DoA Estimation for Lens Antenna Array via Root-MUSIC, Outlier Detection, and Clustering

ZHENGWEI NI¹, (Member, IEEE), YINAN LUO¹, MEHUL MOTANI², (Fellow, IEEE), AND CHUANHUANG LI¹

¹School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China
²Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583

Corresponding author: Zhengwei Ni (zhengwei.ni@zjgsu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61871468, and in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY18F010006.

ABSTRACT

Lens antenna arrays have become attractive for mmWave MIMO systems due to their energy-focusing and path-division properties. However, when the signals cannot be well differentiated at different array elements due to their similar incident angles, we cannot estimate their DoAs separately. In this paper, we present a DoA estimation algorithm for this situation. Our algorithm has three main steps: a special version of root multiple signal classification (MUSIC) for lens antenna arrays, outlier detection, and clustering. Numerical results show that our algorithm can achieve good performance even with a large number of signal sources, a large number of array elements, and a small number of snapshots.

INDEX TERMS

DoA estimation, lens antenna array, root-MUSIC, outlier detection, clustering.

I. INTRODUCTION

A lens antenna array is an advanced antenna design for millimeter wave (mmWave) multiple-input multiple-output (MIMO) systems [1]–[3]. As Fig. 1 shows, it is composed of two main components: an electromagnetic (EM) lens and a matching antenna array. The EM lens can focus the energy of the incident EM wave. After passing through the EM lens, the energy of the incident EM wave is spatially redistributed following a ‘sinc’ function with its center in the focal point. Thus, most of the energy is focused onto the close vicinity of the focal point. All the possible focal points form a focal arc and matching antenna elements are placed on it separately. Since the position of the focal point is determined by the direction-of-arrival (DoA) of the incident EM wave, the waves which come from sufficiently separated directions have distinct energy distributions on the lens antenna array. By the energy-focusing property, the signals with separated DoAs can be regarded as received by different array elements, as Fig. 1 shows. Thus, we can easily distinguish these signals and handle them separately. Because most of the energy is received by a small group of array elements, we can only connect them to radio frequency (RF) chains for further processing and ignore the rest. This makes a lens antenna array a good candidate for mmWave MIMO systems because both the hardware complexity and the power consumption are significantly reduced.

However, when the incident angles of the signals are similar and the groups of array elements receiving these signals overlap, as Fig. 2 shows, we cannot handle these signals separately. Specifically, we cannot use simple methods (such as algorithm A1 provided in Section VI) to estimate their DoAs independently. This situation can arise when the devices are deployed densely and communicate simultaneously.

In this paper, we provide a DoA estimation algorithm for the scenario that the incident angles of the signals are similar, as described above. Our algorithm has three main steps: (i) a special version of root multiple signal classification (MUSIC) [4] for lens antenna array, (ii) outlier detection, and (iii) clustering. The numerical results show that our algorithm can achieve a good performance even with a large number of signal sources (more than 10), a large number of array elements (more than 40), and a small number of snapshots (no more than 10).

We note that DoA estimation is considered in some papers studying channel estimation [1]–[3]. However, these papers study strategies which have explicit training overhead. This is not the case in our paper, in which the receiver does not know the transmitted symbols and treats them as Gaussian random variables, which is true for applications like localization.
FIGURE 1. For lens antenna array, the signals with separated DoAs can be regarded as received by different array elements.

II. LENS ANTENNA ARRAY MODEL

The lens antenna array is described in [1]–[3]. As shown in Fig. 1 and Fig. 2, it consists of a planar EM lens and an antenna array. The EM lens is placed on the y-z plane and its size is \( D_y \times D_z \). The thickness of this lens is negligible compared with the focal length. The center of this lens is placed at the origin. We only consider azimuth DoAs, so the focal arc of the lens is on the x-y plane. The lens is well designed so that the focal arc is a semicircle around the center of the lens. The focal length of the lens, as well as the radius of the semicircle, is denoted as \( F \).

To receive the focused energy, the array elements should be placed on the focal arc. Let \( \Phi (0 \leq \Phi \leq \pi) \) and \( \lambda \) denote the azimuth coverage angle by the lens antenna array and the signal wavelength, respectively. According to [3], the array elements are placed based on the angle resolution of the lens such that

\[
\sin \varphi_m = \frac{m\lambda}{D_y}, \quad m \in \mathcal{M},
\]

where \( m \) is the index of antenna, \( \mathcal{M} = \{0, \pm 1, \ldots, \pm M\} \) is the set of indices, \( \tilde{M} = \left[ D_y \sin (\Phi/2)/\lambda \right] \), and \( \varphi_m \in [-\Phi/2, \Phi/2] \) is the angle of antenna \( m \) relative to the x-axis. Here \([a]\) means the largest integer no greater that \( a \). The number of array elements, denoted by \( M \), is given as \( M = 1 + 2\tilde{M} \).

According to [1, Lemma 1], for the lens antenna array described above, the array response to a signal from direction \( \Phi \in [-\Phi/2, \Phi/2] \), \( a(\Phi) = [a_\Phi(\Phi), \ldots, a_\Phi(\Phi)]^T \in \mathbb{Z}^{M \times 1} \), can be expressed as

\[
a_m(\Phi) \approx e^{-j\varphi_0} \sqrt{\text{sinc}} \left( m - \frac{D_y}{\lambda} \sin \phi \right), \quad m \in \mathcal{M},
\]

where \( \varphi_0 \) is a phase shift from lens’ aperture to the array and it is common for all array elements. Since DoA estimation is based on the relative values of the array response, \( \varphi_0 \) will not influence our further analysis so we ignore this term by setting \( \varphi_0 = 2n\pi \) for some integer \( n \), just like [1], [2].

\( L \) \((L < M)\) non-coherent narrowband signal sources impinge the antenna array from directions \( \Phi = [\phi_1, \ldots, \phi_L]^T \in \mathbb{Z}^{L \times 1} \), where \( \phi_l \in [-\Phi/2, \Phi/2], l = 1, \ldots, L. [\cdot]^T \) means the transpose. For snapshot \( t \in \mathbb{N} \), the received vector \( x(t) \in \mathbb{C}^{M \times 1} \) is given by

\[
x(t) = \sum_{l=1}^{L} a(\phi_l)s_l(t) + n(t),
\]

where \( s_l(t) \) is the signal of the \( l \)th source and \( n(t) \in \mathbb{C}^{M \times 1} \) is the noise vector at time \( t \). Let us define \( x(t) = [s_1(t), \ldots, s_L(t)]^T \in \mathbb{C}^{L \times 1} \) and \( A = [a(\phi_1), \ldots, a(\phi_L)] \in \mathbb{Z}^{M \times L} \). Then, we can rewrite (3) as

\[
x(t) = As(t) + n(t).
\]

As [5], we assume both the signals and the noise are stationary, second-order ergodic, spatially and temporally white complex Gaussian processes. \( s(t) \) and \( n(t) \) are circularly-symmetric complex joint-Gaussian with distributions \( \mathcal{N}(0, P) \) and \( \mathcal{N}(0, \sigma^2 I_M) \), respectively. \( I_M \) is the identity matrix of size \( M \). Since the signal sources are non-coherent, \( P \) is assumed to be non-singular and diagonal. The \( i \)th element on the diagonal is \( P_i \), which is the signal power of the \( i \)th source. The rank of \( P \) is \( L \).

The array covariance matrix is given as

\[
R = APA^T + I_M.
\]
When $i \neq j$, the $(i,j)$th element of this matrix, can be expressed as

$$R_{ij} = \sum_{k=1}^{L} \text{Asinc} \left( i - M - 1 - \frac{D_y}{\lambda} \sin \phi_k \right) \cdot \text{sinc} \left( j - M - 1 - \frac{D_x}{\lambda} \sin \phi_k \right) P_k,$$  

and the $(i,i)$th element can be expressed as

$$R_{ii} = \sum_{k=1}^{L} \text{Asinc} \left( i - M - 1 - \frac{D_x}{\lambda} \sin \phi_k \right)^2 P_k + \sigma_N^2.$$  

We can observe that the $(i,j)$th element of the array covariance matrix is a weighted sum of all the signal power (added by the power of noise if the element is on diagonal). The weight of the $k$th signal power is dependent on the distances between its focal point and antenna $i - M - 1$, antenna $j - M - 1$. The nearer their distances are, the larger the weight is. This observation coincides with the energy-focusing property mentioned in Section I. Assuming the number of snapshots is $N$, the goal of this paper is to provide an algorithm which estimates $\phi$ from $x(1), \ldots, x(N)$.

### III. OVERVIEW OF OUR ALGORITHM

In this section, we introduce the idea of our algorithm. In general, our algorithm can be divided into three main steps.

First, DoA estimation has been widely investigated for a long time and many classical algorithms, e.g., MUSIC [6], estimation of signal parameters via rotational invariance techniques (ESPRIT) [7], and Iterative Quadratic Maximum Likelihood (IQML) [8] are provided to obtain an accurate estimation of DoA. Interested readers may refer to relevant textbooks and surveys, e.g., [9]–[11]. However, these algorithms are not suitable for lens antenna arrays. Algorithms like IQML are only suitable for conventional antenna array where every array element receives the incident wave with the same energy, and only the phases of the wave at array elements are different. Actually, these algorithms are designed by exploiting the DoA information implicit in the difference of the phases. However, due to the unique angle-dependent energy focusing capability of lens antenna array, its array responses are quite different from the conventional ones. Hence, algorithms like IQML cannot be directly used. Hence, modifying these algorithms for lens antenna array should be a good idea. Thus, in the first step of our algorithm, we provide a special version of root-MUSIC for lens antenna arrays.

Compared with conventional root-MUSIC, our version has two main differences.

First, in conventional root-MUSIC, the Vandermonde structure, which plays an important part in the algorithm, is obtained by nature. However, the array response of a lens antenna array does not have a Vandermonde structure, which makes direct application of root-MUSIC and its variants (e.g., unitary root-MUSIC [12]) impossible. Here, by re-expressing the ‘sinc’ function inside the array response in another form, in our root-MUSIC version, we construct a new vector which has the Vandermonde structure.

Second, in conventional root-MUSIC, the estimated DoAs are obtained by solving a $(2M - 2)$th degree equation (see equation (27) provided later) and find the desired $L$ ones of all $2M - 2$ roots. It can be done because the $L$ roots containing the DoA information must be with the largest magnitude inside the unit circle. However, in the root-MUSIC for lens antenna array, where is no such rule to help us distinguish the desired roots from the others.

To find the desired roots, in the second step of our algorithm, we use outlier detection. Here, we solve $M - L$ ($M - 1$)th degree equations. For each equation, there are $L$ desired roots containing the DoA information and $M - 1 - L$ useless ones. Ideally, these desired roots should be the same for all $M - L$ equations and the others are quite different. Hence, we can treat the undesired roots as isolated points and remove them by outlier detection.

Due to the noise and the limited number of snapshots, the DoAs obtained from the different equations cannot be the same but are perturbed around the true value. Hence, in the third step, we use clustering to divide all DoAs obtained into $L$ clusters, and treat the center of each cluster as the estimated DoA.

The proposed method is summarized as Algorithm 1. There are three main steps in the algorithm:

1) A specially modified version of the Root-MUSIC algorithm for lens antenna arrays. The details of the algorithm are presented in Section IV.

2) Outlier detection method. This is presented in Section V.

3) Clustering method. For clustering, we use the K-medoids clustering algorithm [13]–[15]. Briefly, the K-medoids algorithm works as follows: First, we greedily select $L$ of all the data points as the medoids to minimize the cost and associate each data point to the closest medoid. Then, for each medoid and for each non-medoid data point, we consider the swap of these two points, and compute the cost change.

### Algorithm 1 A DoA Estimation Algorithm for Lens Antenna Array via Root-MUSIC, Outlier Detection, and Clustering

**Inputs:** $L$ - the number of signal sources, $x(t)$ - the received vectors at snapshot $t$

**Output:** $\phi$ - the estimated DoAs

1. Use a special version of Root-MUSIC for lens antenna array to obtain the root candidates, $Z = LAA\text{\_}\text{\_}MUSIC(L, x(1), \ldots, x(N)).$

2. Use the outlier detection method to find and clean the undesired roots, and obtain the DoA data, $\phi_{\text{data}} = \text{outlier\_detection}(L, Z)$

3. Divide the DoA data into $L$ clusters and treat the center of each cluster as the estimated DoA, $\phi = \text{clustering}(L, \phi_{\text{data}})$

4. return $\phi$
Finally, we perform the best swap. If it decreases the cost function, we do the swap again. Otherwise, the algorithm terminates. Since it is a standard method, we refer to the reader to [13]–[15] for a more detailed explanation.

IV. ROOT-MUSIC FOR LENS ANTENNA ARRAY

The sample data covariance matrix is given as

$$\hat{R} = \frac{1}{N} \sum_{t=1}^{N} x(t)x(t)^H.$$  

Let \(\hat{\lambda}_1, \ldots, \hat{\lambda}_M\) be the ordered eigenvalues of \(\hat{R}\), such that \(\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_M > 0\), and let \(\hat{\xi}_i\) be the eigenvectors associated with \(\hat{\lambda}_i\). According to [4], [16], [17], we can partition these eigenvectors into signal eigenvectors and noise eigenvectors. Ideally, the signal eigenvectors span the range space of \(A\) whereas the noise eigenvectors span its orthogonal complement, i.e., the nullspace of \(A^H\). Hence, DoAs are estimated using

$$a(\phi)^H \hat{G}^H a(\phi) = 0, \quad l = 1, \ldots, L,$$  

where \(\hat{G} = [\hat{e}_{L+1}, \ldots, \hat{e}_M] \in \mathbb{C}^{M \times (M-L)}\), which is made up of the noise eigenvectors.

A. CONSTRUCTING VANDERMONDE STRUCTURE

In conventional root-MUSIC, the array response is a vector with Vandermonde structure and can be written as \(a(z) = [1, z^{-1}, \ldots, z^{-(M-1)}]^T\). By solving (9), we can obtain \(2M - 2\) roots. Among them, the \(L\) number of the roots which are inside and closest to the unit circle are picked. Then, the estimated DoA are derived using these picked roots. For lens antenna array, the array response is not a Vandermonde vector any more, so the conventional root-MUSIC cannot work. Hence, we construct a new vector with Vandermonde structure in this subsection.

First, as \(N \to +\infty\), \(\hat{R}\) converges to \(APA^H + IM\). Since \(A\), \(P\), and \(IM\) are all real, \(\hat{R}\) should be real as \(N \to +\infty\). Hence, for \(\hat{R}\), we only consider its real part, because its imaginary part is caused by the limited number of snapshots. We define \(\tilde{R}\) as the real part of \(\hat{R}\), with eigenvalues \(\tilde{\lambda}_m\) and eigenvectors \(\tilde{e}_m, m = 1, \ldots, M\). Let \(\tilde{G} = [\tilde{e}_{L+1}, \ldots, \tilde{e}_M] \in \mathbb{R}^{M \times (M-L)}\), and we use \(\tilde{G}\) instead of \(\hat{G}\) in (9).

The \(i\)th element of \(\tilde{e}_m\) is denoted by \(\tilde{e}_{i,m}\), so \(\tilde{e}_m\) can be expressed as \([\tilde{e}_{-M,m}, \ldots, \tilde{e}_{M,m}]^T\). In the following, we define

$$z = \frac{D_s \sin \phi}{\lambda},$$  

and express the response of antenna \(m\) in terms of \(z\), as

$$a_m(z) = \sqrt{\lambda} \text{sinc}(m - z).$$  

To construct the Vandermonde structure, we first give the following Lemma.

**Lemma 1:** The expression

$$\sum_{i=-M}^{M} (-1)^i e_{i,m} \prod_{n=-M,n \neq i}^{M} (z-n)$$  

can be expanded to \([z^{M-1}, \ldots, 1] \cdot f_m\), where \(f_m = [f_{m,0}, \ldots, f_{m,M-1}]^T \in \mathbb{R}^{M+1}\), and

$$f_{m,j} = \sum_{i=-M}^M (-1)^i e_{i,m} \chi_{j,i}, \quad j = 0, \ldots, M-1.$$  

For all \(i \in [-M, \ldots, M]\) and \(j \in [1, \ldots, M-1]\), \(\chi_{j,i}\) can be recursively expressed as

$$\chi_{j,i} = \theta_j + i\chi_{j-1,i},$$  

and \(\chi_{0,i} = 1\), \(\forall i \in [-M, \ldots, M]\), \(\theta_0, \ldots, \theta_{M-1}\) are given by

$$\theta_s = \sum_{1 \leq j_1 < j_2 < \cdots < j_s \leq M} (j_1 - M - 1)(j_2 - M - 1) \cdots (j_s - M - 1), \quad s = 1, \ldots, M,$$  

and \(\theta_0 = 1\).

**Proof:** According to the property of elementary symmetric polynomials [18], we can obtain

$$\prod_{n=-M}^M (z-n) = \sum_{j=0}^{M-1} \theta_j z^{M-j}.$$  

In addition, we define that

$$\prod_{n=-M,n \neq i}^M (z-n) = \sum_{j=0}^{M-1} \chi_{j,i} z^{M-1-j}.$$  

From (16) and (17), we can observe that

$$\sum_{j=0}^M \theta_j z^{M-j} = (z-i) \sum_{j=0}^{M-1} \chi_{j,i} z^{M-1-j}. $$  

The corresponding coefficients of the polynomials on the left and right sides of (18) should be equal, so we can obtain (14) and \(\chi_{0,i} = \theta_0 = 1\). Substituting (17) into (12), we can obtain (13) and the lemma is proven.

Now, we give the following theorem.

**Theorem 1:** \(a(z)^T \tilde{e}_m = 0\) is equivalent to the following equations.

$$[z^{M-1}, \ldots, 1] \cdot f_m = 0.$$  

**Proof:**

1) **SUFFICIENCY**

In this part, we prove any \(z\) satisfying \(a(z)^T \tilde{e}_m = 0\) must satisfy (19). If there exists a \(z_0 \in [-M, \ldots, M]\) such that \(a(z_0)^T \tilde{e}_m = 0\), since we have

$$a(z_0)^T \tilde{e}_m = \sum_{i=-M,i \neq z_0}^{M} \sqrt{\lambda} \text{sinc}(i - z_0) e_{i,m} + \sqrt{\lambda} \text{sinc}(z_0 - z_0) e_{z_0,m} = \sqrt{\lambda} e_{z_0,m},$$  

we can obtain that \(e_{z_0,m} = 0\). Thus, when \(z_0 \in [-M, \ldots, M]\), we can derive that

$$\sum_{i=-M}^{M} (-1)^i e_{i,m} \prod_{n=-M,n \neq i}^{M} (z_0-n) = (-1)^{z_0} e_{z_0,m} \prod_{n=-M,n \neq z_0}^{M} (z_0-n) = 0.$$  

If there exists a \( z_0 \not\in \{-M, \ldots, M\} \) such that \( a(z_0)^{\dagger} \hat{e}_m = 0 \), using the definition of sinc(x) that sinc(x) = \( \sin(\pi x)/(\pi x) \), we can rewrite \( a(z_0), i = -M, \ldots, M, \) as

\[
a(z_0) = \sqrt{A} \frac{\sin(\pi i) \cos(z_0) - \cos(\pi i) \sin(z_0)}{\pi \pi - z_0}. \tag{22}
\]

Since \( i \) is an integer, we have sinc(\( \pi i \)) = 0 and cos(\( \pi i \)) = (-1)^{\!i}. Then, we can further simplify the equation above as

\[
a(z_0) = \frac{\sqrt{A} \sin(z_0)}{\pi} \frac{(-1)^{\!i}}{z_0 - i}. \tag{23}
\]

Note that since \( z_0 \) is not an integer, sinc(\( \pi z_0 \)) \( \neq \) 0. Then, from \( a(z_0)^{\dagger} \hat{e}_m = 0 \), we can derive that

\[
\sum_{i=-M}^{M} \frac{(-1)^{\!i} e_{i,m}}{z_0 - i} = 0. \tag{24}
\]

Multiplying both sides of (24) by \( \prod_{n=-M}^{M} (z_0 - n) \), we can obtain the term in (12) is also 0. Summarizing the above results, according to Lemma 1, we can obtain (19). Thus, the sufficiency is proven.

2) NECESSITY

In this part, we prove any \( z \) satisfying (19) must satisfy \( a(z)^{\dagger} \hat{e}_m = 0 \). First, according to Lemma 1, (19) is equivalent to

\[
\sum_{i=-M}^{M} (-1)^{\!i} e_{i,m} \prod_{n=-M}^{M} (z - n) = 0. \tag{25}
\]

If there exists a \( z_0 \in \{-M, \ldots, M\} \) satisfying (25), similarly, we can observe that

\[
(-1)^{\!i} e_{z_0,m} \prod_{n=-M}^{M} (z_0 - n) = 0. \tag{26}
\]

Since \( \prod_{n=-M}^{M} (z_0 - n) \neq 0 \), we must have \( e_{z_0,m} = 0 \). According to (20), we have \( a(z_0)^{\dagger} \hat{e}_m = \sqrt{A} e_{z_0,m} = 0 \).

If there exists a \( z_0 \not\in \{-M, \ldots, M\} \) satisfying (25), dividing both sides of (25) by \( \prod_{n=-M}^{M} (z_0 - n) \), we can obtain (24).

Then, multiplying both sides of (24) by \( \sqrt{A} \sin(z_0)/\pi \) and using (22) and (23), we can obtain that \( a(z_0)^{\dagger} \hat{e}_m = 0 \). Thus, the necessity is proven.

From the above theorem, (9) becomes

\[
[z^{M-1}, \ldots, 1] F F^{\dagger} [z^{M-1}, \ldots, 1]^{\dagger} = 0, \tag{27}
\]

where \( F = [f_{L+1}, \ldots, f_{M}] \in \mathbb{R}^{M \times (M-L)} \). Compared with (9), we use transpose instead of Hermitian transpose because both \( z \) and \( F \) are real. Thus, we successfully construct a vector with Vandermonde structure.

### B. Obtaining the Root Candidates

However, different from the conventional case, it is difficult to find our desired \( L \) roots containing the information of the DoAs among all the \( 2M - 2 \) roots in (27), so we here use another method. By observation, we find that (27) can be re-expressed as

\[
\sum_{i=L+1}^{M} \left( [z^{M-1}, \ldots, 1] f_i \right)^{\dagger} = 0. \tag{28}
\]

From (10), we can see that the roots containing DoA information should be in the field of reals. Hence, in principle, those \( L \) roots should also satisfy all the following \( M - L \) useless roots containing no information about the DoAs. Here, we construct a matrix \( Z \in \mathbb{C}^{(M-L) \times (M-1)} \) to store all the root candidates obtained from (29), where the \( (m - L) \)th row of \( Z \) is the vector of roots derived from solving \( [z^{M-1}, \ldots, 1] f_i = 0 \), \( i = L + 1, \ldots, M \). However, due to the noise and the limited number of snapshots, the desired roots derived from different equations above cannot be exactly the same but perturb around the true value. In addition, for each equation, we will also obtain \( M - 1 \) useless roots containing no information about the DoAs. Here, we construct a matrix \( Z \in \mathbb{C}^{(M-L) \times (M-1)} \) to store all the root candidates obtained from (29), where the \( (m - L) \)th row of \( Z \) is the vector of roots derived from solving \( [z^{M-1}, \ldots, 1] f_i = 0 \). We leave the task of distinguishing the desired roots from all the root candidates to the next step.

Even though there is no analytical solution for (29) when \( M - 1 \) is equal to or larger than 5, there are lots of methods (please refer to [19]) available to find the roots of a polynomial quickly and accurately, such as Newton’s method [20].

The special version of Root-MUSIC for lens antenna array provided in this section is summarized as Algorithm 2.

### V. Outlier Detection

In this section, we treat the roots containing no information about the DoAs as outliers and remove them using outlier detection.
From IV-B, we know among the roots obtained from each equation of (29), there are \( L \) ones containing the information about the DoAs. Thus, for the root candidates obtained from each equation, we leave no more than \( L \) of them. The rule of selection is given as follows: First, from (11) we know that the desired rules should be real and not larger than \( \leq D_i/\lambda \), so we remove all the root candidates which do not satisfy these two conditions. For many cases, we can further remove the root candidates by roughly limiting the range of DoAs using the energy received. For example, if starting with one element, the array elements have continuous drops in energy all the way to the edge, it means no incident waves arrive at this part. Second, ideally, the desired DoA should be real and not larger than \( D_i/\lambda \), so we remove all the root candidates which do not satisfy (28). Due to the noise and the limited number of snapshots, \( \sum_{i=L+1}^{M} (|z_i^{M-1}, \ldots, 1| \cdot f_i)^2 \) cannot be exactly equal to 0. Hence, here we choose \( L \) left roots with minimal value of \( \sum_{i=L+1}^{M} (|z_i^{M-1}, \ldots, 1| \cdot f_i)^2 \). If there are less than \( L \) roots left, we choose all of them.

Then, we map all the root candidates left to their corresponding DoA values by

\[
\phi = \arcsin \left( \frac{z\lambda}{D_y} \right),
\]

and put them into one big set. We call this set DoA dataset, and each element inside is one DoA data.

The DoA dataset can be illustrated as Fig. 3. First, it contains some groups of points that very close to each other, as the solid black circles. The points in the same group actually correspond to one incident wave, and they will later be divided into the same cluster, of which the center will be the estimated DoA of this incident wave. However, besides these points, there are still some isolated points which will influence the results of clustering and should be removed. We set a threshold \( D_{t1} \), if the distance from one DoA data to its nearest point is larger than \( D_{t1} \), we believe this DoA data is isolated and remove it. Using this method, we can successfully remove the points like the hollow circles in Fig. 3. However, this method cannot remove the points like the hollow squares. To solve this problem, we calculate the local outlier factor (LOF) [21] of each DoA data and remove the data point whose LOF is larger than a given threshold \( D_{t2} \). LOF is a score that tells how likely a certain data point is an outlier. The higher the LOF of a data point, the more likely it is to be an outlier. Note that LOF requires a new parameter, \( K_{near} \), which is the number of neighbors that the LOF calculation is considering, and in our following simulations we will assign specific values to it.

In summary, the outlier detection is given as Algorithm 3. After the outlier detection, we perform K-medoids clustering, and the center of one partitioned cluster is the estimated DoA of one incident wave.

**Algorithm 3 An Outlier Detection Method to Find and Clean the Undesired Points, outlier_detection\( (L, Z) \)**

**Inputs:** \( L \) - the number of signal sources, \( Z \) - the matrix of root candidates

**Output:** \( \phi_{data} \) - the desired DoA data points after data clean

1. Initialize \( \phi_{data} \) as an empty set \( \{ \} \).
2. for \( i = 1 \) to \( M - L \)
   3. Initialize \( z_{this} \) as the \( i \)th row of \( Z \), and \( z_{value} \) as \( 0_{1 \times (L-1)} \).
4. for \( j = M - 1 \) to 1 do
   5. Initialize \( z_{p} \) as \( z_{this}(j) \).
   6. if \( z_{p} \notin \mathbb{R} \) or \( |z_{p}| > D_{y}/\lambda \) then
      7. \( z_{this} \leftarrow z_{this}\setminus\{z_{p}\} \)
      8. \( z_{value} \leftarrow z_{value}\setminus\{z_{value}(j)\} \)
   else
      9. \( z_{value}(j) \leftarrow \sum_{m=L+1}^{M} (\{z_{p}^{M-1}, \ldots, 1\} \cdot f_m)^2 \)
10. end if
11. end for
12. if \( \text{length}(z_{this}) \leq L \) then
   13. \( \phi_{data} \leftarrow \phi_{data} \cup \text{mapping}(z_{this}) \)
14. else
   15. \( v_{index} \leftarrow \text{indices of } z_{value} \text{ with minimal } L \text{ values} \)
   16. \( \phi_{data} \leftarrow \phi_{data} \cup \text{mapping}(z_{this}(v_{index})) \)
   end if
19. end for
20. Derive \( \phi_{Dmin} \), where the \( i \)th element is the distance between \( \phi_{data}(i) \) and its nearest point.
21. for \( i = 1 \) to \( \text{length}(\phi_{data}) \)
22. if \( \phi_{Dmin}(i) > D_{t1} \) then \( \phi_{data} \leftarrow \phi_{data}\setminus\{\phi_{data}(i)\} \)
   Remove the isolated point whose distance from the nearest point is larger than a given threshold.
23. end if
24. end for
25. Derive \( \phi_{LOF} \), where the \( i \)th element is the local outlier factor of \( \phi_{data}(i) \).
26. for \( j = 1 \) to \( \text{length}(\phi_{data}) \)
27. if \( \phi_{LOF}(j) > D_{t2} \) then \( \phi_{data} \leftarrow \phi_{data}\setminus\{\phi_{data}(i)\} \)
   Remove the isolated point whose local outlier factor is larger than a given threshold.
28. end if
29. end for
30. return \( \phi_{data} \)

**VI. NUMERICAL RESULTS**

In this section, we give some numerical results to show the performance of our method. We set \( D_y/\lambda = 20 \) and \( A = 1 \). The azimuth coverage angle \( \Phi = \pi \), so the DoA can come from \(-90^\circ\) to \(90^\circ\). \( \bar{M} = |D_y \sin(\Phi/2)/\lambda| = 20 \).

Hence, 41 array elements placed on the focal arc of the lens,
we set the number of snapshots $N = 10$, SNR = 20dB, $D_{11} = 4.5\degree$, $D_{12} = 3.5$, and $K_{\text{near}} = 5$.

![Table 1](image)

TABLE 1. True and estimated DoAs where the number of snapshots $N = 10$, SNR = 20dB, $D_{11} = 4.5\degree$, $D_{12} = 3.5$, and $K_{\text{near}} = 5$.

| True DoAs | Estimated DoAs | True DoAs | Estimated DoAs |
|-----------|----------------|-----------|----------------|
| $-35\degree$ | $-34.94\degree$ | $10\degree$ | $15\degree$ |
| $-30\degree$ | $-30.13\degree$ | $15\degree$ | $20\degree$ |
| $-25\degree$ | $-25.22\degree$ | $-15\degree$ | $-15\degree$ |
| $-20\degree$ | $-20.58\degree$ | $-10\degree$ | $-10\degree$ |
| $-15\degree$ | $-15.49\degree$ | $-5\degree$ | $-5\degree$ |
| $-10\degree$ | $-10.86\degree$ | $X$ | $20\degree$ |
| $-5\degree$ | $-5.13\degree$ | | $35\degree$ |
| $X$ | $20\degree$ | | $35\degree$ |

![Figure 4](image)

**Figure 4.** Array response of array elements with indices 3, 4, and 5 for different DoAs.

where the angle of antenna $m$ relative to the x-axis is $\phi_m = \arcsin(0.05m)$, $m = 0, \pm 1, \ldots, \pm 20$.

To simulate a large number of signals arriving at the receiver simultaneously, we assume there are $L = 14$ equally powered uncorrelated narrowband sources with DoAs $\{\pm 35\degree, \pm 30\degree, \ldots, \pm 5\degree\}$. Thus, $P$ can be given as $\sigma_P^2 I_L$, where $\sigma_P^2$ is the average power of the signals. Fig. 4 shows array response of array elements with indices 3, 4, and 5 as an example, and the situations for other array elements are similar. We can observe that since the incident angles of the adjacent signals are similar, the groups of array elements receiving these signals overlap. Hence, it is impossible to divide the array elements into some separated subsets and make each one handle one incident wave.

Table 1 is the result of our method for one simulation. Here we set the number of snapshots $N = 10$ and the signal-to-noise ratio (SNR) as 20dB. For outlier detection, we use $D_{11} = 4.5\degree$, $D_{12} = 3.5$, and $K_{\text{near}} = 5$. We roughly limit the range of DoAs by the energy received at first. From this table, we can find the error between the estimated and the true values come from two parts. One is that the true DoA fails to be detected and another angle is chosen instead by mistake. In table 1, the receiver fails to detect the incident wave with DoA 10\degree, but believes that there exists a signal coming from 0.25\degree. Usually, the difference between the estimated and the true values can be large in this case (9.75\degree in table 1). We call this kind of error “detection error”. The other one is the small deviation from true DoA due to the noise and limited number of snapshots, which exists for every incident wave. We call this error “estimation error”.

We analyze the performance of our provided method from these two kinds of errors. For detection error, we use the probability that a detection error occurs as the performance metric. In a simulation, if we run $T_{\text{run}}$ independent runs, and the detection error occurs $T_{l}$ times for the $l$th incident wave, we can say the detection error probability (DEP) of the $l$th incident wave, $\text{DEP}_l$, is approximated as

$$\text{DEP}_l = \frac{T_l}{T_{\text{run}}}. \quad (31)$$

Here we give a strict requirement that a detection error occurs as long as the difference between the estimated DoA and true DoA is larger than 2\degree.

For the DoAs which are successfully detected, we use root mean square error (RMSE) to study estimation error. Let $D_l$ be the set of simulation indices where the $l$th incident wave is successfully detected. Then, the RMSE of the $l$th incident wave is given by

$$\text{RMSE}_l = \sqrt{\frac{1}{|D_l|} \sum_{n \in D_l} \left| \hat{\phi}_l(n) - \phi_l \right|^2 }, \quad (32)$$

where $\hat{\phi}_l(n)$ denotes the estimated DoA of the $l$th incident wave at the $n$th run.

Now, we investigate these two performance metrics. Since the figures will become messy if we draw the curves of all incident waves, in the following figures, we only give the curves of incident waves with DoA 5\degree, 10\degree, 15\degree, and 20\degree as examples, while other signals show similar trends in performance. For each simulation point we run 200000 independent runs.

First, we make $N$ change from 1 to 10. Other parameter settings are the same with table 1. Fig. 5 shows the DEP versus the number of snapshots $N$. It is observed that in general, the DEP decreases as more snapshots are available, as expected. There are some small fluctuations because we do not dynamically choose $D_{11}$, $D_{12}$, and $K_{\text{near}}$ for different simulation points. In addition, we can find that the waves with smaller incident angles generally performs worse than those with larger incident angles. It is because that the waves with smaller incident angles are focused onto the array elements near the center, which are heavily influenced by other signals on both sides, while the waves with larger incident angles are
influenced by the signals only from one side. It is worthy to note that our algorithm can provide a good performance with only a small number of snapshots. Moreover, even in the extreme case where \( N = 1 \) or \( N = 2 \), the results of our algorithm are still acceptable.

Fig. 6 shows the RMSE versus the number of snapshots \( N \). We can find that the RMSE also decreases as we use more snapshots. Again, we can find that the waves with smaller incident angles generally performs worse than those with larger incident angles. Finally, our algorithm also provides a good RMSE performance with only a small number of snapshots.

Then, we investigate the DEP and the RMSE under different SNRs. We let the SNR change from 2dB to 20dB and set other parameters same with table 1. Fig. 7 shows the DEP versus the SNR. We can observe that as the SNR increases, the DEPs of all incident waves decrease, as expected. Also, the same with Fig. 5, we can still find that the waves with smaller incident angles have worse performance than those with larger incident angles. A similar situation can be seen in Fig. 8, which illustrates the RMSE performance. Finally, Fig. 7 and Fig. 8 show that our algorithm can provide a good performance even in a low-SNR regime.

To further show that our algorithm can achieve a good performance, here we provide two other algorithms for comparison.

In algorithm A1, we simply use the two array elements closest to the focal point. Assuming these two elements are with index \( m_{c1} \) and \( m_{c2} \), the estimation is derived by solving

\[
\frac{\sin \left( m_{c1} - \frac{D_c}{X} \sin \phi \right)}{\sin \left( m_{c2} - \frac{D_c}{X} \sin \phi \right)} = \frac{1}{N} \sum_{i=1}^{N} \left| x_{m_{c1}}(t) \right| \left| x_{m_{c2}}(t) \right|,
\]

where \( x_{m_{c1}}(t) \) and \( x_{m_{c2}}(t) \) are the signals received by elements \( m_{c1} \) and \( m_{c2} \) at snapshot \( t \), respectively.

Notice that this algorithm is feasible when the signals have separated DoAs, as the case in Fig. 1. Since there is little overlap in energy distribution, the array elements where the energy of the received signals are larger must be closer to the focal points. Thus, for one focal point, we can find the two array elements closest to it.

However, for the case investigated in this paper, when there are a large number of incident waves with similar DoAs, the groups of array elements receiving these signals overlap. Thus, as Fig. 4 shows, it is quite difficult to distinguish which array elements are closest to the focal points, since each element is influenced by multiple signals. Here, for the purpose of comparison, we assume the information about the closest array elements is known in advance, perhaps via an omniscient genie. We refer to this version of algorithm A1 as A1 without a genie. If we do not have such genie, we simply believe the information of DoAs by solving the equation below,

\[
\begin{bmatrix} c^T \end{bmatrix} \begin{bmatrix} 1 & c^T \end{bmatrix}^T = 0,
\]

where \( c \in \mathbb{R}^{L \times 1} \) can be obtained by

\[
c = F_2 F_1^{-1} [0, \ldots, 0, 1]^T,
\]

and the estimated DoAs can be derived by (30).
For comparison, we use the DEP and the RMSE over all antenna elements. Thus,

$$\text{DEP} = \sum_{l=1}^{L} T_l, \quad (36)$$

$$\text{RMSE} = \sqrt{\frac{1}{\sum_{l=1}^{L} |D_l|} \sum_{l=1}^{L} \sum_{n \in D_l} |\phi_l(n) - \phi_l|^2}. \quad (37)$$

Fig. 9 and Fig. 10 show the DEP and the RMSE performance versus the SNR for different algorithms, respectively. Here we still assume the number of snapshots $N = 10$. From both figures we can observe that the algorithm A2, which is based on the fast root-MUSIC method, performs much worse than other two algorithms in the situation where there are a large number of incident waves and array elements. Briefly, it is because a large amount of information is lost when we ‘compress’ $M - L$ equations in (29) into only one equation (34). Though the algorithm A1 with a genie performs slightly better than our algorithm, do not forget that it is based on an impractical assumption that we know which array elements are closest to the focal point of the incident wave. It actually considerably narrows the range because the DoA must be inside the range where its corresponding focal point is between these two array elements. For the case of A1 without a genie, we can observe that our algorithm performs better both in DEP and RMSE. Thus, these results verify that our algorithm is competitive and has good performance.

For supplement, we consider another scenario. Here, the number of array elements $M = 13$, and there are $L = 6$ more closely spaced non-coherent sources. We fix the number of snapshots $N = 10$. To test the performance of our algorithm when the SNR is very low, we let the SNR change from $-20$dB to 20dB. In this scenario, we still compare our algorithm with A1 with a genie, A1 without a genie, and A2. For our algorithm, we set $D_{t1} = 3.5^\circ$, $D_{t2} = 7$, and $K_{\text{near}} = 8$. From Fig. 11 and Fig. 12, we can observe that our algorithm performs better than other three algorithms. In this case, A1 with a genie has a worse performance, which differs from the case shown in Fig. 9 and Fig. 10. This is because the array elements become sparse while DoAs of the sources become close. Thus, different incident waves may share the same closest array elements. However, from (33), we can find that only one DoA can be estimated using the signals received by two adjacent array elements, making A1 with a genie inaccurate in this case.

VII. CONCLUSION

In this paper, we provide a DoA estimation algorithm for the lens antenna array, which is suitable for the situation where the groups of array elements receiving the signals overlap so we cannot handle these signals separately. Our algorithm has three main steps: a special version of root multiple signal
classification (MUSIC) for lens antenna array, outlier detection, and clustering. In numerical results, our algorithm is shown to be a good candidate for DoA estimation even when the numbers of signal sources and array elements are large and the number of snapshots is small.

REFERENCES

[1] Y. Zeng and R. Zhang, “Millimeter wave MIMO with lens antenna array: A new path division multiplexing paradigm,” IEEE Trans. Commun., vol. 64, no. 4, pp. 1557–1571, Apr. 2016.

[2] L. Yang, Y. Zeng, and R. Zhang, “Channel estimation for millimeter-wave MIMO communications with lens antenna arrays,” IEEE Trans. Veh. Technol., vol. 67, no. 4, pp. 3239–3251, Apr. 2018.

[3] Y. Zeng, L. Yang, and R. Zhang, “Multi-user millimeter wave MIMO with full-dimensional lens antenna array,” IEEE Trans. Wireless Commun., vol. 17, no. 4, pp. 2800–2814, Apr. 2018.

[4] A. Barabell, “Improving the resolution performance of eigenstructure-based direction-finding algorithms,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), vol. 8, Apr. 1983, pp. 336–339.

[5] P. Stoica and A. Nehorai, “Performance study of conditional and unconditional direction-of-arrival estimation,” IEEE Trans. Acoust., Speech, Signal Process., vol. 38, no. 10, pp. 1783–1795, 1990.

[6] R. Schmidt, “Multiple emitter location and signal parameter estimation,” IEEE Trans. Antennas Propag., vol. 34, no. 3, pp. 276–280, Mar. 1986.

[7] R. Roy and T. Kailath, “ESPRIT-estimation of signal parameters via rotational invariance techniques,” IEEE Trans. Acoust., Speech, Signal Process., vol. 37, no. 7, pp. 984–995, Jul. 1989.

[8] R. Kumaresan, L. Scharf, and A. Shaw, “An algorithm for pole-zero modeling and spectral analysis,” IEEE Trans. Acoust., Speech, Signal Process., vol. 34, no. 3, pp. 637–640, Jun. 1986.

[9] S. Chandran, Advances in Direction-of-Arrival Estimation. Norwood, MA, USA: Artech House, 2005.

[10] Y. Khmou, S. Safi, and M. Frikel, “Comparative study between several direction of arrival estimation methods,” J. Telecommun. Inf. Technol., vol. 1, pp. 41–48, Jan. 2014.

[11] A. Naaz and R. Rao, “DOA Estimation—A comparative analysis,” Int. J. Comput. Commun. Eng., vol. 3, no. 2, pp. 141–144, 2014.

[12] M. Pesavento, A. B. Gershman, and M. Haardt, “Unitary root-MUSIC with a real-valued eigendecomposition: A theoretical and experimental performance study,” IEEE Trans. Signal Process., vol. 48, no. 5, pp. 1306–1314, May 2000.

[13] Wikipedia Contributors. (2020). K-Medoids—Wikipedia, the Free Encyclopedia, Accessed: Jul. 19, 2020. [Online]. Available: https://en.wikipedia.org/wiki/K-medoids

[14] L. Kaufman and P. Rousseeuw, “Clustering by means of medoids,” in Numerical Methods for Roots of Polynomials: Part 1, P. Brach收到的B.E. degree in electronic information engineering from Zhejiang Gongshang University, where she is currently pursuing the M.E. degree in information and communication engineering.

[15] H. Krim and M. Viberg, “Two decades of array signal processing research: A historical perspective,” Proc. IEEE, vol. 87, no. 4, pp. 67–94, Apr. 1999.

[16] P. Stoica and A. Nehorai, “Performance study of conditional and unconditional direction-of-arrival estimation,” IEEE Trans. Acoust., Speech, Signal Process., vol. 38, no. 10, pp. 1796–1803, Oct. 1990.

[17] L. Kaufman and P. Rousseeuw, “Clustering by means of medoids,” in Numerical Methods for Roots of Polynomials: Part 1, P. Brach收到的B.E. degree in electronic information engineering from Zhejiang Gongshang University, where she is currently pursuing the M.E. degree in information and communication engineering.

[18] T. Kailath, Linear Systems, Englewood Cliffs, NJ, USA: Prentice-Hall, 1980.

[19] J. McNamee and V. Pan, Polynomials and Polynomial Inequalities, Providence, RI, USA: American Mathematical Society, 1961.

[20] K. Madsen, “A root-finding algorithm based on Newton’s method,” BIT, vol. 13, no. 1, pp. 71–75, Mar. 1973.

[21] M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: Identifying density-based local outliers,” in Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD), 2000, vol. 29, no. 2, pp. 93–104.

[22] Q. S. Ren and A. J. Willis, “Fast root MUSIC algorithm,” Electron. Lett., vol. 33, no. 6, pp. 450–451, Mar. 1997.