Subspace-Based Method for Direction Estimation of Coherent Signals with Arbitrary Linear Array

Xiao Chen[†], *Jingmin Xin*[†‡], *Weiliang Zuo*[†‡], *Jie Li*[†‡], *Nanning Zheng*[†‡], *and Akira Sano*[§]
[†] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China
[‡] National Engineering Laboratory for Visual Information Processing and Applications, Xi'an Jiaotong University, Xi'an 710049, China
[§] Department of System Design Engineering, Keio University, Yokohama 223-8522, Japan

***Abstract*—In this paper, we propose an interpolated computationally efficient subspace-based method without eigendecomposition (ISUMWE) for the direction-of-arrivals (DOA) estimation of narrowband coherent signals in arbitrary linear arrays. ISUMWE estimates DOA based on the outputs of the virtual array by using an interpolation transform technique. Therefore, it overcomes the common restriction of uniform linear array (ULA) geometry when estimating coherent signals and becomes suitable for more general array geometry than ordinary methods. Meanwhile, the coherency of incident signals is decorrelated through a linear operation of a matrix formed from the cross-correlations between some sensor data in a designed virtual array which can be computed from the linear transformation of sensor data in the real array, where the effect of additive noise is eliminated. Consequently, the DOA can be estimated without performing eigendecomposition, and the noise pre-whitening which is required in traditional interpolation procedure can be avoided. As a result, the ISUMWE extends the application of the original subspace-based method without eigendecomposition (SUMWE) into arbitrary linear array with high accuracy and low computational complexity. The numerical results demonstrate the validity of the proposed method.**

***Index Terms*—Direction-of-arrival, coherent signals, computationally efficient , array interpolation**

I. Introduction

The direction-of-arrivals (DOA) estimation of incident signals is a fundamental problem in array processing, and it has received considerable attention of many researchers [1], [2]. In practice, multipath propagation is usually encountered due to various reflections, and there are many DOA estimation methods proposed for coherent signals [3]–[7]. The beamforming is one of the oldest ideas for DOA estimation, and perhaps the most well known one is the Capon beamformer [8], [9], which can be applied to the array with arbitrary geometrical configurations and has low computational complexity, while its performance degrades significantly when the number of snapshots is small and/or the signal to noise ratio (SNR) is low [10]. Although some modified techniques are proposed to solve the performance degradation and geometrical restriction of coherent signals estimation [11], their estimation performance still needs to be improved. Besides, subspace-based DOA estimation methods such as multiple signal classification (MUSIC) [12] are well studied due to its good estimation accuracy. However, these traditional subspace-based methods are based on eigendecomposition of an array covariance matrix, which is very complicated and usually cannot meet the calculation requirements when the number of sensors is large or the real-time estimation is required. To avoid the computationally intensive operations of eigendecomposition, many new subspace-based methods have been proposed, such as the bearing estimation without eigendecomposition (BEWE) [13], [14], [15], orthonormal propagator method (OPM) [16], [17], subspace methods without eigendecomposition (SWEDE) [18], subspace-based method without eigendecomposition (SUMWE) [19] and so on. Among them, the SUMWE method can reduce the computational complexity to a large extent and reach the accuracy level of traditional subspace method. However, the use of SUMWE is restricted to the uniform linear array. Unfortunately, the ideal uniform linear array is difficult to obtain in practical applications. A DOA estimation method for arbitrary array geometries is proposed in [20], but it is based on noise-free model and limited in low noise case. Capon beamformer is modified and extended to arbitrary linear array without high computational complexity eigendecomposition in [21], however it's still puzzled by high computation when its parameter m is high.

Here, we propose an interpolated computationally efficient subspace-based method without eigendecomposition (ISUMWE) for the direction-of-arrivals (DOA) estimation of coherent signals in arbitrary linear arrays. Based on the interpolation transform technique [22] [23] and benefited from the design of virtual array, the virtual cross-correlations data can be computed from the linear transformation of real cross-correlations data, and the noise pre-whitening which is required in traditional interpolation procedure can be avoided. As a result, the restriction of uniform linear array geometry required by the SUMWE is overcome, and the SUMWE is extended to arbitrary linear array successfully.

II. Problem formulation

As shown in Fig. 1, we consider an arbitrary linear array composed of M identical and omnidirectional sensors, and suppose that p narrowband signals $\{s_i(n)\}_{i=1}^{p}$ with the center frequency f_0 impinge on the array from far-field along the distinct direction $\{\theta_i\}_{i=1}^{p}$. The received noisy array data at the nth snapshot can be expressed as

$$\boldsymbol{y}(n) = \boldsymbol{A}(\theta)\boldsymbol{s}(n) + \boldsymbol{\omega}(n) \quad (1)$$

where $\boldsymbol{y}(n)$, $\boldsymbol{s}(n)$ and $\boldsymbol{\omega}(n)$ are the vectors of the received noisy signals, the incident signals, and additive noise respec-

This work was supported in part by the National Natural Science Foundation of China under Grant 61790563 and 61671373.

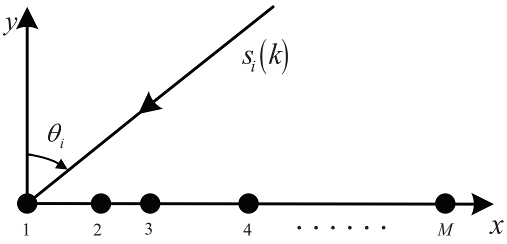

Fig. 1. The geometric configuration of the arbitrary linear array

tively given by $\boldsymbol{y}(n) \triangleq [y_1(n), y_2(n), \ldots, y_M(n)]^T$, $\boldsymbol{s}(n) \triangleq [s_1(n), s_2(n), \ldots, s_p(n)]^T$ and $\boldsymbol{\omega}(n) \triangleq [\omega_1(n), \omega_2(n), \ldots, \omega_M(n)]^T$, $\boldsymbol{A}(\theta)$ is the array response matrix given by $\boldsymbol{A}(\theta) \triangleq [\boldsymbol{a}(\theta_1), \boldsymbol{a}(\theta_2), \ldots, \boldsymbol{a}(\theta_p)]$, and $(\cdot)^T$ denotes transpose.

In this paper, we make the following assumptions: 1) Given a set of distinct $\{\theta_1, \theta_2, \cdots \theta_p\}$, the $\{\boldsymbol{a}(\theta_1), \boldsymbol{a}(\theta_2), \ldots, \boldsymbol{a}(\theta_p)\}$ are linearly independent. 2) The incident signals $\{s_i(n)\}$ are all coherent, which can be expressed as complex multiples of a common signal $s_1(n)$: $s_i(n) = \xi_i s_1(n)$ where ξ_i is a complex attenuation coefficient of ith signal. 3) The incident signal $s_1(n)$ is temporally complex Gaussian random process with zero-mean and the variance given by $E\{s_1(n)s_1^*(t)\} = r_s\delta_{n,t}$, $E\{s_1(n)s_1(t)\} = 0, \forall n, t$ where $E\{\cdot\}$, $(\cdot)^*$ and $\delta_{n,t}$ denote the expectation, the complex conjugate, and the Kronecker delta respectively. 4) The additive noises $\{\omega_i(n)\}$ are temporally and spatially complex white Gaussian random process with zero-mean and variance σ^2, where $E\{\boldsymbol{\omega}(n)\boldsymbol{\omega}^H(t)\} = \sigma^2\boldsymbol{I}_M\delta_{n,t}$, $E\{\boldsymbol{\omega}(n)\boldsymbol{\omega}^T(t)\} = \boldsymbol{O}_{M\times M}, \forall n, t$, while $\boldsymbol{I}_m$, $\boldsymbol{O}_{m\times q}$ and $(\cdot)^H$ indicate the $m \times m$ identity matrix, the $m \times q$ null matrix, and the Hermitian transpose. In addition, the additive noise is uncorrelated with the incident signals. 5) The number of incident signals p is known and it satisfies the relation $M > p$.

This paper concentrates on estimating the DOA $\{\theta_i\}_{i=1}^p$ of coherent signals from the noisy array data $\{\boldsymbol{y}(n)\}_{n=1}^N$ in an arbitrary linear array.

III. ISUMWE METHOD

A. Interpolation Transformation

We divide the array scanning area into L_s sectors and implement interpolation transform on each sector separately. Take lth sector: $\Theta_l : [\theta_l^{(1)}\theta_l^{(2)}]$ as example, we define a set of angles $\boldsymbol{\Theta}_l$ with $\Delta\theta$ intervals:

$$\boldsymbol{\Theta}_l = [\theta_l^{(1)}\theta_l^{(1)} + \Delta\theta, \theta_l^{(1)} + 2\Delta\theta, \cdots, \theta_l^{(2)}] \tag{2}$$

As shown in Fig. 2, we design the the virtual array as an uniformly spaced linear array with the spacing d to make sure the effective implementation of subarray averaging and polynomial rooting. In addition, the first sensor and the last sensor of the virtual array have same location with that of the real array, which is the basis for subsequent derivation. We divide real array into two subarrays: real subarray α which consists of the first $M-1$ sensors and real subarray β which consists of the last $M-1$ sensors. Similarly, virtual array is divided to: virtual subarray α which consists of the first $M_v - 1$ sensors and virtual subarray β which consists of the last $M_v - 1$ sensors. Empirically, we hope that the difference between the position of the real array and that of the virtual array is as small as possible [22]. And we often adjust the total aperture of virtual array to be approximately equal to that of the real array so as to decrease the interpolation error.

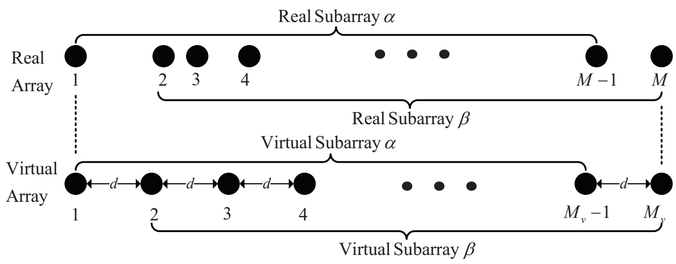

Fig. 2. The design of virtual array

The response matrices for set $\boldsymbol{\Theta}_l$ of real subarray α and virtual subarray α can be computed as:

$$\begin{aligned} \boldsymbol{A}_{\alpha l} &= [\boldsymbol{a}_\alpha(\theta_l^{(1)}), \boldsymbol{a}_\alpha(\theta_l^{(1)} + \Delta\theta), \cdots, \boldsymbol{a}_\alpha(\theta_l^{(2)})] \\ \bar{\boldsymbol{A}}_{\alpha l} &= [\bar{\boldsymbol{a}}_\alpha(\theta_l^{(1)}), \bar{\boldsymbol{a}}_\alpha(\theta_l^{(1)} + \Delta\theta), \cdots, \bar{\boldsymbol{a}}_\alpha(\theta_l^{(2)})] \end{aligned} \tag{3}$$

where $\boldsymbol{a}_\alpha(\theta) = [1, e^{-j(2\pi/\lambda)D_2 \sin\theta}, \cdots, e^{-j(2\pi/\lambda)D_{M-1}\sin\theta}]^T$; D_i represents the distance between the first sensor and the ith sensor in the real array; $\bar{\boldsymbol{a}}_\alpha(\theta) = [1, e^{-j(2\pi/\lambda)d\sin\theta}, \cdots, e^{-j(2\pi/\lambda)(M_v-2)d\sin\theta}]^T$. There exists a constant matrix $\boldsymbol{B}_{\alpha l}$ which satisfies that:

$$\boldsymbol{B}_{\alpha l}\boldsymbol{A}_{\alpha l} = \bar{\boldsymbol{A}}_{\alpha l} \tag{4}$$

Response matrices $\boldsymbol{A}_{\alpha l}$ and $\bar{\boldsymbol{A}}_{\alpha l}$ are known since the $\boldsymbol{\Theta}_l$ is certain. Therefore, the interpolation matrix $\boldsymbol{B}_{\alpha l}$ can be estimated as the least squares solution of (4). And the interpolation error of $\boldsymbol{A}_{\alpha l}$ and $\bar{\boldsymbol{A}}_{\alpha l}$ is defined as [22]:

$$\varepsilon_{\alpha l} = \frac{\left\|\bar{\boldsymbol{A}}_{\alpha l} - \boldsymbol{B}_{\alpha l}\boldsymbol{A}_{\alpha l}\right\|}{\left\|\bar{\boldsymbol{A}}_{\alpha l}\right\|} \tag{5}$$

Similarly, the interpolation matrix $\boldsymbol{B}_{\beta l}$ can be calculated by solving the following least squares problem:

$$\boldsymbol{B}_{\beta l}\boldsymbol{A}_{\beta l} = \bar{\boldsymbol{A}}_{\beta l} \tag{6}$$

where $\boldsymbol{A}_{\beta l}$ is the response matrix of real subarray β, giving by $\boldsymbol{A}_{\beta l} = [\boldsymbol{a}_\beta(\theta_l^{(1)}), \boldsymbol{a}_\beta(\theta_l^{(1)} + \Delta\theta), \cdots, \boldsymbol{a}_\beta(\theta_l^{(2)})]$; $\bar{\boldsymbol{A}}_{\beta l}$ is the response matrix of virtual subarray β, giving by $\bar{\boldsymbol{A}}_{\beta l} = [\bar{\boldsymbol{a}}_\beta(\theta_l^{(1)}), \bar{\boldsymbol{a}}_\beta(\theta_l^{(1)} + \Delta\theta), \cdots, \bar{\boldsymbol{a}}_\beta(\theta_l^{(2)})]$; $\boldsymbol{a}_\beta(\theta) = [e^{-j(2\pi/\lambda)D_2\sin\theta}, e^{-j(2\pi/\lambda)D_3\sin\theta}, \cdots, e^{-j(2\pi/\lambda)D_M\sin\theta}]^T$ and $\bar{\boldsymbol{a}}_\beta(\theta) = [e^{-j(2\pi/\lambda)d\sin\theta}, e^{-j(2\pi/\lambda)2d\sin\theta}, \cdots, e^{-j(2\pi/\lambda)(M_v-1)d\sin\theta}]^T$.

With the increase of the number of divided sectors, the interpolation error decreases while the computational burden becomes heavier. Typically we choose the minimum in the set of L_s which meet the condition that $(\varepsilon_{\alpha l} + \varepsilon_{\beta l})/2 < 10^{-3}$ as the number of sectors [22]. Then compute $\boldsymbol{B}_{\alpha 1}, \boldsymbol{B}_{\alpha 2}, \cdots \boldsymbol{B}_{\alpha L_s}$ and $\boldsymbol{B}_{\beta 1}, \boldsymbol{B}_{\beta 2}, \cdots \boldsymbol{B}_{\beta L_s}$ in each sector separately. For notational convenience, we use $\boldsymbol{B}_\alpha$ and $\boldsymbol{B}_\beta$ to respectively represent $\boldsymbol{B}_{\alpha 1}, \boldsymbol{B}_{\alpha 2}, \cdots \boldsymbol{B}_{\alpha L_s}$ and $\boldsymbol{B}_{\beta 1}, \boldsymbol{B}_{\beta 2}, \cdots \boldsymbol{B}_{\beta L_s}$ hereafter, then (4) and (6) become:

$$\boldsymbol{B}_\alpha\boldsymbol{A}_\alpha = \bar{\boldsymbol{A}}_\alpha \tag{7}$$

$$\boldsymbol{B}_\beta\boldsymbol{A}_\beta = \bar{\boldsymbol{A}}_\beta \tag{8}$$

It is worth noticing that the interpolation matrix $\boldsymbol{B}_\alpha$ and $\boldsymbol{B}_\beta$ need to be computed only once when the real array is given and the virtual array is designed, and it can be done off-line since we don't use the information of the incident directions of signals.

The received signal $y_i(n)$ of the ith sensor in real array and $\bar{y}_i(n)$ of ith sensor in virtual array can be reexpressed by:

$$y_i(n) = \boldsymbol{b}_i^T(\theta)\boldsymbol{s}(n)+\omega_i(n) \quad (9)$$

$$\bar{y}_i(n) = \bar{\boldsymbol{b}}_i^T(\theta)\boldsymbol{s}(n)+\bar{\omega}_i(n) \quad (10)$$

where $\boldsymbol{b}_i \triangleq [e^{-j(2\pi/\lambda)D_i\sin\theta_1}, e^{-j(2\pi/\lambda)D_i\sin\theta_2}, \cdots, e^{-j(2\pi/\lambda)D_i\sin\theta_p}]^T$; $\bar{\boldsymbol{b}}_i \triangleq [e^{-j(2\pi/\lambda)(i-1)d\sin\theta_1}, e^{-j(2\pi/\lambda)(i-1)d\sin\theta_2}, \cdots, e^{-j(2\pi/\lambda)(i-1)d\sin\theta_p}]^T$; $\omega_i(n)$ and $\bar{\omega}_i(n)$ represent noise in the ith sensor of real array and virtual array respectively. Based on above design of the virtual array, we have:

$$\bar{\boldsymbol{b}}_1^T(\theta) = \boldsymbol{b}_1^T(\theta) \quad (11)$$

$$\bar{\boldsymbol{b}}_{M_v}^T(\theta) = \boldsymbol{b}_M^T(\theta) \quad (12)$$

We use $\boldsymbol{\varphi}_\alpha$ to represent the correlation vector between the signal $y_M(n)$ and the signal vector $\boldsymbol{y}_\alpha(n)$ of real subarray α; and use $\bar{\boldsymbol{\varphi}}_\alpha$ to represent the correlation vector between the signal $\bar{y}_{M_v}(n)$ and the signal vector $\bar{\boldsymbol{y}}_\alpha(n)$ of virtual subarray α :

$$\begin{aligned}\boldsymbol{\varphi}_\alpha &\triangleq E\{y_M(n)\boldsymbol{y}_\alpha^{\mathrm{H}}(n)\} = [r_{M,1}, r_{M,2}, \cdots, r_{M,M-1}]\\ \bar{\boldsymbol{\varphi}}_\alpha &\triangleq E\{\bar{y}_{M_v}(n)\bar{\boldsymbol{y}}_\alpha^{\mathrm{H}}(n)\} = [\bar{r}_{M_v,1}, \bar{r}_{M_v,2}, \cdots, \bar{r}_{M_v,M_v-1}]\end{aligned} \quad (13)$$

where $r_{i,k}$ is the correlation between signals $y_i(n)$ and $y_k(n)$ defined as $r_{i,k} \triangleq E\{y_i(n)y_k^*(n)\}$; $\bar{r}_{i,k}$ is the correlation between signals $\bar{y}_i(n)$ and $\bar{y}_k(n)$ defined as $\bar{r}_{i,k} \triangleq E\{\bar{y}_i(n)\bar{y}_k^*(n)\}$.

Based on (9), $\boldsymbol{\varphi}_\alpha$ can be expanded as:

$$\begin{aligned}\boldsymbol{\varphi}_\alpha &= E\{y_M(n)\boldsymbol{y}_\alpha^{\mathrm{H}}(n)\}\\ &= E\{(\boldsymbol{b}_M^T(\theta)\boldsymbol{s}(n) + \omega_M(n))(\boldsymbol{s}^{\mathrm{H}}(n)\boldsymbol{A}_\alpha^{\mathrm{H}} + \boldsymbol{\omega}_\alpha^{\mathrm{H}}(n))\}\\ &= E\{\boldsymbol{b}_M^T(\theta)\boldsymbol{s}(n)\boldsymbol{s}^{\mathrm{H}}(n)\boldsymbol{A}_\alpha^{\mathrm{H}}\} + E\{\boldsymbol{b}_M^T(\theta)\boldsymbol{s}(n)\boldsymbol{\omega}_\alpha^{\mathrm{H}}(n)\}\\ &\quad + E\{\omega_M(n)\boldsymbol{s}^{\mathrm{H}}(n)\boldsymbol{A}_\alpha^{\mathrm{H}}\} + E\{\omega_M(n)\boldsymbol{\omega}_\alpha^{\mathrm{H}}(n)\}\end{aligned} \quad (14)$$

Because of the assumption 4), the last three items in (14) will be zero, then we get:

$$\boldsymbol{\varphi}_\alpha = E\{\boldsymbol{b}_M^T(\theta)\boldsymbol{s}(n)\boldsymbol{s}^{\mathrm{H}}(n)\boldsymbol{A}_\alpha^{\mathrm{H}}\} = \boldsymbol{b}_M^T(\theta)\boldsymbol{R_S}\boldsymbol{A}_\alpha^{\mathrm{H}} \quad (15)$$

where $\boldsymbol{R_S}$ indicates the covariance matrix of incident signals defined as $\boldsymbol{R_S} \triangleq E\{\boldsymbol{s}(n)\boldsymbol{s}^{\mathrm{H}}(n)\}$. Similarly, based on (10), $\bar{\boldsymbol{\varphi}}_\alpha$ can be expanded as:

$$\bar{\boldsymbol{\varphi}}_\alpha = E\{\bar{\boldsymbol{b}}_{M_v}^T(\theta)\boldsymbol{s}(n)\boldsymbol{s}^{\mathrm{H}}(n)\bar{\boldsymbol{A}}_\alpha^{\mathrm{H}}\} = \bar{\boldsymbol{b}}_{M_v}^T(\theta)\boldsymbol{R_S}\bar{\boldsymbol{A}}_\alpha^{\mathrm{H}} \quad (16)$$

Based on (7),(12) and (15), $\bar{\boldsymbol{\varphi}}_\alpha$ in (16) can be converted to:

$$\begin{aligned}\bar{\boldsymbol{\varphi}}_\alpha &= \bar{\boldsymbol{b}}_{M_v}^T(\theta)\boldsymbol{R_S}\bar{\boldsymbol{A}}_\alpha^{\mathrm{H}}\\ &= \boldsymbol{b}_M^T(\theta)\boldsymbol{R_S}\bar{\boldsymbol{A}}_\alpha^{\mathrm{H}}\\ &= \boldsymbol{b}_M^T(\theta)\boldsymbol{R_S}\boldsymbol{A}_\alpha^{\mathrm{H}}\boldsymbol{B}_\alpha^{\mathrm{H}}\\ &= \boldsymbol{\varphi}_\alpha\boldsymbol{B}_\alpha^{\mathrm{H}}\end{aligned} \quad (17)$$

In the same way, we define $\boldsymbol{\varphi}_\beta$ and $\bar{\boldsymbol{\varphi}}_\beta$ as:

$$\boldsymbol{\varphi}_\beta \triangleq E\{y_1(n)\boldsymbol{y}_\beta^{\mathrm{H}}(n)\} = \boldsymbol{b}_1^T(\beta)\boldsymbol{R_S}\boldsymbol{A}_\beta^{\mathrm{H}} \quad (18)$$

$$\bar{\boldsymbol{\varphi}}_\beta \triangleq E\{\bar{y}_1(n)\bar{\boldsymbol{y}}_\beta^{\mathrm{H}}(n)\} = \bar{\boldsymbol{b}}_1^T(\beta)\boldsymbol{R_S}\bar{\boldsymbol{A}}_\beta^{\mathrm{H}} \quad (19)$$

Then based on (8),(11), we get the transform relationship between $\boldsymbol{\varphi}_\beta$ and $\bar{\boldsymbol{\varphi}}_\beta$:

$$\bar{\boldsymbol{\varphi}}_\beta = \boldsymbol{\varphi}_\beta\boldsymbol{B}_\beta^{\mathrm{H}} \quad (20)$$

Evidently by using received data from the real arbitrary linear array and the interpolation matrices computed off-line, we obtain the correlation vectors $\bar{\boldsymbol{\varphi}}_\alpha$ and $\bar{\boldsymbol{\varphi}}_\beta$ of virtual array when same incident signals are imping on the virtual array. Hence the given arbitrary array is converted into virtual ULA successfully.

B. DOA Estimation Without Eigendecompositon

Using the correlation vectors $\bar{\boldsymbol{\varphi}}_\alpha = [\bar{r}_{M_v,1}, \bar{r}_{M_v,2}, \cdots, \bar{r}_{M_v,M_v-1}]$ and $\bar{\boldsymbol{\varphi}}_\beta = [\bar{r}_{1,2}, \bar{r}_{1,3}, \cdots, \bar{r}_{1,M_v}]$ of virtual array, based on SUMWE [19], we can estimate DOA of coherent signals as follows. Firstly, form the correlation matrices $\boldsymbol{\Phi}_f$, $\bar{\boldsymbol{\Phi}}_f$, $\boldsymbol{\Phi}_b$ and $\bar{\boldsymbol{\Phi}}_b$ by using correlation coefficients in $\bar{\boldsymbol{\varphi}}_\alpha$ and $\bar{\boldsymbol{\varphi}}_\beta$:

$$\begin{aligned}\boldsymbol{\Phi}_f &\triangleq [\varphi_{f,1}, \varphi_{f,2}, \cdots, \varphi_{f,L-1}]^T\\ \bar{\boldsymbol{\Phi}}_f &\triangleq [\bar{\varphi}_{f,2}, \bar{\varphi}_{f,3}, \cdots, \bar{\varphi}_{f,L}]^T\\ \boldsymbol{\Phi}_b &\triangleq [\varphi_{b,1}, \varphi_{b,2}, \cdots, \varphi_{b,L-1}]^T\\ \bar{\boldsymbol{\Phi}}_b &\triangleq [\bar{\varphi}_{b,2}, \bar{\varphi}_{b,3}, \cdots, \bar{\varphi}_{b,L}]^T\end{aligned} \quad (21)$$

where $\boldsymbol{\varphi}_{f,l} \triangleq [\bar{r}_{l,M_v}, \bar{r}_{l+1,M_v}, \cdots, \bar{r}_{l+p-1,M_v}]^T$, $\bar{\boldsymbol{\varphi}}_{f,l} \triangleq [\bar{r}_{l,1}, \bar{r}_{l+1,1}, \cdots, \bar{r}_{l+p-1,1}]^T$, $\boldsymbol{\varphi}_{b,l} \triangleq [\bar{r}_{1,l}, \bar{r}_{1,l+1}, \cdots, \bar{r}_{1,l+p-1}]^T$, $\bar{\boldsymbol{\varphi}}_{b,l} \triangleq [\bar{r}_{M_v,1}, \bar{r}_{M_v,l+1}, \cdots, \bar{r}_{M_v,l+p-1}]^T$ for $l = 1, 2, \cdots, L-1$, and $L = M_v - p + 1$. Then, divide correlation matrices $\boldsymbol{\Phi}_f$, $\bar{\boldsymbol{\Phi}}_f$, $\boldsymbol{\Phi}_b$ and $\bar{\boldsymbol{\Phi}}_b$ in (21) as:

$$\begin{aligned}\boldsymbol{\Phi}_f &\triangleq \begin{bmatrix}\boldsymbol{\Phi}_{f1}\\ \boldsymbol{\Phi}_{f2}\end{bmatrix} \begin{matrix}\} & p\\ \} & L-p-1\end{matrix}\\ \bar{\boldsymbol{\Phi}}_f &\triangleq \begin{bmatrix}\bar{\boldsymbol{\Phi}}_{f1}\\ \bar{\boldsymbol{\Phi}}_{f2}\end{bmatrix} \begin{matrix}\} & p\\ \} & L-p-1\end{matrix}\\ \boldsymbol{\Phi}_b &\triangleq \begin{bmatrix}\boldsymbol{\Phi}_{b1}\\ \boldsymbol{\Phi}_{b2}\end{bmatrix} \begin{matrix}\} & p\\ \} & L-p-1\end{matrix}\\ \bar{\boldsymbol{\Phi}}_b &\triangleq \begin{bmatrix}\bar{\boldsymbol{\Phi}}_{b1}\\ \bar{\boldsymbol{\Phi}}_{b2}\end{bmatrix} \begin{matrix}\} & p\\ \} & L-p-1\end{matrix}\end{aligned} \quad (22)$$

Then, we can form a linear operator $\boldsymbol{T}$ [19]:

$$\boldsymbol{T} = (\boldsymbol{\Phi}_1\boldsymbol{\Phi}_1^{\mathrm{H}})^{-1}\boldsymbol{\Phi}_1\boldsymbol{\Phi}_2^{\mathrm{H}} \quad (23)$$

where $\boldsymbol{\Phi}_1 \triangleq [\boldsymbol{\Phi}_{f1}, \bar{\boldsymbol{\Phi}}_{f1}, \boldsymbol{\Phi}_{b1}, \bar{\boldsymbol{\Phi}}_{b1}]$, and $\boldsymbol{\Phi}_2 \triangleq [\boldsymbol{\Phi}_{f2}, \bar{\boldsymbol{\Phi}}_{f2}, \boldsymbol{\Phi}_{b2}, \bar{\boldsymbol{\Phi}}_{b2}]$. The estimated orthogonal projector $\boldsymbol{\Pi}_Q$ can be calculated by:

$$\boldsymbol{\Pi}_Q = \boldsymbol{Q}(\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{Q})^{-1}\boldsymbol{Q}^{\mathrm{H}} \quad (24)$$

where $\boldsymbol{Q} \triangleq [\boldsymbol{T}^T, -\boldsymbol{I}_{L-p-1}]^T$.

Finally, the DOA $\{\theta_i\}_{i=1}^p$ can be estimated by searching the p highest peaks of the spatial spectrum $P(\theta)$:

$$P(\theta) \triangleq 1/\bar{\boldsymbol{a}}^{\mathrm{H}}(\theta)\boldsymbol{\Pi}_Q\bar{\boldsymbol{a}}(\theta) \quad (25)$$

where $\bar{\boldsymbol{a}}(\theta) \triangleq [1, e^{-j(2\pi/\lambda)d\sin\theta}, \cdots, e^{-j(2\pi/\lambda)(L-2)d\sin\theta}]$. Benefiting from the Vandermonde structure of the response matrix $\bar{\boldsymbol{A}}$ of the virtual array, we can use polynomial rooting to replace the peaks searching here. By defining a complex variable z as $z = e^{-j(2\pi/\lambda)d\sin(\theta)}$, from (25), we have

$$P(\theta) = P(z) = 1/\bar{\boldsymbol{a}}^{\mathrm{H}}(z)\boldsymbol{\Pi}_Q\bar{\boldsymbol{a}}(z) \quad (26)$$

where $\bar{\boldsymbol{a}}(z) = [1, z, \cdots, z^{L-2}]^T$. DOA estimates can be obtained by finding the phases of the p zeros of the polynomial

$P(z)$ in (26) closest to the unit circle in the z-plane [3], [24].

C. Implementation of ISUMWE

Based on the above discussion, when the finite array data $\{\boldsymbol{y}(n)\}_{n=1}^{N}$ are available, we can summarize the implementation of the proposed ISUMWE method as follows:

1) Calculate the estimated correlation vector $\boldsymbol{\varphi}_\alpha$ of $y_M(n)$ and $\boldsymbol{y}_\alpha(n)$ and that $\boldsymbol{\varphi}_\beta$ of $y_1(n)$ and $\boldsymbol{y}_\beta(n)$ as

$$\begin{aligned}\boldsymbol{\varphi}_\alpha &= E\{y_M(n)\boldsymbol{y}_\alpha^{\mathrm{H}}(n)\}\\ \boldsymbol{\varphi}_\beta &= E\{y_1(n)\boldsymbol{y}_\beta^{\mathrm{H}}(n)\}\end{aligned} \tag{27}$$

2) Estimate the interpolation matrix $\boldsymbol{B}_\alpha$ and $\boldsymbol{B}_\beta$ by calculating the least squares solution of (7) and (8) respectively.
3) Calculate the correlation vector $\bar{\boldsymbol{\varphi}}_\alpha$ and $\bar{\boldsymbol{\varphi}}_\beta$ in virtual ULA with (17),(20) by using $\boldsymbol{\varphi}_\alpha$, $\boldsymbol{B}_\alpha$, $\boldsymbol{\varphi}_\beta$ and $\boldsymbol{B}_\beta$.
4) Form the correlation matrices $\boldsymbol{\Phi}_f$, $\bar{\boldsymbol{\Phi}}_f$, $\boldsymbol{\Phi}_b$ and $\bar{\boldsymbol{\Phi}}_b$ by using correlation coefficients in $\bar{\boldsymbol{\varphi}}_\alpha$ and $\bar{\boldsymbol{\varphi}}_\beta$ with (21).
5) Estimate the linear operator $\boldsymbol{T}$ with $\boldsymbol{T} = (\boldsymbol{\Phi}_1\boldsymbol{\Phi}_1^{\mathrm{H}})^{-1}\boldsymbol{\Phi}_1\boldsymbol{\Phi}_2^{\mathrm{H}}$ by dividing $\boldsymbol{\Phi}_f$, $\bar{\boldsymbol{\Phi}}_f$, $\boldsymbol{\Phi}_b$ and $\bar{\boldsymbol{\Phi}}_b$ with (22) and calculate the estimated orthogonal projector $\boldsymbol{\Pi}_Q$ as

$$\boldsymbol{\Pi}_Q = \boldsymbol{Q}(\boldsymbol{I}_{L-p-1} - \boldsymbol{T}^{\mathrm{H}}(\boldsymbol{T}\boldsymbol{T}^{\mathrm{H}} + \boldsymbol{I}_p)^{-1}\boldsymbol{T})\boldsymbol{Q}^{\mathrm{H}} \tag{28}$$

6) Estimate the directions $\{\theta_i\}_{i=1}^{p}$ by finding the phases of the p zeros of the polynomial $P(z)$ in (26) closest to the unit circle in the z-plane.

The implementation of ISUMWE algorithm requires three major steps: (i) computation of the correlation vectors $\boldsymbol{\varphi}_\alpha$ and $\boldsymbol{\varphi}_\beta$ in (27); (ii) calculate the virtual correlation vectors $\bar{\boldsymbol{\varphi}}_\alpha$ and $\bar{\boldsymbol{\varphi}}_\beta$ using interpolation matrix $\boldsymbol{B}_\alpha$, $\boldsymbol{B}_\beta$ and real correlation vectors $\boldsymbol{\varphi}_\alpha$, $\boldsymbol{\varphi}_\beta$ with (17),(20), and (iii) estimation of the orthogonal projector $\boldsymbol{\Pi}_Q$ with (23),(28). Here we use the number of MATLAB flops as a measure of the algorithm complexity, where a flop is defined as a floating-point addition or multiplication operation. Then we approximately get the computation in step (i) is about $16NM$ flops, that in (ii) is about $16MM_v$ flops, and for steps(iii), the computation of $\boldsymbol{T}$ takes about $64p^2(M_v - 2p) + F_{inv}(p) + 32p^3$ flops, and that of $\boldsymbol{\Pi}_Q$ is about $16p^2(M_v - 2p) + 8M_v(M_v - 2p)^2 + 8(M_v - p)^2(M_v - 2p) + F_{inv}(p)$ flops, where $F_{inv}(p)$ indicates the complexity of inversion operation of $p \times p$ matrix. When $N >> M, M_v >> p$, the complexity of ISUMWE is nearly $16NM + 16MM_v + 16M(M - p)$ flops, which occurs often in applications of DOA estimation, where the computation of interpolation matrices is computed off-line and the computations needed in the remaining steps are negligible.

Some existing interpolated DOA estimation methods of coherent signals such as the FBSS-based root-MUSIC with interpolation (I-FBSS-root-MUSIC) [23], the FBSS-based standard Capon beamformer with interpolation (I-FBSS-root-SCB) [8], [21] and the FBSS-based Modified Capon beamformer with interpolation (I-FBSS-root-MCB) [21] were proposed. The I-FBSS-root-MUSIC involves one eigendecomposition operation, and its computational complexity is about $8NM^2 + F_{evd}(M_0) + 16M_v^2M + 8M_vM^2$ flops, where M_0 is the number of sensors in subarray of spacing smoothing and $F_{evd}(M_0)$ indicates the complexity of eigendecomposition of $M_0 \times M_0$ matrix. On the contrary, the complexity of I-FBSS-root-MCB is $8NM^2 + F_{inv}(M_0) + 16M_v^2M + 8M_vM^2 + 8(m-1){M_0}^3$ flops, which equals to the complexity of I-FBSS-root-SCB when the parameter $m = 1$. Beamformer based methods(I-FBSS-root-SCB, I-FBSS-root-MCB) estimate DOA without eigendecomposition, hence have lower complexity than I-FBSS-root-MUSIC. While the proposed ISUMWE avoids eigendecomposition and doesn't need calculate all correlations in array covariance matrix, so it has much lower complexity.

IV. Numerical results

In this section, we evaluate effectiveness of the proposed method for DOA estimation of coherent narrowband signals by using an arbitrary linear array, which is constructed by adding a horizontal shift Δd_i to the ith $(i = 1, 2, \cdots M)$ element of a ULA consisting of $M = 10$ sensors with element spacing $d = \lambda/2$. As it is shown in Fig. 3, the horizontal shift vector is given by

$$\begin{aligned}\boldsymbol{\Delta_d} \;=\;& [0, 0, 0.1\lambda, -0.1\lambda, 0.1\lambda,\\ & -0.1\lambda, 0.1\lambda, -0.1\lambda, 0.1\lambda, -0.1\lambda]^T\end{aligned} \tag{29}$$

Two coherent signals with equal power arrive from angles θ_1=8^o and θ_2= 20^o.

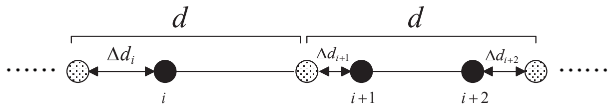

Fig. 3. Simulation array setup

Meanwhile, the behavior of I-FBSS-root-SCB [8], [21], I-FBSS-root-MCB [21], I-FBSS-root-MUSIC [23], and the Cramer-Rao lower bound (CRB) [25], [26] are also carried out for comparison. The results shown below are all based on 1000 independent trials. The empirical root-mean-square error (RMSE) in the simulations is defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{Kp}\sum_{k=1}^{K}\sum_{i=1}^{p}\left(\theta_i - \hat{\theta}_i^{(k)}\right)^2} \tag{30}$$

where the K denotes the total number of trials and $\hat{\theta}_i^{(k)}$ is the estimate of θ_i obtained in the kth trial.

Example 1–Performance versus SNR: The number of snapshots used is $N = 600$. The RMSEs of estimates of the DOA versus SNRs are shown in Fig. 4. From the curves in Fig. 4, we can see that all methods estimate the DOA of the coherent signals relatively accurately and the RMSEs decrease significantly with the increase of SNR, benefiting from the utilizing of interpolation transform and array decorrelation. Meanwhile, the proposed method completely outperforms the I-FBSS-root-SCB method, and outperforms the I-FBSS-root-MCB method at low SNR. With the increase of SNR, the resolution of proposed method becomes close to that of I-FBSS-root-MUSIC method which is known for its precision.

Example 2–Performance versus Number of Snapshots: The SNR is fixed at 5dB, and the number of snapshots varies from 100 to 10000. The results are shown in Fig. 5. It can be observed that the proposed method outperforms the I-FBSS-root-MCB method at high snapshots, and its RMSEs decrease

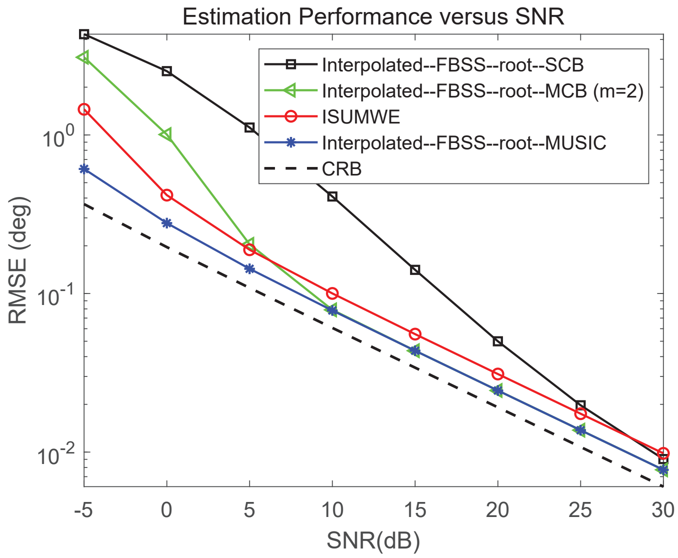

Fig. 4. DOA estimation performance versus SNR.

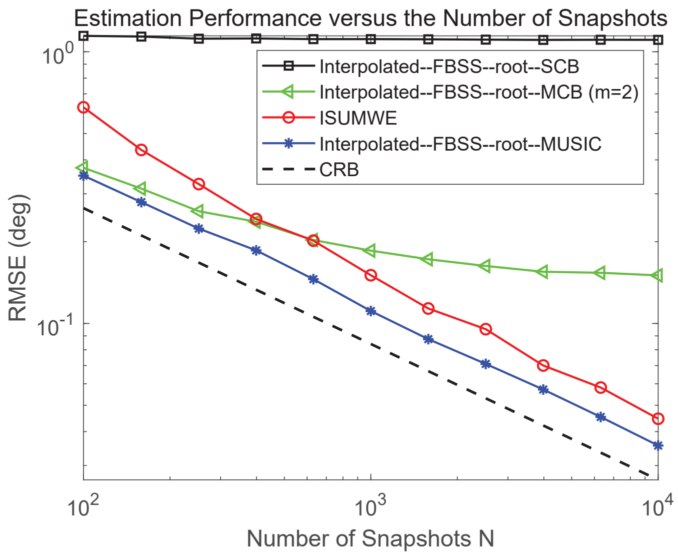

Fig. 5. DOA estimation performance versus the number of snapshots.

monotonically with the increase of the number of snapshots and becomes close to that of I-FBSS-root-MUSIC method with the increase of snapshots, while the RMSEs of the I-FBSS-root-SCB method remain large for all numbers of snapshots.

V. Conclusion

In this paper, we propose an ISUMWE for DOA estimation of coherent signals impinging on an arbitrary linear array. By using an interpolation transform technique, the restriction of ULA geometry required by subarray averaging and polynomial rooting is overcome. As shown in simulations, the proposed method performs well in arbitrary linear array. Moreover, our method avoids the computationally intensive operations of eigendecomposition. Therefore, it is still valid when number of sensors is large or the real-time estimation is required.

References

[1] S. Haykin, Ed., Array Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[2] H. Krim and M. Viberg, "Two decades of array signal processing research: The parametric approach," *IEEE Signal Process. Mag.*, vol. 13, no. 4, pp. 67-94, 1996.

[3] T. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Trans. Acous., Speech, Signal Process.*, vol. 33, no. 4, pp. 806-811, 1985.

[4] J. Yin and T. Chen, "Direction-of-Arrival estimation using a sparse representation of array covariance vectors," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4489-4493, 2011.

[5] Y. Zhang, M. Amin, B. Himed, et al "Sparsity-based DOA estimation using co-prime arrays," *IEEE Int. Conf. Acous., Speech, Signal Process.*, pp. 3967-3971, 2013.

[6] S. Junpeng, H. Guoping, S. Fenggang, et al "Improved spatial differencing scheme for 2-D DOA estimation of coherent signals with uniform rectangular arrays," *Sensors*, vol. 17, no. 9, pp. 1956- , 2017.

[7] Y. Hu, Y. Liu, X. Wang, "DOA estimation of coherent signals using fourth-order cumulants on coprime arrays," *Sensors*, vol. 17, no. 4, 2016.

[8] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, no. 8, pp. 1408-1418, 1969.

[9] J. Capon and N. R. Goodman, "Probability distributions for estimators of the frequency-wavenumber spectrum," *Proc. IEEE*, vol. 58, no. 10, pp. 1785-1786, 1970.

[10] U. Nickel, "Algebraic formulation of Kumaresan-Tufts superresolution method, showing relation to ME and MUSIC methods," *IEE Proc.*, vol. 135, Pt. F, no. 1, pp. 7-10, 1988.

[11] C. Liu, J. Xin, Y. Nishio, N. Zheng, and A. Sano, "Modified Capon beamformer for high-resolution direction-of-arrival estimation," *Proc. 2014 Int. Symp. Nonlinear Theory and Its App.*, pp. 116-119, 2014.

[12] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276-280, 1986.

[13] C.-C. Yeh, "Simple computation of projection matrix for bearing estimations," *Proc. Inst. Elect. Eng. F*, vol. 134, no. 2, pp. 146-150, 1987.

[14] C.-C. Yeh, "Projection approach to bearing estimation," *IEEE Trans. Acoust.,Speech, Signal Processing*, vol. ASSP-34, pp. 1347-1349, May 1986.

[15] P. Stoica and T. Söderström, "Statistical analysis of a subspace method for bearing estimation without eigendecomposition," *Proc. Inst. Elect. Eng. F*, vol. 139, no. 4, pp. 301-305, 1992.

[16] S. Marcos and M. Benidir, "On a high resolution array processing method nonbased on the eigenanalysis approach," *in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 2955-2958, Albuquerque, NM, Apr. 1990.

[17] S. Marcos, A. Marsal, and M. Benidir, "The propagator method for source bearing estimation," *Signal Process.*, vol. 42, no. 2, pp. 121-138, 1995.

[18] A. Eriksson, P. Stoica, and T. Söderström, "On-line subspace algorithms for tracking moving sources," *IEEE Trans. Signal Process.*, vol. 42, pp. 2319-2330, Sept. 1994.

[19] J. Xin and A. Sano, "Computationally efficient subspace-based method for direction-of-arrival estimation without eigendecomposition," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 876-893, 2004

[20] A. Raj and J. Mcclellan, "Single snapshot super-resolution DOA estimation for arbitrary array geometries," *IEEE Signal Processing Letters.*, vol. 1, no. 1, pp. 99, 2018

[21] X. Chen, J. Xin, N. Zheng, and A. Sano, "Modified Capon beamformer for high-resolution direction-of-arrival estimation," *IEEE International Workshop on Signal Process. Systems*, pp. 1-5, 2017.

[22] B. Friedlander, "Direction finding using an interpolated array," *Proc. 1990 IEEE Int. Conf. Acous., Speech, Signal Process.*, vol. 5, pp. 2951-2954, 1990.

[23] B. Friedlander and A.J. Weiss, "Direction finding using spatial smoothing with interpolated arrays," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 2, pp. 574-587, 1992.

[24] J. Wu, T. Wang, Z. Suo, and Z. Bao, "DOA estimation for ULA by spectral Capon rooting method," *Electron. Lett.*, vol. 45, no. 1, pp. 84-85, 2009.

[25] J. Sheinvald and M. Wax and A. J. Weiss, "On maximum-likelihood localization of coherent signals," *IEEE Trans. Signal Process.*, vol. 44, no. 10, pp. 2475-2482, 1996.

[26] P. Stoica and E. G. Larsson and A. B. Gershman, "The stochastic CRB for array processing: a textbook derivation," *IEEE Signal Process. Lett.*, vol. 8, no. 5, pp. 148-150, 2001.