

Article

An Efficient Direct Position Determination Method for Multiple Strictly Noncircular Sources

Jiexin Yin ^{1,2} , Ding Wang ^{1,2,*}  and Ying Wu ^{1,2}

¹ National Digital Switching System Engineering and Technology Research Center, Zhengzhou 450002, China; Cindyin0807@163.com (J.Y.); hnwuying22@163.com (Y.W.)

² Zhengzhou Information Science and Technology Institute, Zhengzhou 450002, China

* Correspondence: wang_ding814@aliyun.com; Tel.: +86-136-7369-6412

Received: 5 November 2017; Accepted: 19 December 2017; Published: 23 January 2018

Abstract: This paper focuses on the localization methods for multiple sources received by widely separated arrays. The conventional two-step methods extract measurement parameters and then estimate the positions from them. In the contrast to the conventional two-step methods, direct position determination (DPD) localizes transmitters directly from original sensor outputs without estimating intermediate parameters, resulting in higher location accuracy and avoiding the data association. Existing subspace data fusion (SDF)-based DPD developed in the frequency domain is computationally attractive in the presence of multiple transmitters, whereas it does not use special properties of signals. This paper proposes an improved SDF-based DPD algorithm for strictly noncircular sources. We first derive the property of strictly noncircular signals in the frequency domain. On this basis, the observed frequency-domain vectors at all arrays are concatenated and extended by exploiting the noncircular property, producing extended noise subspaces. Fusing the extended noise subspaces of all frequency components and then performing a unitary transformation, we obtain a cost function for each source location, which is formulated as the smallest eigenvalue of a real-valued matrix. To avoid the exhaustive grid search and solve this nonlinear function efficiently, we devise a Newton-type iterative method using matrix Eigen-perturbation theory. Simulation results demonstrate that the proposed DPD using Newton-type iteration substantially reduces the running time, and its performance is superior to other localization methods for both near-field and far-field noncircular sources.

Keywords: array signal processing; passive localization; direct position determination (DPD); noncircular source; frequency domain; extended subspace data fusion (SDF); Newton-type iteration

1. Introduction

Noncircular complex signals (e.g., binary-phase-shift-keying (BPSK), multiple-amplitude-shift-keying (MASK) and offset-quadrature-phase-shift-keying (OQPSK) modulated signals) are extensively employed in modern communication systems. During the last few years, the algorithms for noncircular complex sources with application to signal processing [1–7] have received an upsurge of attention. Different from circular signals, the unconjugated statistical property of noncircular complex signals adds to the amount of available information, which can help to enhance the performance. Since many communication systems offer location-based services, the problem of accurate localization for noncircular sources is worth further investigation.

Conventional localization methods employ two-step processing [8–11], where the measurement parameters (e.g., direction of arrival (DOA), time of arrival (TOA) and frequency difference of arrival (FDOA)) are first extracted, and then, the source positions are estimated. In the first step, parameter estimation algorithms for noncircular (NC) signals such as the NC-MUSIC algorithm

(multiple signal classification algorithm for noncircular signals) [5] can improve the accuracy, due to the exploitation of the noncircular property. However, the localization algorithm in the second step cannot utilize the properties of signals. Moreover, when multiple sources exist, the two-step localization system confronts the association problem of deciding which of the multiple measurements reported by the observers corresponds to which source. If the measurements are not correctly related to the transmitter, additional errors are produced. As a consequence, these conventional two-step methods are suboptimal and cannot guarantee high localization precision for noncircular sources.

Compared with the conventional two-step methods, direct position determination (DPD) [12–22] is a single-step localization method without computing the intermediate parameters and augments the position estimation with the constraint that all measurements correspond to the same geolocation of the transmitter. Therefore, DPD can avoid the association problem, and its location accuracy has shown to be significantly higher than that of the conventional two-step methods, especially under low signal-to-noise ratio (SNR) conditions [13]. However, the DPD technique often requires the transmission of the sampling data to a central processing location and yields a large amount of computations [16].

Nowadays, DPD algorithms applied to the scenario of widely-separated arrays have been intensively investigated [13–18]. A maximum likelihood (ML)-based DPD algorithm was presented in [16], where the locations of multiple sources are decoupled into several lower dimensional optimization problems with the information of uncorrelated waveforms. However, in passive localization systems, signal waveforms are always unknown at the receiver. When solving ML estimators for multiple sources without known waveforms, there is a variety of stray parameters, which requires a substantial computational effort. To this end, Amar and Weiss proposed an iterative algorithm for DPD of multiple unknown signals to reduce the complexity [17]. Although this ML estimator can approach the associated Cramér–Rao bound, its iterative procedure is complicated as it has to perform a grid search during each iteration.

To make the localization computationally attractive for multiple transmitters, an alternative method is the subspace data fusion (SDF). It has lower computational complexity than ML methods, due to the fact that all source positions are estimated from a MUSIC-like cost function that depends on the parameters of the same dimension as that for only one source. Two SDF-based DPDs were developed in [16] and [18], respectively. One is based on the time domain and implicitly uses the array responses [18], and the other processes the frequency-domain observations and relies on the assumption that the envelopes of the signals are the same at all observers, up to delay and amplitude caused by the propagation channel [16]. Therefore, the latter can exploit the location information embedded in both array responses and TOAs, leading to higher accuracy. However, these SDF-based DPD algorithms were designed for general sources (i.e., circular signals) and did not consider the property of noncircular signals. For this reason, we proposed an SDF-based DPD for strictly noncircular sources observed by a moving array in our early work [19], which exploits the time-domain property of noncircular signals. It can be easily adapted to the scenario of widely-separated arrays. Note that similar to the SDF-based DPD in [18], the DPD developed in our previous work utilizes only the array responses, neglecting the location information in the propagation time between the transmitter and the observer.

In light of the aforementioned related works, in this study, we consider the scenario of widely-separated arrays and assume a line of sight (LOS) propagation of multiple signals with unknown waveforms and unknown complex attenuation at each observer array. The purpose of this study is to develop an efficient SDF-based DPD method in the frequency domain for noncircular sources. The contributions of this paper can be summarized as follows:

- We derive the frequency-domain property of strictly noncircular signals and propose an improved SDF estimator. Based on this noncircular property, we establish an extended frequency-domain observation received by all arrays and compute the extended subspaces, which are implicitly related to array responses and TOAs. Fusing the extended subspaces of all frequency components,

a cost function is formulated as the smallest eigenvalue of a symmetric real-valued matrix for each source location, due to a unitary transformation. Therefore, the real-valued eigen-decomposition is required instead of complex computations. Compared with the primitive SDF-based DPDs, this improved SDF estimator retains its superiority for requiring low-dimensional optimization and has higher robustness to noise that comes from exploiting noncircularity.

- We devise a Newton-type iterative algorithm to efficiently solve the prescribed cost function based on matrix Eigen-perturbation theory. It substantially reduces the computations of the straightforward implementation of the optimization for each position, which is always accomplished via a two- or three-dimensional grid search.

The remainder of this paper is organized as follows. Section 2 presents the signal model and formulates the problem. In Section 3, we propose the extended SDF DPD estimator for noncircular sources and devise a Newton-type iterative solution. In Section 4, we show the simulation results and make the discussion. Section 5 concludes the study.

Throughout the paper, upper case and lower case boldface letters will represent matrices and column vectors, respectively. For convenience, we list the notations used in this paper:

$\{\cdot\}^*$	Conjugate.
$\{\cdot\}^T$	Transpose.
$\{\cdot\}^H$	Conjugate transpose.
$\text{blkdiag}\{\cdot\}$	Composition of the block diagonal matrix.
$\text{diag}\{\cdot\}$	Composition of the diagonal matrix.
$\text{vec}\{\cdot\}$	The “vectorization” operator that turns a matrix into a vector by stacking the columns of the matrix, one below another.
\otimes	Kronecker matrix product.
$\mathbb{E}\{\cdot\}$	Expectation.
$\text{tr}\{\cdot\}$	Trace.
$\text{Re}\{\cdot\}$	Real part.
$\text{Im}\{\cdot\}$	Imaginary part.
$[\cdot]_n$	The n -th element of a vector.
$[\cdot]_{nm}$	The n,m -th entry of a matrix.
$\ \cdot\ _2$	Euclidean norm.
$\mathbb{C}^{N \times M}$	Set of the $N \times M$ complex matrices.
$\mathbb{R}^{N \times M}$	Set of the $N \times M$ real matrices.

2. Problem Formulation

2.1. Property of the Noncircular Signal

A signal is normally circular when an arbitrary rotation cannot change its first-order and second-order statistical characteristics, namely rotation invariance. If a signal’s property of rotation invariance is not satisfied, then we take it as a noncircular signal. For a straightforward description, let $s(n)$ be a zero-mean signal, $s(n)$ is considered to be circular with the condition “ $\mathbb{E}[|s(n)|^2] \neq 0$ and $\mathbb{E}[s^2(n)] = 0$ ”, whereas $s(n)$ is referred to as being noncircular with the condition “ $\mathbb{E}[|s(n)|^2] \neq 0$ and $\mathbb{E}[s^2(n)] \neq 0$ ”. Specifically, a strictly noncircular signal can be generated through a phase shift of a real-valued signal [23]. According to this property, we assume $s(n)$ to be a strictly noncircular signal, and therefore, it can be generated as [23]:

$$s(n) = \widetilde{s}(n)e^{i\phi}, \quad (1)$$

where $\widetilde{s}(n)$ is a real-valued signal and ϕ signifies the initial phase of $s(n)$.

Using this time-domain property of a strictly noncircular signal, we can express the discrete Fourier transform (DFT) of $s(n)$ as follows:

$$\begin{aligned}\bar{s}(j) &= \sum_{n=0}^{J-1} s(n) \exp(-i2\pi nj/J) \\ &= \sum_{n=0}^{J-1} \tilde{s}(n) e^{i\phi} \exp(-i2\pi nj/J) \quad j = 0, 1, \dots, J-1,\end{aligned}\quad (2)$$

where $\bar{s}(j)$ is the q -th DFT coefficient of $s(n)$. The conjugate of $\bar{s}(j)$ is given by:

$$\bar{s}^*(j) = \sum_{n=0}^{J-1} \tilde{s}(n) e^{-i\phi} \exp(i2\pi nj/J) \quad 0, 1, \dots, J-1. \quad (3)$$

Using $\exp(i2\pi nj/J) = \exp(-i2\pi n(J-j)/J)$, we rewrite (3) by:

$$\begin{aligned}\bar{s}^*(j) &= \sum_{n=0}^{J-1} \tilde{s}(n) e^{-i\phi} \exp(-i2\pi n(J-j)/J) \\ &= e^{-i2\phi} \sum_{n=0}^{J-1} \tilde{s}(n) e^{i\phi} \exp(-i2\pi n(J-j)/J) \quad j = 0, 1, \dots, J-1.\end{aligned}\quad (4)$$

Based on (2), we have:

$$\bar{s}(J-j) = \sum_{n=0}^{J-1} \tilde{s}(n) e^{i\phi} \exp(-i2\pi n(J-j)/J). \quad (5)$$

Combing (4) and (5), it can be checked that:

$$\bar{s}^*(j) = \bar{s}(J-j) e^{-i2\phi} \quad j = 0, 1, \dots, J-1. \quad (6)$$

The equation related to the frequency-domain signals as shown in (6) lays the foundation of subsequent theoretical development.

2.2. Frequency-Domain Signal Model and Problem Formulation

Let us consider L stationary observers, each of whom is equipped with an antenna array comprising M isotropic sensors. Q transmitters are assumed to radiate uncorrelated narrowband and strictly noncircular signals in the field of interest. We denote $\mathbf{p}_q \in \mathbb{R}^{D \times 1}$ as the position of the q -th transmitter for $q = 1, 2, \dots, Q$. The complex envelope of the outputs of the l -th observer array at time $0 \leq t \leq T$ is expressed as [16,17]:

$$\mathbf{r}_l(t) = \sum_{q=1}^Q b_{ql} \mathbf{a}_l(\mathbf{p}_q) s_q(t - \tau_l(\mathbf{p}_q) - t_{q0}) + \mathbf{n}_l(t) \quad (7)$$

for $l = 1, 2, \dots, L$. Here, $s_q(t - \tau_l(\mathbf{p}_q) - t_{q0})$ is the complex envelope of the q -th source signal, transmitted at time t_{q0} and delayed by $\tau_l(\mathbf{p}_q)$, which indicates the propagation time between the q -th transmitter and the l -th observer. We assume that the source signals are wide-sense stationary. b_{ql} is the unknown complex attenuation coefficient representing the channel effect between the q -th transmitter and the l -th observer. $\mathbf{a}_l(\mathbf{p}_q)$ is the spatial array response of the l -th observer array to the signal transmitted from the position \mathbf{p}_q , which is related to the corresponding DOA. $\mathbf{n}_l(t)$ is the white and circularly Gaussian noise mixed through the l -th observer array. The sources and noises are assumed to be uncorrelated and to have a mean of zero.

Similarly to [16], we partition $\mathbf{r}_l(t)$ ($0 \leq t \leq T$) into K sections, and each length equals T/K . Then, sample the signals of each section at time intervals jT_s ($j = 0, 1, \dots, J-1$) where T_s represents the sampling period. Thus, the DFT of the J samples in the k -th section can be expressed as [16]:

$$\bar{\mathbf{r}}_l(j, k) = \sum_{q=1}^Q b_{ql} \mathbf{a}_l(\mathbf{p}_q) \bar{s}_q(j, k) \exp(-i2\pi\tau_l(\mathbf{p}_q)j/JT_s) + \bar{\mathbf{n}}_l(j, k) \quad j = 0, 1, \dots, J-1, \quad (8)$$

where $\bar{s}_q(j, k)$ and $\bar{\mathbf{n}}_l(j, k)$ are the j -th DFT coefficients of $s_q(t - t_{q0})$ and $\mathbf{n}_l(t)$ in the k -th section, respectively. For convenience of presentation, we define:

$$\bar{\mathbf{a}}_l(j, \mathbf{p}_q) = \mathbf{a}_l(\mathbf{p}_q) \exp(-i2\pi\tau_l(\mathbf{p}_q)j/JT_s), \quad (9)$$

and rewrite $\bar{\mathbf{r}}_l(j, k)$ by:

$$\bar{\mathbf{r}}_l(j, k) = \sum_{q=1}^Q b_{ql} \bar{\mathbf{a}}_l(j, \mathbf{p}_q) \bar{s}_q(j, k) + \bar{\mathbf{n}}_l(j, k) \quad j = 0, 1, \dots, J-1. \quad (10)$$

We now composite the j -th DFT coefficients of all the outputs of L observer arrays and obtain:

$$\begin{aligned} \bar{\mathbf{r}}(j, k) &= [\bar{\mathbf{r}}_1^T(j, k), \bar{\mathbf{r}}_2^T(j, k), \dots, \bar{\mathbf{r}}_L^T(j, k)]^T \\ &= \sum_{q=1}^Q \mathbf{\Gamma}(j, \mathbf{p}_q) \mathbf{b}_q \bar{s}_q(j, k) + \bar{\mathbf{n}}(j, k) \\ &= \sum_{q=1}^Q \boldsymbol{\alpha}_q(j) \bar{s}_q(j, k) + \bar{\mathbf{n}}(j, k), \end{aligned} \quad (11)$$

where:

$$\bar{\mathbf{n}}(j, k) = [\bar{\mathbf{n}}_1^T(j, k), \bar{\mathbf{n}}_2^T(j, k), \dots, \bar{\mathbf{n}}_L^T(j, k)]^T, \quad (12)$$

$$\mathbf{\Gamma}(j, \mathbf{p}_q) = \text{blkdiag}\{\bar{\mathbf{a}}_1(j, \mathbf{p}_q), \bar{\mathbf{a}}_2(j, \mathbf{p}_q), \dots, \bar{\mathbf{a}}_L(j, \mathbf{p}_q)\}, \quad (13)$$

$$\mathbf{b}_q = [b_{q1}, b_{q2}, \dots, b_{qL}]^T, \quad (14)$$

$$\boldsymbol{\alpha}_q(j) = \mathbf{\Gamma}(j, \mathbf{p}_q) \mathbf{b}_q. \quad (15)$$

Let us denote $\boldsymbol{\Phi}(j) = [\boldsymbol{\alpha}_1(j), \boldsymbol{\alpha}_2(j), \dots, \boldsymbol{\alpha}_Q(j)] \in \mathbb{C}^{LM \times Q}$ and $\bar{\mathbf{s}}(j, k) = [\bar{s}_1(j, k), \bar{s}_2(j, k), \dots, \bar{s}_Q(j, k)]^T$. Then, $\bar{\mathbf{r}}(j, k)$ can be further expressed in the matrix form as:

$$\bar{\mathbf{r}}(j, k) = \boldsymbol{\Phi}(j) \bar{\mathbf{s}}(j, k) + \bar{\mathbf{n}}(j, k). \quad (16)$$

Therefore, the covariance matrix of $\bar{\mathbf{r}}(j, k)$ can be written as:

$$\mathbf{R}_r(j) = \mathbb{E}[\bar{\mathbf{r}}(j, k) \bar{\mathbf{r}}^H(j, k)] = \boldsymbol{\Phi}(j) \mathbf{R}_s(j) \boldsymbol{\Phi}^H(j) + \sigma_n^2 \mathbf{I}_{LM}. \quad (17)$$

where $\mathbf{R}_s(j) = \mathbb{E}[\bar{\mathbf{s}}(j, k) \bar{\mathbf{s}}^H(j, k)]$, σ_n^2 denotes the noise power and \mathbf{I}_{LM} signifies the $LM \times LM$ identity matrix.

Note that $\boldsymbol{\Phi}(j)$ is associated with both DOAs and TOAs, and therefore, it contains the location information. The SDF DPD based on the frequency domain [16] relies on the general signal model as shown in (17), which cannot take full advantage of the signal properties. To this end, we come to exploit the noncircularity of signals derived in Section 2.1.

As the conjugate of $\bar{\mathbf{r}}(j, k)$ can be expressed as:

$$\bar{\mathbf{r}}^*(j, k) = \boldsymbol{\Phi}^*(j) \bar{\mathbf{s}}^*(j, k) + \bar{\mathbf{n}}^*(j, k), \quad (18)$$

applying (6) to (18) leads to:

$$\bar{\mathbf{r}}^*(j, k) = \mathbf{\Phi}^*(j) \Delta_\phi \bar{\mathbf{s}}(J - j, k) + \bar{\mathbf{n}}^*(j, k), \tag{19}$$

where $\Delta_\phi = \text{diag}\{e^{-i2\phi_1}, e^{-i2\phi_2}, \dots, e^{-i2\phi_Q}\}$ with ϕ_q being the initial phase of the q -th strictly noncircular source. Replacing j with $J - j$ in (19), it follows that:

$$\bar{\mathbf{r}}^*(J - j, k) = \mathbf{\Phi}^*(J - j) \Delta_\phi \bar{\mathbf{s}}(j, k) + \bar{\mathbf{n}}^*(J - j, k), \tag{20}$$

for $j = 0, 1, \dots, J - 1$. Note that $\bar{\mathbf{r}}^*(J, k) = \bar{\mathbf{r}}^*(0, k)$ as the DFT coefficient is periodic with the period J .

By exploiting the unconjugated property of noncircular signals, we combine (16) and (20), and therefore construct the extended observation:

$$\tilde{\mathbf{r}}(j, k) = [\bar{\mathbf{r}}^T(j, k), \bar{\mathbf{r}}^H(J - j, k)]^T = \tilde{\mathbf{\Phi}}(j) \bar{\mathbf{s}}(j, k) + \bar{\mathbf{n}}(j, k), \tag{21}$$

where $\bar{\mathbf{n}}(j, k) = [\bar{\mathbf{n}}^T(j, k), \bar{\mathbf{n}}^H(J - j, k)]^T$ and $\tilde{\mathbf{\Phi}}(j) = [\mathbf{\Phi}^T(j), \Delta_\phi^T \mathbf{\Phi}^H(J - j)]^T \in \mathbb{C}^{2LM \times Q}$, whose q -th column is given by:

$$\tilde{\boldsymbol{\alpha}}_q(j) = \tilde{\Gamma}(j, \mathbf{p}_q) \tilde{\mathbf{b}}_q, \tag{22}$$

with:

$$\tilde{\Gamma}(j, \mathbf{p}_q) = \text{blkdiag}\{\Gamma(j, \mathbf{p}_q), \Gamma^*(J - j, \mathbf{p}_q)\}, \tag{23}$$

$$\tilde{\mathbf{b}}_q = [\mathbf{b}_q^T, \mathbf{b}_q^H e^{-i2\phi_q}]^T. \tag{24}$$

Then, the covariance matrix of the extended observation $\tilde{\mathbf{r}}(j, k)$ has the following form:

$$\mathbf{R}_{\tilde{\mathbf{r}}}(j) = \mathbb{E}[\tilde{\mathbf{r}}(j, k) \tilde{\mathbf{r}}^H(j, k)] = \tilde{\mathbf{\Phi}}(j) \mathbf{R}_s(j) \tilde{\mathbf{\Phi}}^H(j) + \sigma_n^2 \mathbf{I}_{2LM}. \tag{25}$$

Until now, we have obtained the frequency-domain data model for our work as shown in (25). It is noteworthy that the dimension of the extended covariance matrix $\mathbf{R}_{\tilde{\mathbf{r}}}(j)$ is twice that of the covariance matrix $\mathbf{R}_r(j)$ for circular signals, leading to more available information. To this end, we will apply $\mathbf{R}_{\tilde{\mathbf{r}}}(j)$ to our problem. The problem that we address now is, given $\mathbf{R}_{\tilde{\mathbf{r}}}(j)$ for $j = 0, 1, \dots, J - 1$, to directly estimate the locations of multiple transmitters without explicitly computing TOAs and DOAs.

3. Methods

3.1. Extended SDF

Based on (25), the eigen-decomposition of $\mathbf{R}_{\tilde{\mathbf{r}}}(j)$ yields:

$$\mathbf{R}_{\tilde{\mathbf{r}}}(j) = \mathbf{U}_k^s(j) \boldsymbol{\Sigma}_k^s(j) \mathbf{U}_k^{sH}(j) + \sigma_n^2 \mathbf{U}_k^n(j) \mathbf{U}_k^{nH}(j), \tag{26}$$

where $\boldsymbol{\Sigma}_k^s(j)$ is a diagonal matrix with diagonal elements of the Q largest eigenvalues of $\mathbf{R}_{\tilde{\mathbf{r}}}(j)$. $\mathbf{U}_k^s(j) \in \mathbb{C}^{2LM \times Q}$ is the extended signal subspace comprising the eigenvectors corresponding to the Q largest eigenvalues, and it spans the same space as $\mathbf{\Phi}(j)$ [5,16,19]. $\mathbf{U}_k^n(j) \in \mathbb{C}^{2LM \times (2LM - Q)}$ is the extended noise subspace consisting of the eigenvectors corresponding to the $2LM - Q$ smallest eigenvalue σ_n^2 .

According to the subspace orthogonality principle [5,16,19], namely the columns of $\mathbf{\Phi}(j)$ are orthogonal to the noise subspace of $\mathbf{R}_{\tilde{\mathbf{r}}}(j)$, we have:

$$\tilde{\boldsymbol{\alpha}}_q^H(j) \mathbf{U}_k^n(j) \mathbf{U}_k^{nH}(j) \tilde{\boldsymbol{\alpha}}_q(j) = 0 \quad j = 0, 1, \dots, J - 1 \tag{27}$$

for $q = 1, 2, \dots, Q$. Fusing J DFT components leads to:

$$\sum_{j=0}^{J-1} \tilde{\alpha}_q^H(j) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j) \tilde{\alpha}_q(j) = 0. \quad (28)$$

Inserting (22) into (28), it follows that:

$$\tilde{\mathbf{b}}_q^H \mathbf{Q}(\mathbf{p}_q) \tilde{\mathbf{b}}_q = 0, \quad (29)$$

where $\mathbf{Q}(\mathbf{p}_q)$ is expressed as:

$$\mathbf{Q}(\mathbf{p}_q) = \sum_{j=0}^{J-1} \tilde{\Gamma}^H(j, \mathbf{p}_q) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j) \tilde{\Gamma}(j, \mathbf{p}_q) \in \mathbb{C}^{2L \times 2L}. \quad (30)$$

It is noteworthy that $\tilde{\mathbf{b}}_q$ in (29) is formulated as $\tilde{\mathbf{b}}_q = [\mathbf{b}_q^T, \mathbf{b}_q^H e^{-i2\phi_q}]^T$. Next, we will utilize this structure of $\tilde{\mathbf{b}}_q$ to transform the complex-valued equation in (29) into a real-valued one.

We now rewrite (29) by:

$$(\tilde{\mathbf{b}}_q e^{i\phi_q})^H \mathbf{Q}(\mathbf{p}_q) \tilde{\mathbf{b}}_q e^{i\phi_q} = \tilde{\mathbf{b}}_q'^H \mathbf{Q}(\mathbf{p}_q) \tilde{\mathbf{b}}_q' = 0, \quad (31)$$

where $\tilde{\mathbf{b}}_q' = \tilde{\mathbf{b}}_q e^{i\phi_q}$. As $\tilde{\mathbf{b}}_q = [\mathbf{b}_q^T, \mathbf{b}_q^H e^{-i2\phi_q}]^T$, $\tilde{\mathbf{b}}_q'$ can be given by:

$$\tilde{\mathbf{b}}_q' = [\mathbf{b}'_q{}^T, \mathbf{b}'_q{}^H]^T \quad (32)$$

with $\mathbf{b}'_q = \mathbf{b}_q e^{i\phi_q}$. Based on (32), we perform a unitary transformation of $\tilde{\mathbf{b}}_q'$ to establish a real-valued vector:

$$\tilde{\mathbf{b}}_q = \frac{1}{\sqrt{2}} \mathbf{T} \tilde{\mathbf{b}}_q' \quad (33)$$

where \mathbf{T} is the unitary matrix:

$$\mathbf{T} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_L & \mathbf{I}_L \\ -i\mathbf{I}_L & i\mathbf{I}_L \end{bmatrix}, \quad (34)$$

and thus, $\tilde{\mathbf{b}}_q$ can be expressed in the form of (35):

$$\tilde{\mathbf{b}}_q = [\text{Re}\{\mathbf{b}'_q{}^T\}, \text{Im}\{\mathbf{b}'_q{}^T\}]^T. \quad (35)$$

Since $\mathbf{T}^H \mathbf{T} = \mathbf{I}_{2L}$, it follows that:

$$\tilde{\mathbf{b}}_q' = \sqrt{2} \mathbf{Q}^H \tilde{\mathbf{b}}_q. \quad (36)$$

Substituting (36) back into (31), we get:

$$\tilde{\mathbf{b}}_q^H \mathbf{Q}'(\mathbf{p}_q) \tilde{\mathbf{b}}_q = 0, \quad (37)$$

where:

$$\mathbf{Q}'(\mathbf{p}_q) = \mathbf{T} \mathbf{Q}(\mathbf{p}_q) \mathbf{T}^H \in \mathbb{C}^{2L \times 2L}. \quad (38)$$

Given that $\widetilde{\mathbf{b}}_q$ is real-valued, we rewrite the left side of the equation in (37) by:

$$\begin{aligned} & \widetilde{\mathbf{b}}_q^H \mathbf{Q}'(\mathbf{p}_q) \widetilde{\mathbf{b}}_q \\ &= \widetilde{\mathbf{b}}_q^T (\text{Re}\{\mathbf{Q}'(\mathbf{p}_q)\} + i\text{Im}\{\mathbf{Q}'(\mathbf{p}_q)\}) \widetilde{\mathbf{b}}_q \\ &= \widetilde{\mathbf{b}}_q^T \text{Re}\{\mathbf{Q}'(\mathbf{p}_q)\} \widetilde{\mathbf{b}}_q + i \widetilde{\mathbf{b}}_q^T \text{Im}\{\mathbf{Q}'(\mathbf{p}_q)\} \widetilde{\mathbf{b}}_q. \end{aligned} \quad (39)$$

Due to the fact that $\widetilde{\mathbf{b}}_q^H \mathbf{Q}'(\mathbf{p}_q) \widetilde{\mathbf{b}}_q$ is real-valued, we discard the imaginary term, $i \widetilde{\mathbf{b}}_q^T \text{Im}\{\mathbf{Q}'(\mathbf{p}_q)\} \widetilde{\mathbf{b}}_q$, in (39). Then, $\widetilde{\mathbf{b}}_q^H \mathbf{Q}'(\mathbf{p}_q) \widetilde{\mathbf{b}}_q$ is equivalent to:

$$\widetilde{\mathbf{b}}_q^H \mathbf{Q}'(\mathbf{p}_q) \widetilde{\mathbf{b}}_q = \widetilde{\mathbf{b}}_q^T \text{Re}\{\mathbf{Q}'(\mathbf{p}_q)\} \widetilde{\mathbf{b}}_q. \quad (40)$$

According to the above derivation, the complex-valued equation in (29) is transformed into the following real-valued equation:

$$\widetilde{\mathbf{b}}_q^T \text{Re}\{\mathbf{Q}'(\mathbf{p}_q)\} \widetilde{\mathbf{b}}_q = 0. \quad (41)$$

We now replace $\mathbf{U}_k^n(j)$ with $\hat{\mathbf{U}}_k^n(j)$, which is computed by the eigen-decomposition of the estimated $\mathbf{R}_{\tilde{\mathbf{r}}}(j)$:

$$\hat{\mathbf{R}}_{\tilde{\mathbf{r}}}(j) = \frac{1}{K} \sum_{k=1}^K \tilde{\mathbf{r}}(j,k) \tilde{\mathbf{r}}^H(j,k). \quad (42)$$

Then, the following optimization model presents the solution for \mathbf{p}_q and $\widetilde{\mathbf{b}}_q$:

$$\{\hat{\mathbf{p}}_q, \hat{\mathbf{b}}_q\} = \underset{\mathbf{p}, \widetilde{\mathbf{b}}}{\text{argmin}} \widetilde{\mathbf{b}}^T \text{Re}\{\mathbf{Q}'(\mathbf{p})\} \widetilde{\mathbf{b}}. \quad (43)$$

We observe that the estimation of $\widetilde{\mathbf{b}}_q$ is the eigenvector corresponding to the smallest eigenvalue of $\text{Re}\{\mathbf{Q}'(\mathbf{p})\}$, and the estimation of \mathbf{p}_q is the point that minimizes the smallest eigenvalue of $\text{Re}\{\mathbf{Q}'(\mathbf{p})\}$. Consequently, the position of the q -th transmitter can be obtained by:

$$\hat{\mathbf{p}}_q = \underset{\mathbf{p}}{\text{argmin}} f(\mathbf{p}) \quad (44)$$

for $q = 1, 2, \dots, Q$, where:

$$f(\mathbf{p}) = \lambda_{\min}\{\text{Re}\{\mathbf{Q}'(\mathbf{p})\}\}. \quad (45)$$

Here, $\lambda_{\min}\{\cdot\}$ signifies the smallest eigenvalue of the input matrix. As $f(\mathbf{p})$ is a nonlinear function in terms of the source position, the minimization of $f(\mathbf{p})$ is usually accomplished by a D -dimensional grid search.

3.2. Iterative Solution

When the area of interest is large and the grid step size is small, the grid search is computationally demanding. To locate the transmitters quickly, we will devise a Newton-type iterative method to efficiently solve the proposed cost function for each source location. As is known to all, the Newton-type iterative method provides an efficient way to solve nonlinear optimization problems [24]. It requires computing the gradient and Hessian matrix of the cost function, which is quite easy if the cost function is explicitly written. However, the cost function in (45) is the smallest eigenvalue of a symmetric matrix, whose gradient and Hessian matrix cannot be obtained straightforwardly. Therefore, we apply the

matrix Eigen-perturbation theory to derive the gradient and Hessian matrix of (45) for each source position. For this purpose, a proposition needs to be introduced first.

Proposition 1. Assume a symmetric real-valued matrix $\mathbf{X} \in \mathbb{R}^{N \times N}$ with $\lambda_{\min} = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N = \lambda_{\max}$ being its eigenvalues and $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N$ being the corresponding eigenvectors. Denote $\delta\mathbf{X}$ as the perturbation of \mathbf{X} , and hence $\hat{\mathbf{X}} = \mathbf{X} + \delta\mathbf{X}$ where $\delta\mathbf{X}$ and $\hat{\mathbf{X}}$ are also symmetric. Then, the eigenvalues of the perturbed matrix $\hat{\mathbf{X}}$, $\hat{\lambda}_{\min} = \hat{\lambda}_1 \leq \hat{\lambda}_2 \leq \dots \leq \hat{\lambda}_N = \hat{\lambda}_{\max}$, can be expressed as [25]:

$$\hat{\lambda}_n = \lambda_n + \mathbf{e}_n^T \delta\mathbf{X} \mathbf{e}_n + \mathbf{e}_n^T \delta\mathbf{X} \mathbf{E}_n \delta\mathbf{X} \mathbf{e}_n + o\left(\|\delta\mathbf{X}\|_2^2\right) \quad (n = 1, 2, \dots, N), \tag{46}$$

where:

$$\mathbf{E}_n = \sum_{\substack{m=1 \\ m \neq n}}^N (\lambda_n - \lambda_m)^{-1} \mathbf{e}_m \mathbf{e}_m^T \quad (n = 1, 2, \dots, N), \tag{47}$$

and $o\left(\|\delta\mathbf{X}\|_2^2\right)$ signifies the infinitesimal term of $\|\delta\mathbf{X}\|_2^2$. This proposition reveals the relationship between the matrix perturbation and the eigenvalues of the disturbed matrix. The proof of (46) can be found in [25].

Now, let us denote $\mathbf{X}(\mathbf{p}) = \text{Re}\{\mathbf{Q}'(\mathbf{p})\}$, whose eigenvalues and eigenvectors are $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{2L}$ and $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{2L}$. Define $\delta\mathbf{p}$ as the perturbation of \mathbf{p} and $\hat{\mathbf{p}} = \mathbf{p} + \delta\mathbf{p}$ as the perturbed position. Then, the eigenvalues and eigenvectors of $\hat{\mathbf{X}}(\hat{\mathbf{p}})$ are $\hat{\lambda}_1 \leq \hat{\lambda}_2 \leq \dots \leq \hat{\lambda}_{2L}$ and $\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \dots, \hat{\mathbf{e}}_{2L}$, respectively. Denoting $\delta\mathbf{X} = \hat{\mathbf{X}}(\hat{\mathbf{p}}) - \mathbf{X}(\mathbf{p})$, we obtain the following equation according to the above proposition:

$$\hat{\lambda}_1 = \lambda_1 + \mathbf{e}_1^T \delta\mathbf{X} \mathbf{e}_1 + \mathbf{e}_1^T \delta\mathbf{X} \mathbf{E}_1 \delta\mathbf{X} \mathbf{e}_1 + o\left(\|\delta\mathbf{X}\|_2^2\right), \tag{48}$$

in which:

$$\mathbf{E}_1 = \sum_{m=2}^{2L} (\lambda_1 - \lambda_m)^{-1} \mathbf{e}_m \mathbf{e}_m^T. \tag{49}$$

Applying $\text{vec}\{\mathbf{ABC}\} = (\mathbf{C}^T \otimes \mathbf{A})\text{vec}\{\mathbf{B}\}$ and $\text{tr}\{\mathbf{ABCD}\} = \text{vec}^T\{\mathbf{D}^T\}(\mathbf{C}^T \otimes \mathbf{A})\text{vec}\{\mathbf{B}\}$ (see [26]), (48) can be further expressed as a function of $\delta\mathbf{x} = \text{vec}\{\delta\mathbf{X}\}$:

$$\begin{aligned} \hat{\lambda}_1 &= \lambda_1 + \text{vec}\{\mathbf{e}_1^T \delta\mathbf{X} \mathbf{e}_1\} + \text{tr}\{\mathbf{e}_1 \mathbf{e}_1^T \delta\mathbf{X} \mathbf{E}_1 \delta\mathbf{X}\} + o\left(\|\delta\mathbf{X}\|_2^2\right) \\ &= \lambda_1 + (\mathbf{e}_1^T \otimes \mathbf{e}_1^T) \delta\mathbf{x} + \delta\mathbf{x}^T (\mathbf{E}_1 \otimes \mathbf{e}_1 \mathbf{e}_1^T) \delta\mathbf{x} + o\left(\|\delta\mathbf{x}\|_2^2\right). \end{aligned} \tag{50}$$

where the symmetry of \mathbf{E}_1 and $\delta\mathbf{X}$ is used.

Note that $\hat{\lambda}_1$ is the cost function of the proposed estimator, and therefore, our purpose is to link $\delta\mathbf{p}$ to $\hat{\lambda}_1$. As (50) displays the relationship between $\delta\mathbf{x}$ and $\hat{\lambda}_1$, we proceed to derive the relationship between $\delta\mathbf{x}$ and $\delta\mathbf{p}$ in what follows.

Denote $\mathbf{x}(\mathbf{p}) = \text{vec}\{\mathbf{X}(\mathbf{p})\}$ and $\hat{\mathbf{x}}(\hat{\mathbf{p}}) = \text{vec}\{\hat{\mathbf{X}}(\hat{\mathbf{p}})\}$. Based on the result in [27], we can take the second-order Taylor expansion of $\hat{\mathbf{x}}(\hat{\mathbf{p}})$ around \mathbf{p} as:

$$\hat{\mathbf{x}}(\hat{\mathbf{p}}) = \mathbf{x}(\mathbf{p}) + \dot{\mathbf{X}}_p(\mathbf{p}) \delta\mathbf{p} + \frac{1}{2} (\mathbf{I}_{4L^2} \otimes \delta\mathbf{p}^T) \ddot{\mathbf{X}}_{pp}(\mathbf{p}) \delta\mathbf{p} + o\left(\|\delta\mathbf{p}\|_2^2\right), \tag{51}$$

where:

$$\dot{\mathbf{X}}_p(\mathbf{p}) = \frac{\partial \mathbf{x}(\mathbf{p})}{\partial \mathbf{p}^T} \in \mathbb{R}^{4L^2 \times D} \tag{52}$$

and:

$$\ddot{\mathbf{X}}_{pp}(\mathbf{p}) = \begin{bmatrix} \frac{\partial^2[x(\mathbf{p})]_1}{\partial \mathbf{p} \partial \mathbf{p}^T} \\ \frac{\partial^2[x(\mathbf{p})]_2}{\partial \mathbf{p} \partial \mathbf{p}^T} \\ \vdots \\ \frac{\partial^2[x(\mathbf{p})]_{4L^2}}{\partial \mathbf{p} \partial \mathbf{p}^T} \end{bmatrix} \in \mathbb{R}^{4L^2 D \times D}. \quad (53)$$

For the further derivations of (52) and (53), see Appendixs A and B, respectively. As $\delta \mathbf{x} = \hat{\mathbf{x}}(\hat{\mathbf{p}}) - \mathbf{x}(\mathbf{p})$, according to (51), we can relate $\delta \mathbf{x}$ with $\delta \mathbf{p}$:

$$\delta \mathbf{x} = \dot{\mathbf{X}}_{\mathbf{p}}(\mathbf{p})\delta \mathbf{p} + \frac{1}{2}(\mathbf{I}_{4L^2} \otimes \delta \mathbf{p}^T)\ddot{\mathbf{X}}_{pp}(\mathbf{p})\delta \mathbf{p} + o(\|\delta \mathbf{p}\|_2^2). \quad (54)$$

Then, substituting (54) back into (50) yields:

$$\begin{aligned} \hat{\lambda}_1 &= \lambda_1 + (\mathbf{e}_1^T \otimes \mathbf{e}_1^T)\delta \mathbf{x} + \delta \mathbf{x}^T(\mathbf{E}_1 \otimes \mathbf{e}_1 \mathbf{e}_1^T)\delta \mathbf{x} + o(\|\delta \mathbf{x}\|_2^2) \\ &= \lambda_1 + (\mathbf{e}_1^T \otimes \mathbf{e}_1^T)\dot{\mathbf{X}}_{\mathbf{p}}(\mathbf{p})\delta \mathbf{p} + \\ &\quad \frac{1}{2}(\mathbf{e}_1^T \otimes \mathbf{e}_1^T)(\mathbf{I}_{4L^2} \otimes \delta \mathbf{p}^T)\ddot{\mathbf{X}}_{pp}(\mathbf{p})\delta \mathbf{p} + \delta \mathbf{p}^T \dot{\mathbf{X}}_{\mathbf{p}}^T(\mathbf{p})(\mathbf{E}_1 \otimes \mathbf{e}_1 \mathbf{e}_1^T)\dot{\mathbf{X}}_{\mathbf{p}}(\mathbf{p})\delta \mathbf{p} + o(\|\delta \mathbf{p}\|_2^2). \end{aligned} \quad (55)$$

Because:

$$(\mathbf{e}_1^T \otimes \mathbf{e}_1^T)(\mathbf{I}_{4L^2} \otimes \delta \mathbf{p}^T)\ddot{\mathbf{X}}_{pp}(\mathbf{p})\delta \mathbf{p} = \delta \mathbf{p}^T(\mathbf{e}_1^T \otimes \mathbf{e}_1^T \otimes \mathbf{I}_D)\ddot{\mathbf{X}}_{pp}(\mathbf{p})\delta \mathbf{p}, \quad (56)$$

(55) is equivalent to:

$$\begin{aligned} \hat{\lambda}_1 &= \lambda_1 + (\mathbf{e}_1^T \otimes \mathbf{e}_1^T)\dot{\mathbf{X}}_{\mathbf{p}}(\mathbf{p})\delta \mathbf{p} + \frac{1}{2}\delta \mathbf{p}^T \left\{ (\mathbf{e}_1^T \otimes \mathbf{e}_1^T \otimes \mathbf{I}_D)\ddot{\mathbf{X}}_{pp}(\mathbf{p}) + 2\dot{\mathbf{X}}_{\mathbf{p}}^T(\mathbf{p})(\mathbf{E}_1 \otimes \mathbf{e}_1 \mathbf{e}_1^T)\dot{\mathbf{X}}_{\mathbf{p}}(\mathbf{p}) \right\} \delta \mathbf{p} + o(\|\delta \mathbf{p}\|_2^2) \\ &= \lambda_1 + \mathbf{g}^T(\mathbf{p})\delta \mathbf{p} + \frac{1}{2}\delta \mathbf{p}^T \mathbf{H}(\mathbf{p})\delta \mathbf{p} + o(\|\delta \mathbf{p}\|_2^2), \end{aligned} \quad (57)$$

where $\mathbf{g}(\mathbf{p})$ and $\mathbf{H}(\mathbf{p})$ are given by:

$$\mathbf{g}(\mathbf{p}) = \dot{\mathbf{X}}_{\mathbf{p}}^T(\mathbf{p})(\mathbf{e}_1 \otimes \mathbf{e}_1) \quad (58)$$

and:

$$\mathbf{H}(\mathbf{p}) = (\mathbf{e}_1^T \otimes \mathbf{e}_1^T \otimes \mathbf{I}_D)\ddot{\mathbf{X}}_{pp}(\mathbf{p}) + 2\dot{\mathbf{X}}_{\mathbf{p}}^T(\mathbf{p})(\mathbf{E}_1 \otimes \mathbf{e}_1 \mathbf{e}_1^T)\dot{\mathbf{X}}_{\mathbf{p}}(\mathbf{p}). \quad (59)$$

Finally, applying the result in (57) to the objective function in (45), we can express $f(\hat{\mathbf{p}})$ in terms of the second-order perturbation of $\delta \mathbf{p}$ as follows:

$$f(\hat{\mathbf{p}}) = f(\mathbf{p}) + \mathbf{g}^T(\mathbf{p})\delta \mathbf{p} + \frac{1}{2}\delta \mathbf{p}^T \mathbf{H}(\mathbf{p})\delta \mathbf{p} + o(\|\delta \mathbf{p}\|_2^2). \quad (60)$$

It can be seen from (60) that $\mathbf{g}(\mathbf{p})$ and $\mathbf{H}(\mathbf{p})$ are the gradient and Hessian matrix of $f(\mathbf{p})$, respectively. Therefore, we can get the estimation of \mathbf{p}_q following the Newton-type iterative procedure [28]:

$$\hat{\mathbf{p}}_q^{(i+1)} = \hat{\mathbf{p}}_q^{(i)} - \mu^i \mathbf{H}^{-1}(\hat{\mathbf{p}}_q^{(i)})\mathbf{g}(\hat{\mathbf{p}}_q^{(i)}), \quad (61)$$

where the superscript i denotes the iteration number and μ is the step factor deciding the iteration step size. In this way, each transmitter can be located when an initial position is given. Note that the variable step size is used to solve the contradiction between convergence speed and misadjustment, which is usually encountered in the case of fixed step size.

To summarize this iterative process, we provide the procedure below:

Method: Eigen-Perturbation-Based Newton-Type Iterative Method

- 1: Set ϵ as a positive number that is small enough. Then, choose a suitable $K \in \mathbb{Z}^+$ and $J \in \mathbb{Z}^+ \mathbb{Z}^+$ denotes the set of positive integers), and compute the DFT of the observed samples.
 - 2: Use (42) to estimate the extended covariance matrix $\hat{\mathbf{R}}_{\bar{r}}(j)$ for $j = 0, 1, \dots, J - 1$.
 - 3: Compute the eigenvectors corresponding to the $2LM - Q$ smallest eigenvalues of $\hat{\mathbf{R}}_{\bar{r}}(j)$ for $j = 0, 1, \dots, J - 1$ and thus obtain the extended noise subspaces $\hat{\mathbf{U}}_k^n(j)$ for $j = 0, 1, \dots, J - 1$.
 - 4: **for** $q = 1, 2, \dots, Q$ **do**
 - 5: Initialize $\hat{\mathbf{p}}_q^{(0)}$, and make $i \leftarrow 0$.
 - 6: Insert $\hat{\mathbf{p}}_q^{(i)}$ into (58) and (59) to compute $\mathbf{g}(\hat{\mathbf{p}}_q^{(i)})$ and $\mathbf{H}(\hat{\mathbf{p}}_q^{(i)})$.
 - 7: Compute $\Delta_i = \|\mathbf{g}(\hat{\mathbf{p}}_q^{(i)})\|_2$. If $\Delta_i \leq \epsilon$, $\hat{\mathbf{p}}_q^{(i)}$ is the estimated source position, and stop iterating; otherwise, continue.
 - 8: Use (61) to update $\hat{\mathbf{p}}_q^{(i+1)}$. Make $i \leftarrow i + 1$, and go to Step 6.
 - 9: **end for**
-

Remark 1. Similarly to existing SDF-based DPDs, the method proposed above has an advantage over ML-based methods in the presence of multiple transmitters as it estimates the positions in a decoupled manner. Different from ML functions, which have only one global extremum, our cost function exhibits a few minima at the estimated source positions. Considering that the Newton-type iteration is a local search method and is bound to converge to a local minimum, we use it as a fast converging search procedure instead of the fine search. The initial guess will determine the position to which source the converging local minimum corresponds. Therefore, our iterative method for solving the proposed cost function relies on the initialization to resolve multiple sources, whereas the optimization methods for solving the ML functions require a good initial guess to avoid the local minima. For the initialization of our iterative method, the initial position of each transmitter can be effectively obtained through a coarse search as long as it can separate multiple sources. This can guarantee the accurate and fast localization.

3.3. Computational Complexity

The main computational costs of the proposed methods using exhaustive grid search and using Newton-type iteration and the SDF DPD method for circular sources based on the frequency domain [16] include four parts, i.e., computing the DFT, estimating the covariance matrix, performing the eigen-decomposition and solving the cost function. Table 1 lists the order of the number of real-valued multiply operations in each part consumed by the aforementioned methods, where N_p denotes the number of position grid points and N_{iter} represents the iteration number required for convergence.

Table 1. Computational complexity.

Method	Complexity				Comments
	Computing DFT	Estimating Covariance Matrix	Eigen-Decomposition	Solving Cost Function	
SDF-based DPD in [16]	$O(2LMKJ \log_2 J)$	$O(4L^2M^2KJ)$	$O(4L^3M^3J)$	$O((4L^2M^2QJ + 6L^2M^2J + 8L^3)QN_p)$	- For Q circular sources
Proposed DPD using exhaustive search	$O(2LMKJ \log_2 J)$	$O(16L^2M^2KJ)$	$O(32L^3M^3J)$	$O\left(\left(\begin{array}{c} 8L^3(M^2 + M)J - \\ 8L^2(M + 1)QJ + 8L^3 \end{array}\right)QN_p\right)$	- For Q strictly noncircular sources
Proposed DPD using Newton-type iteration	$O(2LMKJ \log_2 J)$	$O(16L^2M^2KJ)$	$O(32L^3M^3J)$	$O\left(\left(\begin{array}{c} (8L^3(M^2 + M)J - \\ 8L^2(M + 1)QJ)(D + 1)^2 + \\ 4L^2D^2 + 8L^3 + 4L^2D + \\ 4LD^2 + D^3 + 2LD \end{array}\right)QN_{iter}\right)$	- For Q strictly noncircular sources - Complexity for initialization is not included

As shown in Table 1, the SDF-based DPD in [16] is more computationally efficient than the proposed DPD using the exhaustive search because the dimension of the covariance matrix employed in our algorithm is twice as large as the dimension of the covariance matrix used in [16]. On the other hand, the complexity of the proposed DPD using Newton-type iteration is much lower than that of the exhaustive search implementation since N_p is usually much larger than N_{iter} . For further comparison, we will examine their computer running times through the simulations described in Section 4.

4. Results

The purpose of this section is to present the simulation results and performance analysis relying on Monte Carlo simulations. We consider the scenario of three observers located at the positions $[3,3]^T$ (km), $[3,-3]^T$ (km) and $[-3,3]^T$ (km). Each observer is equipped with a uniform linear array (ULA) composed of $M = 3$ sensors. The adjacent sensors are spaced with 0.5λ , where λ denotes the wavelength. Without loss of generality, we assume two transmitters in the field of interest. They arrive at observers with attenuation coefficient vectors being $\mathbf{b}_1 = [10.94 + 0.34i, 0.77 + 0.64i]^T$ and $\mathbf{b}_2 = [10.87 + 0.5i, 0.77 + 0.64i]^T$, respectively. The baseband signal waveforms are generated as narrowband strictly noncircular signals with identical power σ_s^2 , whose initial phases are $\phi_1 = \pi/5$ (rad) and $\phi_2 = \pi/3$ (rad). The noises are uncorrelated, circularly symmetric Gaussian random variables with constant power σ_n^2 . Unless stated otherwise, each observer collects signals in $K = 50$ sections each of $J = 16$ frequencies using a sample rate of 40k (samples/s). For comparison, we invoke the following five estimators:

- Proposed DPD estimator using the devised Newton-type iterative method.
- Proposed DPD estimator using the exhaustive search.
- SDF-based DPD estimator for general circular sources in the frequency domain [16] (denoted by FD-DPD).
- SDF-based DPD estimator for noncircular sources in time domain [19] (denoted by NC TD-DPD).
- Two-step processing estimator: DOA estimation using the NC-MUSIC algorithm [5] and TOA estimation using the ML criterion for noncircular signals [7] at each observer, and pseudo-linear weighted least square localization with the DOA and TOA estimates from all observers used as the data, where DOA and TOA estimates are assumed to be associated with the correct transmitter (denoted by two-step).

In the simulations, the exhaustive search of DPD estimators is implemented through a coarse search with a 0.01-km resolution and then a fine search with a 0.001-km resolution. For the Newton-type iterative method, the positions are initialized using a coarse search with a 0.1-km resolution, which is capable of resolving sources, as will be shown later on. The step factor is $\mu = 0.9$. Under different experimental conditions, each simulation is conducted in 1000 Monte Carlo trials, and the location root mean square error (RMSE) is used to assess the localization performance:

$$\text{RMSE}_q = \sqrt{\frac{1}{1000} \sum_{n=1}^{1000} \|\mathbf{p}_q - \hat{\mathbf{p}}_{n,q}\|^2} \quad (62)$$

for the q -th source, where $\hat{\mathbf{p}}_{n,q}$ denotes the estimation of \mathbf{p}_q in the n -th Monte Carlo trial.

This section will show the performance of the proposed algorithms in seven examples. We begin with an analysis of the initial condition and the convergence performance of our Newton-type iterative method. Two near-field sources are assumed to be located at $\mathbf{p}_1 = [1, -1]^T$ (km) and $\mathbf{p}_2 = [0, 2]^T$ (km), and the corresponding scenario is shown in Figure 1. For different SNRs, we assess the inverse cost function of the proposed estimator over the candidate positions with a 0.1-km resolution. Figure 2 displays the corresponding evaluation of the inverse cost function within a square area of 2×2 (km \times km) around the true positions of the two transmitters at SNR = 0 dB. It can be seen that the maximum of the inverse cost function appears near the true position of each source,

and there is no pseudo peak in this area, which indicates that the coarse search with a 0.1-km resolution can provide the effective initial positions. Then, we evaluate the normalized Euclidean norm of the gradient for each source versus the number of iteration steps, when the SNR is -10 dB, 0 dB and 10 dB, respectively. As shown in Figure 3, fewer than six iterations are required for convergence in the whole range of SNRs.

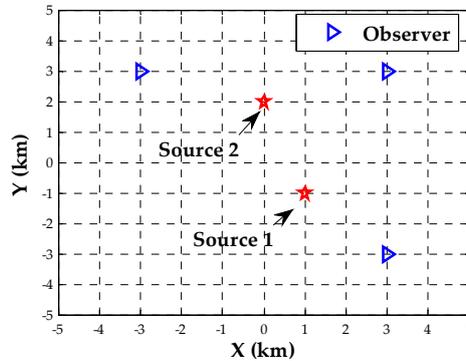


Figure 1. Scenario of three observers and two near-field sources placed on the ground.

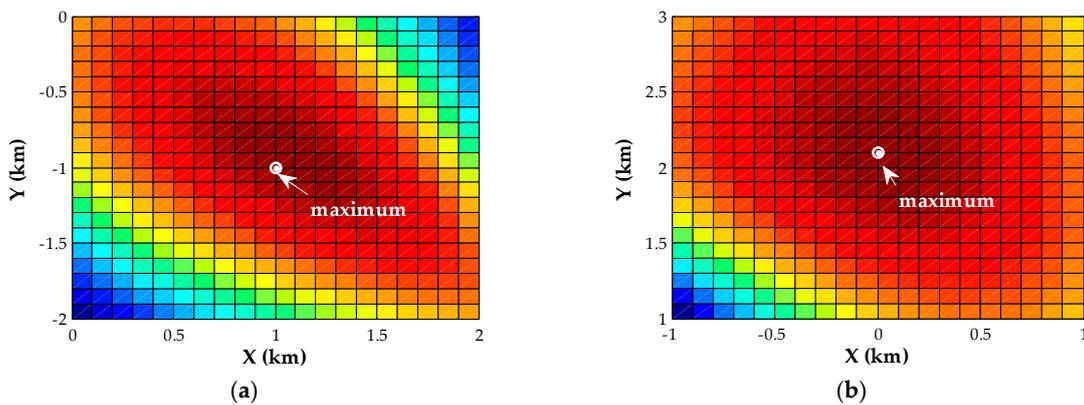


Figure 2. Evaluation of the inverse cost function over the area around the true positions of near-field sources, where different colors represent different magnitudes of amplitude. (a) Source 1; (b) Source 2.

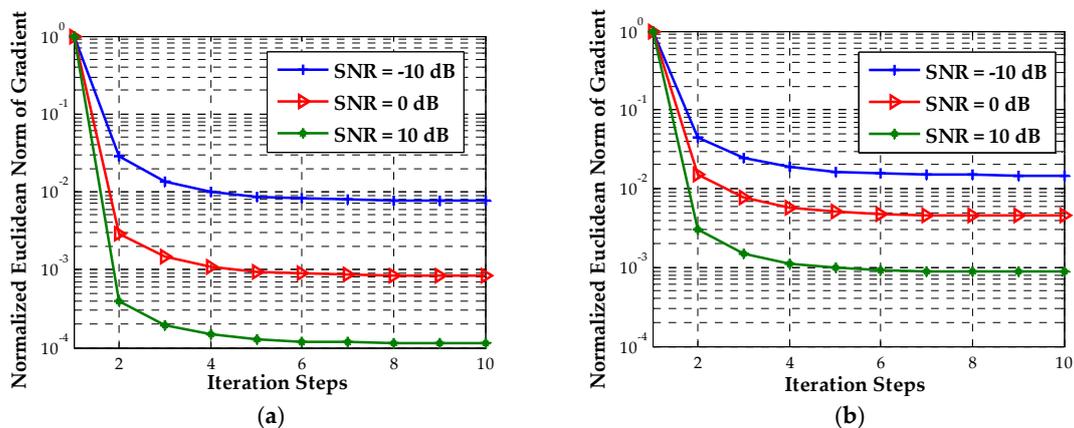


Figure 3. The normalized Euclidean norm of gradient versus iteration steps. (a) Source 1; (b) Source 2.

In the second example, for the simulation conditions given above, we roughly evaluate the computer running time of each of the prescribed location algorithms based on an average of

1000 estimates, when the SNR is 0 dB. The computer machine used is a ThinkPad laptop equipped with a 2.50-GHz Intel Core CPU and 8 GB RAM. The results are listed in Table 2. To make our results as comprehensive as possible, we also implement the fine search through the Nelder–Mead simplex search approach [29], which is a kind of local search method, and compare its average running time with that of the proposed Newton-type iterative method. We note that the complexity of our algorithm using the exhaustive grid search is larger than those of other approaches. As expected, our iterative method is the most computationally attractive among the six algorithms, and it accomplishes the location in about two thirds of the time required for the Nelder–Mead simplex search. Compared with the exhaustive search implementation, the running time of our iterative method is reduced by approximately ten times. This result agrees with the analysis in Section 3.3.

Table 2. Average runtime for near-field sources. FD, frequency domain; TD, time domain; NC, noncircular.

Method	Runtime (s)
Proposed DPD (Newton-type Iterative Method)	0.0818
Proposed DPD (Exhaustive Grid Search)	0.7802
Proposed DPD (Nelder–Mead Simplex Search)	0.1231
FD-DPD	0.5274
NC TD-DPD	0.5070
Two-step	0.3120

The third example will assess the accuracy of our algorithm for the near-field sources as shown in Figure 1. We plot the RMSE curves of the proposed DPD algorithms and other positioning algorithms versus SNR in Figure 4. It is obvious that the proposed DPD algorithms significantly outperform other comparison algorithms at low SNRs. As the SDF-based DPD in the time domain exploits only the DOA information, it exhibits the lowest accuracy even when the SNR is sufficiently high. These results indicate that both the noncircularity and the TOA information are helpful for enhancing accuracy. Furthermore, since the proposed DPD methods and two-step approach utilize the location information embedded in DOAs and TOAs for noncircular sources, the improved localization performance with respect to the traditional two-step method reveals the advantage of the DPD technique. As the RMSE curves of the proposed estimator using the Newton-type iterative method and that using Nelder–Mead simplex search overlap with each other, we provide their RMSE values in Table 3. It can be observed that the difference of the accuracy between two methods is not more than 0.001 km.

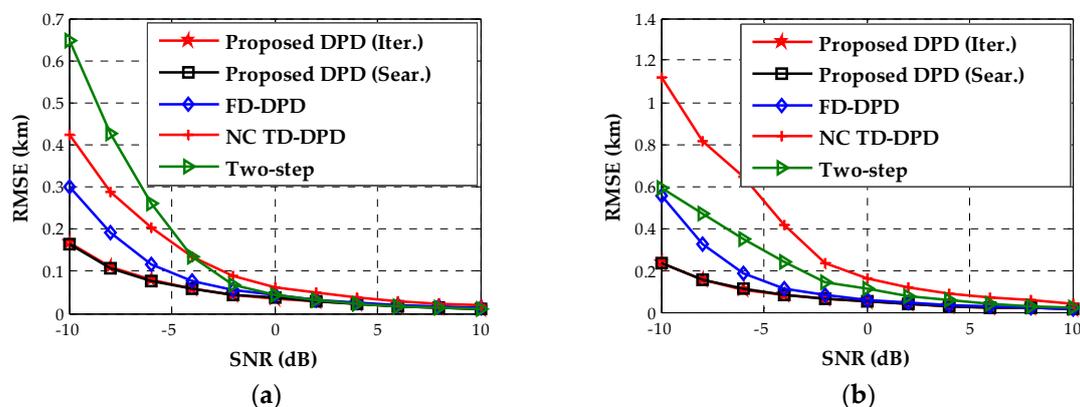


Figure 4. The estimated RMSEs versus SNR for near-field sources. (a) Source 1; (b) Source 2.

Table 3. The estimated RMSEs of the proposed DPDs using the Newton-type iterative method and Nelder–Mead simplex search for near-field sources (km).

Source	Method	SNR (dB)					
		−10	−6	−2	2	6	10
Source 1	Newton-type Iterative Method	0.167	0.079	0.049	0.028	0.017	0.011
	Nelder–Mead Simplex Search	0.166	0.078	0.050	0.029	0.017	0.012
Source 2	Newton-type Iterative Method	0.257	0.109	0.063	0.038	0.024	0.015
	Nelder–Mead Simplex Search	0.258	0.110	0.063	0.039	0.023	0.016

In the fourth example, we simulate the location performance with the number of sections K varying from 20 to 100 when the SNR is 0 dB. The location geometry corresponds to that in the foregoing examples. As shown in Figure 5, the performance of our algorithm remains promising in the range of different numbers of sections. When the sample number becomes smaller, this performance improvement turns out to be more prominent.

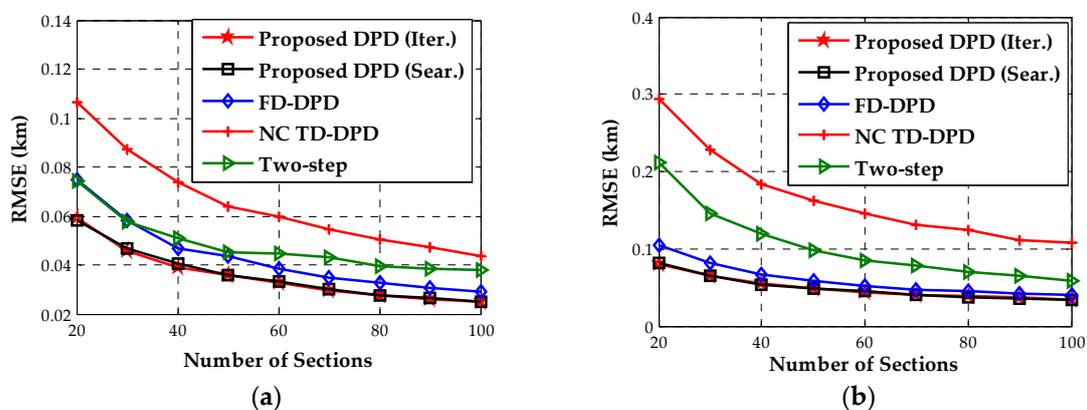


Figure 5. The estimated RMSEs versus the number of sections for near-field sources. (a) Source 1; (b) Source 2.

In the fifth example, we study the location accuracy when the distance between the two transmitters changes. The two transmitters are assumed to be placed at $p_1 = [0, -d/2]^T$ (km) and $p_2 = [0, d/2]^T$ (km), respectively. The SNR is as high as 10 dB. As d increases from 0.8 km to 2 km, we depict the RMSEs of the proposed DPDs and the existing SDF-based DPDs. Since the two-step approach can hardly separate the two transmitters with small distances, we do not show its RMSE curves. It can be seen from Figure 6 that our algorithms performs considerably better than other DPDs when the two sources are closely located. Specifically, the SDF-based DPD in the time domain fails to localize the second transmitter in the case of $d < 1.6$ (km). This demonstrates that our algorithm exhibits superior resolution compared to the existing SDF-based DPDs.

In this example, we consider two far-field transmitters located at $p_1 = [-12, -20]^T$ (km) and $p_2 = [-20, -10]^T$ (km) as illustrated in Figure 7. To examine the initial positions of the far-field sources obtained by the coarse search with a 0.1-km resolution, we compute the inverse cost function of the proposed estimator over the candidate positions with a grid step size of 0.1 km for different SNRs. The results for SNR = 0 dB are shown in Figure 8. We note that the resolution of this coarse search is sufficiently high to resolve the far-field sources and thus to obtain the effective initialization.

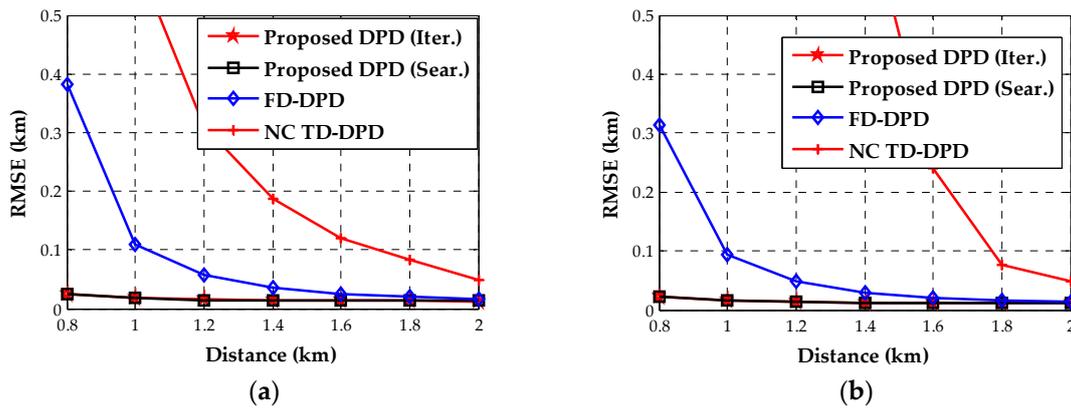


Figure 6. The estimated RMSEs for different distances of near-field sources. (a) Source 1; (b) Source 2.

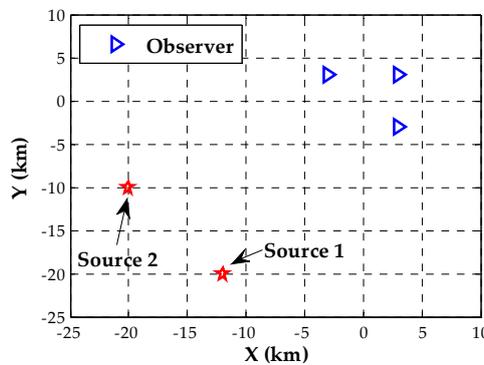


Figure 7. Scenario of three observers and two far-field sources placed on the ground.

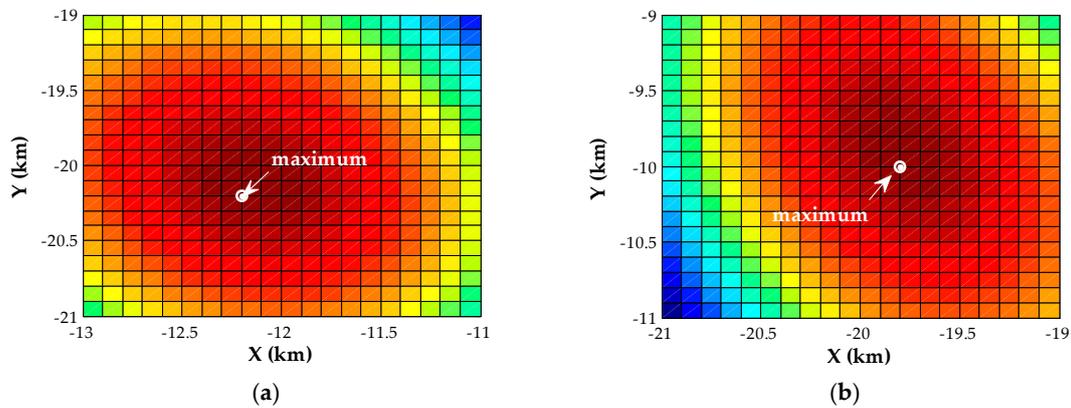


Figure 8. Evaluation of the inverse cost function over the area around the true positions of far-field sources, where different colors represent different magnitudes of amplitude. (a) Source 1; (b) Source 2.

Finally, we come to examine the running time and location accuracy of the proposed algorithms for the far-field sources as shown in Figure 7. When the SNR is 0 dB, we evaluate the average running time of each location algorithm. The corresponding results are shown in Table 4, which confirms that the devised iterative method is less complex than other methods. Especially, in contrast to the implementation of the Nelder–Mead simplex search, our Newton-type iterative method serves as a faster local search procedure. Under the same geometry, Figure 9 displays the location accuracy for the two far-field sources versus SNR, indicating that the proposed algorithms still hold the lowest RMSEs. When the SNR is below 0 dB, the RMSEs of the proposed method are much lower than those of other location algorithms by at least 0.5 km. Moreover, we notice that the performance of

the two-step approach (where the DOA and TOA estimations are designed for noncircular signals in the first step) is severely degraded at low SNRs, whereas it performs better than the SDF-based DPD in the frequency domain when the SNR is higher than 5 dB. Comparing this result with that in Figure 4, we find that the exploitation of the noncircular property is more crucial to the performance enhancement for the far-field sources than for the near-field sources. Besides, we show the RMSEs of the proposed estimators using the Newton-type iterative method and using the Nelder–Mead simplex search in Table 5. It can be seen that the accuracies of these two local search methods have no significant difference for the far-field sources.

Table 4. Average runtime for far-field sources.

Method	Runtime (s)
Proposed DPD (Newton-type Iterative Method)	0.0856
Proposed DPD (Exhaustive Grid Search)	0.7869
Proposed DPD (Nelder–Mead Simplex Search)	0.1330
FD-DPD	0.5144
NC TD-DPD	0.5072
Two-step	0.3132

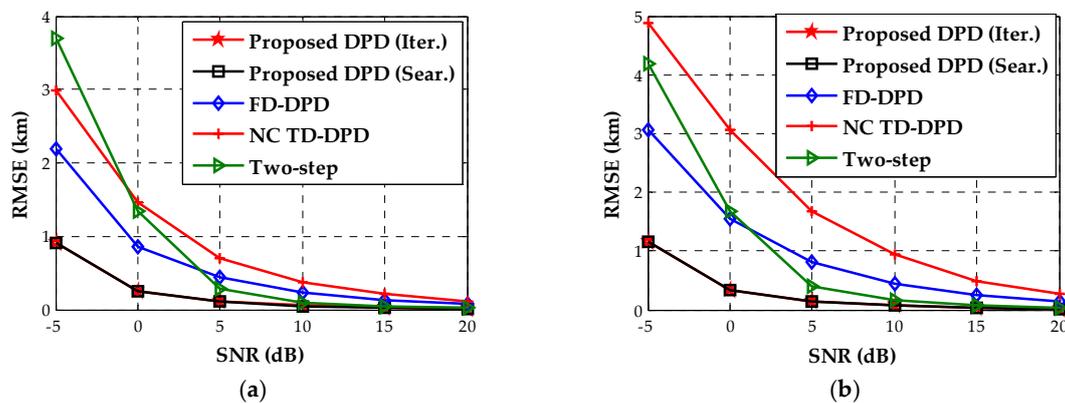


Figure 9. The estimated RMSEs versus SNR for far-field sources. (a) Source 1; (b) Source 2.

Table 5. The estimated RMSEs of the proposed DPDs using the Newton-type iterative method and Nelder–Mead simplex search for far-field sources (km).

Source	Method	SNR (dB)					
		−5	0	5	10	15	20
Source 1	Newton-type Iterative Method	0.935	0.243	0.108	0.053	0.029	0.016
	Nelder–Mead Simplex Search	0.933	0.250	0.101	0.050	0.029	0.016
Source 2	Newton-type Iterative Method	1.168	0.325	0.135	0.068	0.035	0.020
	Nelder–Mead Simplex Search	1.160	0.324	0.141	0.068	0.036	0.021

Additionally, the results indicate that the location accuracy of the proposed DPD using the Newton-type iterative method is approximately the same as those of the Nelder–Mead simplex search and the exhaustive grid search. Considering that the running time of the Newton-type iterative method is much shorter than those of other methods (Tables 2 and 4), this iterative method provides an efficient way to solve our DPD problem.

5. Conclusions

This study has investigated an efficient single-step localization method for noncircular sources received by widely-separated arrays. Through deriving and exploiting the potential of strictly noncircular signals in the frequency domain, the proposed algorithm fuses all the extended noise subspaces of all frequency components for the locations of multiple noncircular sources directly without estimating DOAs and TOAs. Relying on a unitary transformation, we have derived a cost function for each transmitter position, which is formulated as the smallest eigenvalue of a symmetric real-valued matrix. To further reduce the computation load of the exhaustive search for solving this cost function, we have devised a Newton-type iterative method based on the matrix Eigen-perturbation theory. Note that this iterative method can be easily extended to solve any estimator whose cost function is the eigenvalue of a specified matrix. Simulation results demonstrate that the proposed location method using Newton-type iteration is computationally efficient. Its running time is approximately one tenth of that of the exhaustive grid search and is nearly two thirds of the running time of the Nelder–Mead simplex search. Furthermore, our method shows greater performance robustness at low SNRs and under poor location geometry, compared with the conventional two-step approach and existing SDF DPD algorithms. Specifically, when the SNR is below 0 dB, the RMSEs of the proposed method are much lower than those of the existing SDF DPDs by at least 0.5 km for the simulated far-field sources.

To further improve the DPD performance and obtain the optimal location accurately and quickly, it is a possible direction to design a hybrid method combining an evolutionary algorithm and a fast local search approach for solving the ML-based DPD problem.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (Grant No. 61201381, Grant No. 61401513 and Grant No. 61772548), the China Postdoctoral Science Foundation (Grant No. 2016M592989), the Outstanding Youth Foundation of Information Engineering University (Grant No. 2016603201) and the Self-Topic Foundation of Information Engineering University (Grant No. 2016600701).

Author Contributions: Jiexin Yin and Ding Wang derived and developed the method. Jiexin Yin and Ying Wu conceived of and designed the simulations. Jiexin Yin performed the simulations. Ding Wang analyzed the results. Jiexin Yin wrote the paper.

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

Appendix A

Derivation of the expression for $\dot{\mathbf{X}}_p(\mathbf{p})$:

According to the definition of $\dot{\mathbf{X}}_p(\mathbf{p})$, it is computed by:

$$\dot{\mathbf{X}}_p(\mathbf{p}) = \frac{\partial \mathbf{x}(\mathbf{p})}{\partial \mathbf{p}^T} = \frac{\partial \text{vec}\{\mathbf{X}(\mathbf{p})\}}{\partial \mathbf{p}^T} = \frac{\partial \text{vec}\{\text{Re}\{\mathbf{Q}'(\mathbf{p})\}\}}{\partial \mathbf{p}^T}. \quad (\text{A1})$$

For convenience, we restate the expression for $\mathbf{Q}'(\mathbf{p})$ here:

$$\mathbf{Q}'(\mathbf{p}) = \sum_{j=0}^{J-1} \mathbf{T} \tilde{\mathbf{T}}^H(j, \mathbf{p}) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j) \tilde{\mathbf{T}}(j, \mathbf{p}) \mathbf{T}^H. \quad (\text{A2})$$

As the operations $\text{Re}\{\cdot\}$, $\text{vec}\{\cdot\}$ and taking the derivative can be processed out of order, as well as this ends up with the same result, we first come to derive $\frac{\partial \text{vec}\{\mathbf{Q}'(\mathbf{p})\}}{\partial \mathbf{p}^T}$.

Based on (A2) and using $\text{vec}\{\mathbf{ABC}\} = (\mathbf{C}^T \otimes \mathbf{A})\text{vec}\{\mathbf{B}\}$ (see [26]), we can express $\text{vec}\{\mathbf{Q}'(\mathbf{p})\}$ as:

$$\text{vec}\{\mathbf{Q}'(\mathbf{p})\} = \sum_{j=0}^{J-1} \left(\mathbf{T}^* \otimes (\tilde{\mathbf{T}}^H(j, \mathbf{p}) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j)) \right) \text{vec}\{\tilde{\mathbf{T}}(j, \mathbf{p})\}, \quad (\text{A3})$$

and it can be also written as:

$$\begin{aligned} \text{vec}\{\mathbf{Q}'(\mathbf{p})\} &= \sum_{j=0}^{J-1} ((\mathbf{T}^* \tilde{\Gamma}^T(j, \mathbf{p}) \mathbf{U}_k^{n*}(j) \mathbf{U}_k^{nT}(j)) \otimes \mathbf{T}) \text{vec}\{\tilde{\Gamma}^H(j, \mathbf{p})\} \\ &= \sum_{j=0}^{J-1} ((\mathbf{T}^* \tilde{\Gamma}^T(j, \mathbf{p}) \mathbf{U}_k^{n*}(j) \mathbf{U}_k^{nT}(j)) \otimes \mathbf{T}) \mathbf{P} \text{vec}\{\tilde{\Gamma}^*(j, \mathbf{p})\}, \end{aligned} \tag{A4}$$

where \mathbf{P} is a vec-permutation matrix that transforms $\text{vec}\{\tilde{\Gamma}^*(j, \mathbf{p})\}$ to $\text{vec}\{\tilde{\Gamma}^H(j, \mathbf{p})\}$, i.e., $\text{vec}\{\tilde{\Gamma}^H(j, \mathbf{p})\} = \mathbf{P} \text{vec}\{\tilde{\Gamma}^*(j, \mathbf{p})\}$.

Define:

$$\dot{\tilde{\Gamma}}_p(j, \mathbf{p}) = \frac{\partial \text{vec}\{\tilde{\Gamma}^*(j, \mathbf{p})\}}{\partial \mathbf{p}^T}. \tag{A5}$$

Then, combing the results in (A3) and (A4), we can calculate the gradient of $\text{vec}\{\mathbf{Q}'(\mathbf{p})\}$ by:

$$\frac{\partial \text{vec}\{\mathbf{Q}'(\mathbf{p})\}}{\partial \mathbf{p}^T} = \mathbf{w}_1(\mathbf{p}) + \mathbf{w}_2(\mathbf{p}), \tag{A6}$$

where:

$$\mathbf{w}_1(\mathbf{p}) = \sum_{j=0}^{J-1} \left(\mathbf{T}^* \otimes (\mathbf{T} \tilde{\Gamma}^H(j, \mathbf{p}) \mathbf{U}_k^n(j) \mathbf{U}_k^{nH}(j)) \right) \dot{\tilde{\Gamma}}_p(j, \mathbf{p}) \tag{A7}$$

and:

$$\mathbf{w}_2(\mathbf{p}) = \left((\mathbf{T}^* \tilde{\Gamma}^T(j, \mathbf{p}) \mathbf{U}_k^{n*}(j) \mathbf{U}_k^{nT}(j)) \otimes \mathbf{T} \right) \mathbf{P} \dot{\tilde{\Gamma}}_p^*(j, \mathbf{p}). \tag{A8}$$

Finally, extracting the real part of (A6), we obtain:

$$\dot{\mathbf{X}}_p(\mathbf{p}) = \text{Re} \left\{ \frac{\partial \text{vec}\{\mathbf{Q}'(\mathbf{p})\}}{\partial \mathbf{p}^T} \right\} = \text{Re}\{\mathbf{w}_1(\mathbf{p}) + \mathbf{w}_2(\mathbf{p})\}, \tag{A9}$$

which completes our derivation.

Appendix B

Derivation of the expression for $\ddot{\mathbf{X}}_{pp}(\mathbf{p})$:

To obtain the explicit expression for $\ddot{\mathbf{X}}_{pp}(\mathbf{p})$, we first derive each $D \times D$ sub-block as shown in (53), i.e.,

$$\frac{\partial^2 [x(\mathbf{p})]_{2mL-2L+n}}{\partial \mathbf{p} \partial \mathbf{p}^T} = \frac{\partial^2 \text{Re}\{[\mathbf{Q}'(\mathbf{p})]_{n,m}\}}{\partial \mathbf{p} \partial \mathbf{p}^T} = \text{Re} \left\{ \frac{\partial^2 [\mathbf{Q}'(\mathbf{p})]_{n,m}}{\partial \mathbf{p} \partial \mathbf{p}^T} \right\}, \tag{A10}$$

where $n, m = 1, 2, \dots, 2L$.

Let \mathbf{v}_n be the $2L \times 1$ unit vector with the n -th element being one, and \mathbf{v}_m is defined similarly as \mathbf{v}_n . Therefore, the n, m -th entry of $\mathbf{Q}'(\mathbf{p})$ is given by:

$$[\mathbf{Q}'(\mathbf{p})]_{n,m} = \sum_{j=0}^{J-1} \mathbf{v}_n^T \mathbf{T} \tilde{\Gamma}^H(j, \mathbf{p}) \mathbf{U}_k^n(j) \mathbf{U}_k^{nH}(j) \tilde{\Gamma}(j, \mathbf{p}) \mathbf{T}^H \mathbf{v}_m. \tag{A11}$$

Denote $\dot{\tilde{\Gamma}}^d(j, \mathbf{p}) = \frac{\partial \tilde{\Gamma}(j, \mathbf{p})}{\partial p_d}$ and $\ddot{\tilde{\Gamma}}_{d_1 d_2}(j, \mathbf{p}) = \frac{\partial^2 \tilde{\Gamma}(j, \mathbf{p})}{\partial p_{d_1} \partial p_{d_2}}$, where p_d is the d -th element of \mathbf{p} . Using $\text{tr}\{ABCD\} = \text{vec}^T\{D^T\}(C^T \otimes A)\text{vec}\{B\}$ and $\text{tr}\{ABCD\} = \text{vec}^T\{D\}(A \otimes C^T)\text{vec}\{B^T\}$ (see [26]), we can compute the second-order derivative of $[Q'(\mathbf{p})]_{n,m}$ as follows:

$$\begin{aligned} & \frac{\partial^2 [Q'(\mathbf{p})]_{n,m}}{\partial p_{d_1} \partial p_{d_2}} \\ &= \sum_{j=0}^{J-1} \left\{ \begin{aligned} & \mathbf{v}_n^T \mathbf{T} \dot{\tilde{\Gamma}}_{d_1 d_2}^{\cdot H}(j, \mathbf{p}) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j) \tilde{\Gamma}(j, \mathbf{p}) \mathbf{T}^H \mathbf{v}_m + \mathbf{v}_n^T \mathbf{T} \dot{\tilde{\Gamma}}_{d_1}^{\cdot H}(j, \mathbf{p}) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j) \dot{\tilde{\Gamma}}_{d_2}(j, \mathbf{p}) \mathbf{T}^H \mathbf{v}_m + \\ & \mathbf{v}_n^T \mathbf{T} \dot{\tilde{\Gamma}}_{d_2}^{\cdot H}(j, \mathbf{p}) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j) \dot{\tilde{\Gamma}}_{d_1}(j, \mathbf{p}) \mathbf{T}^H \mathbf{v}_m + \mathbf{v}_n^T \mathbf{T} \tilde{\Gamma}^H(j, \mathbf{p}) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j) \ddot{\tilde{\Gamma}}_{d_2 d_1}(j, \mathbf{p}) \mathbf{T}^H \mathbf{v}_m \end{aligned} \right\} \\ &= \sum_{j=0}^{J-1} \left\{ \begin{aligned} & \mathbf{v}_n^T \mathbf{T} \dot{\tilde{\Gamma}}_{d_1 d_2}^{\cdot H}(j, \mathbf{p}) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j) \tilde{\Gamma}(j, \mathbf{p}) \mathbf{T}^H \mathbf{v}_m + \text{tr}\{\mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j) \dot{\tilde{\Gamma}}_{d_2}(j, \mathbf{p}) \mathbf{T}^H \mathbf{v}_m \mathbf{v}_n^T \mathbf{T} \dot{\tilde{\Gamma}}_{d_1}^{\cdot H}(j, \mathbf{p})\} + \\ & \text{tr}\{\mathbf{T}^H \mathbf{v}_m \mathbf{v}_n^T \mathbf{T} \dot{\tilde{\Gamma}}_{d_2}^{\cdot H}(j, \mathbf{p}) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j) \dot{\tilde{\Gamma}}_{d_1}(j, \mathbf{p})\} + \mathbf{v}_n^T \mathbf{T} \tilde{\Gamma}^H(j, \mathbf{p}) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j) \ddot{\tilde{\Gamma}}_{d_2 d_1}(j, \mathbf{p}) \mathbf{T}^H \mathbf{v}_m \end{aligned} \right\} \quad (A12) \\ &= \sum_{j=0}^{J-1} \left\{ \begin{aligned} & \mathbf{v}_m^T \mathbf{T}^* \tilde{\Gamma}^T(j, \mathbf{p}) \mathbf{u}_k^{n*}(j) \mathbf{u}_k^{nT}(j) \ddot{\tilde{\Gamma}}_{d_1 d_2}^*(j, \mathbf{p}) \mathbf{T}^T \mathbf{v}_n + \\ & \text{vec}^T\{\dot{\tilde{\Gamma}}_{d_1}^*(j, \mathbf{p})\} ((\mathbf{T}^T \mathbf{v}_n \mathbf{v}_m^T \mathbf{T}^*) \otimes (\mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j))) \text{vec}\{\dot{\tilde{\Gamma}}_{d_2}(j, \mathbf{p})\} + \\ & \text{vec}^T\{\dot{\tilde{\Gamma}}_{d_2}^*(j, \mathbf{p})\} ((\mathbf{T}^H \mathbf{v}_m \mathbf{v}_n^T \mathbf{T}) \otimes (\mathbf{u}_k^{n*}(j) \mathbf{u}_k^{nT}(j))) \text{vec}\{\dot{\tilde{\Gamma}}_{d_1}(j, \mathbf{p})\} + \\ & \mathbf{v}_n^T \mathbf{T} \tilde{\Gamma}^H(j, \mathbf{p}) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j) \ddot{\tilde{\Gamma}}_{d_1 d_2}(j, \mathbf{p}) \mathbf{T}^H \mathbf{v}_m \end{aligned} \right\} \end{aligned}$$

for $d_1, d_2 = 1, 2, \dots, D$. Note that the terms on the right side of (A12) are the d_1, d_2 -th entries of:

$$\mathbf{W}_1(\mathbf{p}) = \sum_{j=0}^{J-1} (\mathbf{I}_D \otimes (\mathbf{v}_m^T \mathbf{T}^* \tilde{\Gamma}^T(j, \mathbf{p}) \mathbf{u}_k^{n*}(j) \mathbf{u}_k^{nT}(j))) \ddot{\tilde{\Gamma}}_{pp}^*(j, \mathbf{p}) (\mathbf{I}_D \otimes \mathbf{T}^T \mathbf{v}_n), \quad (A13)$$

$$\mathbf{W}_2(\mathbf{p}) = \sum_{j=0}^{J-1} \dot{\tilde{\Gamma}}_p^{\cdot H}(j, \mathbf{p}) ((\mathbf{T}^T \mathbf{v}_n \mathbf{v}_m^T \mathbf{T}^*) \otimes (\mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j))) \dot{\tilde{\Gamma}}_p(j, \mathbf{p}), \quad (A14)$$

$$\mathbf{W}_2^T(\mathbf{p}) = \sum_{j=0}^{J-1} \dot{\tilde{\Gamma}}_p^{\cdot T}(j, \mathbf{p}) ((\mathbf{T}^H \mathbf{v}_m \mathbf{v}_n^T \mathbf{T}) \otimes (\mathbf{u}_k^{n*}(j) \mathbf{u}_k^{nT}(j))) \dot{\tilde{\Gamma}}_p^*(j, \mathbf{p}), \quad (A15)$$

and:

$$\mathbf{W}_3(\mathbf{p}) = \sum_{j=0}^{J-1} (\mathbf{I}_D \otimes (\mathbf{v}_n^T \mathbf{T} \tilde{\Gamma}^H(j, \mathbf{p}) \mathbf{u}_k^n(j) \mathbf{u}_k^{nH}(j))) \ddot{\tilde{\Gamma}}_{pp}(j, \mathbf{p}) (\mathbf{I}_D \otimes \mathbf{T}^H \mathbf{v}_m), \quad (A16)$$

respectively, where $\dot{\tilde{\Gamma}}_p(j, \mathbf{p})$ has been defined in (A5), and $\ddot{\tilde{\Gamma}}_{pp}(j, \mathbf{p})$ has the following structure:

$$\ddot{\tilde{\Gamma}}_{pp}(j, \mathbf{p}) = \begin{bmatrix} \ddot{\tilde{\Gamma}}_{11}(j, \mathbf{p}) & \cdots & \ddot{\tilde{\Gamma}}_{1D}(j, \mathbf{p}) \\ \vdots & \ddots & \vdots \\ \ddot{\tilde{\Gamma}}_{D1}(j, \mathbf{p}) & \cdots & \ddot{\tilde{\Gamma}}_{DD}(j, \mathbf{p}) \end{bmatrix}. \quad (A17)$$

In light of this finding, we can express $\frac{\partial^2 [Q'(\mathbf{p})]_{n,m}}{\partial \mathbf{p} \partial \mathbf{p}^T}$ in the matrix form as:

$$\frac{\partial^2 [Q'(\mathbf{p})]_{n,m}}{\partial \mathbf{p} \partial \mathbf{p}^T} = \mathbf{W}_1(\mathbf{p}) + \mathbf{W}_2(\mathbf{p}) + \mathbf{W}_2^T(\mathbf{p}) + \mathbf{W}_3(\mathbf{p}). \quad (A18)$$

Then, substituting (A18) back into (A10) leads to:

$$\frac{\partial^2 [x(\mathbf{p})]_{2mL-2L+n}}{\partial \mathbf{p} \partial \mathbf{p}^T} = \text{Re}\{\mathbf{W}_1(\mathbf{p}) + \mathbf{W}_2(\mathbf{p}) + \mathbf{W}_2^T(\mathbf{p}) + \mathbf{W}_3(\mathbf{p})\}. \quad (A19)$$

Consequently, $\ddot{\mathbf{X}}_{pp}(\mathbf{p})$ can be constructed by applying (A19) to (53).

References

1. Qian, G.B.; Li, L.P.; Luo, M.G. On the blind channel identifiability of MIMO-STBC systems using noncircular complex fastICA algorithm. *Circuits Syst. Signal Process.* **2014**, *33*, 1859–1881. [[CrossRef](#)]
2. Yang, X.M.; Li, G.J.; Zheng, Z. A novel covariance based noncircular sources direction finding method under impulsive noise environment. *Signal Process.* **2014**, *98*, 252–262.
3. Hassen, S.B.; Bellili, F.; Samet, A. DOA estimation of temporally and spatially correlated narrowband noncircular sources in spatially correlated white noise. *IEEE Trans. Signal Process.* **2011**, *59*, 4108–4121. [[CrossRef](#)]
4. Xie, J.; Tao, H.H.; Rao, X. Efficient method of passive location for near-field noncircular sources. *IEEE Antennas Wirel. Propag. Lett.* **2015**, *14*, 1223–1226. [[CrossRef](#)]
5. Abeida, H.; Delmas, J.-P. MUSIC-like estimation of direction of arrival for noncircular sources. *IEEE Trans. Signal Process.* **2006**, *54*, 2678–2690. [[CrossRef](#)]
6. Yin, J.X.; Wu, Y.; Wang, D. An auto-calibration method for spatially and temporally correlated noncircular sources in unknown noise fields. *Multidimens. Syst. Signal Process.* **2016**, *27*, 511–539. [[CrossRef](#)]
7. Wen, F.; Wan, Q.; Liu, Y.P. Passive time delay estimation for complex noncircular signals. In Proceedings of the IEEE International Symposium on Electronics and Telecommunications, Timisoara, Romania, 15–16 November 2012.
8. Wax, M.; Kailath, T. Decentralized processing in sensor arrays. *IEEE Trans. Acoust. Speech Signal Process.* **1985**, *33*, 1123–1129. [[CrossRef](#)]
9. Stoica, P.; Nehorai, A.; Söderström, T. Decentralized array processing using the MODE algorithm. *Circuits Syst. Signal Process.* **1995**, *14*, 17–38. [[CrossRef](#)]
10. Shen, J.Y.; Molisch, A.F.; Salmi, J. Accurate passive location estimation using TOA measurements. *IEEE Trans. Wirel. Commun.* **2011**, *11*, 253–257.
11. Wang, Y.L.; Wu, Y. An efficient semidefinite relaxation algorithm for moving source localization using TDOA and FDOA measurements. *IEEE Commun. Lett.* **2017**, *21*, 80–83. [[CrossRef](#)]
12. Wax, M.; Kailath, T. Optimum localization of multiple sources by passive arrays. *IEEE Trans. Acoust. Speech Signal Process.* **1983**, *31*, 1210–1217. [[CrossRef](#)]
13. Li, J.; Yang, L.; Guo, F. Coherent summation of multiple short-time signals for direct positioning of a wideband source based on delay and Doppler. *Digit. Signal Process.* **2016**, *48*, 58–70. [[CrossRef](#)]
14. Weiss, A.J. Direct position determination of narrowband radio frequency transmitters. *IEEE Signal Process. Lett.* **2004**, *11*, 513–516. [[CrossRef](#)]
15. Tirer, T.; Weiss, A.J. High resolution direct position determination of radio frequency sources. *IEEE Signal Process. Lett.* **2016**, *23*, 192–196. [[CrossRef](#)]
16. Weiss, A.J.; Amar, A. Direct position determination of multiple radio signals. *EURASIP J. Appl. Signal Process.* **2005**, *1*, 1–13. [[CrossRef](#)]
17. Amar, A.; Weiss, A.J. A decoupled algorithm for geolocation of multiple emitters. *Signal Process.* **2007**, *87*, 2348–2359. [[CrossRef](#)]
18. Huang, Z.Y.; Wu, J. Multi-array data fusion based direct position determination algorithm. In Proceedings of the IEEE Seventh International Symposium on Computational Intelligence and Design, Hangzhou, China, 13–14 December 2014; pp. 121–124.
19. Yin, J.X.; Wu, Y.; Wang, D. Direct position determination of multiple noncircular sources with a moving array. *Circuits Syst. Signal Process.* **2017**, *36*, 1–27. [[CrossRef](#)]
20. Ke, W.; Wang, T.T.; Wu, L.N. Multi-target direct location via approximate l_0 norm minimization. *Electron. Lett.* **2012**, *48*, 1498–1500. [[CrossRef](#)]
21. Oispuu, M.; Nickel, U. Direct detection and position determination of multiple sources with intermittent emission. *Signal Process.* **2010**, *90*, 3056–3064. [[CrossRef](#)]
22. Tzoreff, E.; Weiss, A.J. Expectation-maximization algorithm for direct position determination. *Signal Process.* **2016**, *133*, 32–39. [[CrossRef](#)]
23. Haardt, M.; Römer, F. Enhancements of unitary ESPRIT for non-circular sources. In Proceedings of the IEEE International Conference on Acoustics, Montreal, QC, Canada, 17–21 May 2004; pp. 101–104.
24. Kumar, M.; Singh, A.K.; Srivastava, A. Various Newton-type iterative methods for solving nonlinear equations. *J. Egypt. Math. Soc.* **2013**, *21*, 334–339. [[CrossRef](#)]

25. Wang, D.; Wu, Y. Statistical performance analysis of direct position determination method based on Doppler shifts in presence of model errors. *Multidimens. Syst. Signal Process.* **2015**, *28*, 1–34. [[CrossRef](#)]
26. Zhang, X.D. *Matrix Analysis and Application*; Tsinghua University Press: Beijing, China, 2004.
27. Magnus, J.R.; Neudecker, H. *Matrix Differential Calculus with Application in Statistics and Econometrics*; John Wiley & Sons, Inc.: Essex, UK, 1988.
28. Wang, D. Sensor array calibration in presence of mutual coupling and gain/phase errors by combining the spatial-domain and time-domain waveform information of the calibration sources. *Circuits Syst. Signal Process.* **2013**, *32*, 1257–1292. [[CrossRef](#)]
29. Nelder, J.A.; Mead, R. A simplex method for function minimization. *Comput. J.* **1965**, *7*, 308–313. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).