly calibrated arrays," *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5776–5788, Nov. 2018.
- [33] Y. D. Zhang et al., "DOA estimation exploiting distributed array with arbitrary subarray orientations," in *Proc. IEEE Radar Conf. (Radar-Conf)*, May 2024, pp. 1–5.
- [34] J. Zhang et al., "Design of ultrawideband energy selective surface based on triple-layer structure and semiconductor for hirf prevention," *IEEE Trans. Antennas Propag.*, vol. 73, no. 11, pp. 9619–9624, Nov. 2025.
- [35] H. Wang, L. Wan, M. Dong, K. Ota, and X. Wang, "Assistant vehicle localization based on three collaborative base stations via SBL-based robust DOA estimation," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5766–5777, Jun. 2019.
- [36] C. Li, Q. Wang, H. Chen, and L. Teng, "1-bit DOA estimation using expectation-maximization generalized approximate message passing with two L-shaped arrays," *IEEE Commun. Lett.*, vol. 25, no. 8, pp. 2554–2558, Aug. 2021.
- [37] S. R. Pavel and Y. D. Zhang, "Neural network-based compression framework for DOA estimation exploiting distributed array," in *Proc. ICASSP - IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 4943–4947.
- [38] G. Wang, J. Xin, N. Zheng, and A. Sano, "Computationally efficient subspace-based method for two-dimensional direction estimation with L-shaped array," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3197–3212, Jul. 2011.
- [39] H. Zishu, L. Yang, and X. Jingheng, "A modified root-music algorithm for signal DOA estimation," *J. Syst. Eng. Electron.*, vol. 10, no. 4, pp. 42–47, Dec. 1999.
- [40] J. Solymosi, E. P. White, and C. H. Yip, "On the number of distinct roots of a lacunary polynomial over finite fields," 2020, *arXiv:2008.09962*.
- [41] S. Sunaina, B. D. Shivahare, V. Kumar, S. K. Gupta, P. Singh, and M. Diwakar, "Metaheuristic optimization algorithms and recent applications: A comprehensive survey," in *Proc. Int. Conf. Comput. Intell., Commun. Technol. Netw. (CICTN)*, Apr. 2023, pp. 506–511.
- [42] A. Trivedi, K. Sanyal, P. Verma, and D. Srinivasan, "A unified differential evolution algorithm for constrained optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1231–1238.
- [43] A. M. Akugbe and A. A. Oluyinka, "On the performance of linear decreasing inertia weight particle swarm optimization for global optimization," *Scientific World J.*, vol. 2013, no. 1, Jan. 2013, Art. no. 860289.
- [44] Y. Shi, H. Gao, and D. Wu, "An improved differential evolution algorithm with novel mutation strategy," in *Proc. IEEE Symp. Differ. Evol. (SDE)*, Dec. 2014, pp. 1–8.
- [45] F.-G. Yan, L. Shuai, J. Wang, J. Shi, and M. Jin, "Real-valued root-MUSIC for DOA estimation with reduced-dimension EVD/SVD computation," *Signal Process.*, vol. 152, pp. 1–12, Nov. 2018.
- [46] T. Tirer and O. Bialer, "A method for reducing the performance gap between non-coherent and coherent sub-arrays," *IEEE Trans. Signal Process.*, vol. 68, pp. 3358–3370, 2020.
- [47] A. V. Knyazev and M. E. Argentati, "Principal angles between subspaces in an A-based scalar product: Algorithms and perturbation estimates," *SIAM J. Sci. Comput.*, vol. 23, no. 6, pp. 2008–2040, Jan. 2002.