Fast implementation of sparse iterative covariance-based estimation for source localization*

Qilin Zhang, Habti Abeida, Ming Xue, William Rowe, and Jian Li
Department of Electrical and Computer Engineering, University of Florida, Gainesville, Florida 32611

(Received 12 May 2011; revised 14 November 2011; accepted 18 November 2011)

Fast implementations of the sparse iterative covariance-based estimation (SPICE) algorithm are presented for source localization with a uniform linear array (ULA). SPICE is a robust, user parameter-free, high-resolution, iterative, and globally convergent estimation algorithm for array processing. SPICE offers superior resolution and lower sidelobe levels for source localization compared to the conventional delay-and-sum beamforming method; however, a traditional SPICE implementation has a higher computational complexity (which is exacerbated in higher dimensional data). It is shown that the computational complexity of the SPICE algorithm can be mitigated by exploiting the Toeplitz structure of the array output covariance matrix using Gohberg–Semencul factorization. The SPICE algorithm is also extended to the acoustic vector-sensor ULA scenario with a specific nonuniform white noise assumption, and the fast implementation is developed based on the block Toeplitz properties of the array output covariance matrix. Finally, numerical simulations illustrate the computational gains of the proposed methods.
 [DOI: 10.1121/1.3672656]

PACS number(s): 43.60.Jn, 43.60.Fg, 43.60.Gk [EJS] Pages: 1249–1259

I. INTRODUCTION

The direction-of-arrival (DOA) source localization problem is directly applicable to many fields including radar, sonar, communications, and the geological and biomedical sciences. The goal of the DOA problem in passive sonar applications is to accurately locate all sources with an array of acoustic sensors. Various spectral estimation methods have been applied to solving this problem and may be classified as nonparametric, parametric, and semiparametric (see, e.g., Ref. 1).

Nonparametric methods include the well-known delay-and-sum (DAS) beamformer and Capon beamformer. The DAS beamformer is a computationally efficient and robust method, but suffers in performance due to high sidelobes, low resolution, and poor accuracy. The data adaptive Capon beamformer improves on the DAS method, but it requires noncoherent sources.[2] Parametric methods, such as multiple signal classification (MUSIC), can provide higher resolution than the nonparametric methods. Unfortunately, these methods are typically very sensitive to modeling errors and can even fail if coherent sources are present.[2] The sources in the underwater acoustic environment can be coherent due to multipath reflections from the ocean floor or surface.

Recently, a new semiparametric algorithm, referred to as the sparse iterative covariance-based estimation (SPICE), has been proposed.[3,4] SPICE offers superior resolution and low sidelobe levels, simultaneously retaining robustness against correlated sources.[4] It is also a user parameter-free method that is guaranteed to converge globally. However, SPICE suffers from higher computational complexity when compared with the DAS method.

In this paper, an efficient implementation of SPICE based on Gohberg–Semencul (GS) factorization is presented, which is inspired by previous fast implementations on Capon and amplitude and phase estimation (APES).[5–9] The inverse of the sample covariance matrix $\hat{\mathbf{R}}_M$ is decomposed by Cholesky or GS factorization[5–9] and fast schemes are applied to compute the spectral estimates. Efficient decomposition of $\hat{\mathbf{R}}_M^{-1}$ is achieved using the Toeplitz structure of the approximated $\hat{\mathbf{R}}_M^{-1}$ or the data matrices.[5,7–9] After factorization of $\hat{\mathbf{R}}_M$, a fast Fourier transform (FFT) is applied to further accelerate the computation of the spectral estimates.[6,7] In this paper, the Toeplitz matrix structure produced by the GS factorization of the inverse of the SPICE covariance matrix $\mathbf{R}_M$ is exploited, simultaneously applying a Levinson–Durbin type algorithm (see e.g., Ref. 10). If the received signals are also collected from a uniform linear array (ULA), the computational efficiency can be further increased by use of the FFT and inverse fast Fourier transform (IFFT) to perform the matrix-vector products.

The fast implementation of the SPICE algorithm on ULA of vector sensors is also presented in this paper. Acoustic vector-sensor arrays have attracted much attention in recent years. The U.S. Navy has been using vector-sensor arrays in the form of "DIFAR" (directional low frequency analysis and recording sonobuoys) to study the acoustic properties of the ocean[11] and researchers have been using them to study marine life.[12] A typical vector-sensor measurement consists of a set of measurements of acoustic particle velocities (first order derivatives of the acoustic pressure) as well as the measurement of the omnidirectional acoustic pressure. With the direct measurements of directional velocities, vector-sensor arrays are superior to conventional hydrophone arrays of the same aperture in many performance criteria: free of the left/right bearing ambiguity even for 1D array, capable of simultaneously estimating both azimuth

*Implementation codes available online:
https://qilin-zhang.github.io/publications/

and elevation angles, etc.[13] The cost for these advantages is that the measurement of a vector-sensor array will have a higher dimensionality (higher computational complexity) than that of a standard hydrophone array. To mitigate the computational complexity, the fast implementation of SPICE is also extended to the vector-sensor array scenario by exploiting the block Toeplitz structure of the array output covariance matrix (e.g., Ref. 14) using a GS type factorization based on a generalized Levinson–Durbin algorithm (LDA) and FFT/IFFT.

Notation: We denote vectors and matrices by boldface lowercase and boldface uppercase letters, respectively. Table I specifies symbols and their meanings in the paper.

The rest of the paper is organized as follows. The DOA source localization problem is formulated in Sec. II. In Section III, the SPICE algorithm for array processing is briefly reviewed. In Sec. IV, the fast implementation of SPICE for a hydrophone ULA is presented. In Sec. V, properties of vector-sensor arrays are reviewed and the fast implementation of SPICE for vector-sensor ULAs is presented. Section VI provides simulated numerical results that show the computational gains of our method. Section VII contains a concise summary of the results.

II. DATA MODEL AND PROBLEM FORMULATION

Let a ULA of M omnidirectional sensors with half-wavelength interelement spacing receive narrowband signals impinging from the sources with unknown locations. Let Ω denote the set of possible impinging angles, and θ be the angle-of-arrival. Also, let $\{\theta_k\}_{k=0}^{K-1}$ denote a grid that covers Ω. The $M \times 1$ complex snapshot vectors can be modeled as (e.g., Ref. 3)

$$\mathbf{y}_M(n) = \mathbf{A}_M \mathbf{x}(n) + \mathbf{e}(n), \qquad n = 1, ..., N, \tag{1}$$

where $\mathbf{A}_M \triangleq [\mathbf{a}_M(\theta_0), ..., a_M(\theta_{K-1})]$ is the steering matrix with each steering vector $a_M(\theta_k) \triangleq \left[1, e^{j\pi \sin(\theta_0)}, ..., e^{j\pi(M-1)\sin(\theta_k)}\right]^T$ being parameterized by the scalar location parameter θ_k. The vector $\mathbf{x}(n) \triangleq [\mathbf{x}_0(n), ..., \mathbf{x}_{K-1}(n)]^T$ contains the K unknown complex-valued signals, and $\mathbf{e}(n)$ is the noise term. We assume $E[\mathbf{e}(n)\mathbf{e}^H(\bar{n})] = \sigma \mathbf{I}_M \delta_{n,\bar{n}}$, where $\delta_{n,\bar{n}} = 1$ if $n = \bar{n}$ and 0 otherwise. Let us further assume that $\mathbf{e}(n)$ and $\mathbf{x}(n)$ are statistically independent. Thus, $E[\mathbf{x}(n)\mathbf{x}^H(\bar{n})] = \mathbf{P}_K \delta_{n,\bar{n}}$, where $\mathbf{P}_K \triangleq \text{Diag}(p_0, \ldots, p_{k-1})$, with p_k denoting the unknown signal power at θ_k. This leads to the covariance matrix of $\mathbf{y}_M(n)$ (e.g., Refs. 3 and 4).

TABLE I. Notation used in the text.

$\triangleq$	a definition
$\|\cdot\|_F$	Frobenius norm
$\odot$	the Hadamard (elementwise) matrix product
$\otimes$	the Kronecker matrix product
$(\cdot)^T$	transpose of a vector or matrix
$(\cdot)^*$	complex conjugate
$(\cdot)^H$	conjugate transpose of a vector or matrix
Diag$(\cdot)$	diagonal matrix with a center dot $(\cdot)$as its diagonal elements
$\mathbf{I}_p$	Identity matrix of dimension $p \times p$
$\mathbf{J}_p$	exchange matrix of dimension $p \times p$, with ones on the antidiagonal

$$\mathbf{R}_M \triangleq \mathbf{A}_M \mathbf{P}_K \mathbf{A}_M^H + \sigma \mathbf{I}_M. \tag{2}$$

This covariance matrix is traditionally estimated by the sample covariance matrix $\hat{\mathbf{R}}_M \triangleq \frac{1}{N}\mathbf{Y}\mathbf{Y}^H$, where $\mathbf{Y} \triangleq [\mathbf{y}_M(1), ..., \mathbf{y}_M(N)]$.

III. THE SPICE ALGORITHM

SPICE is a recently introduced method for sparse signal recovery in linear models derived from a robust covariance fitting criterion. It does not depend on any hyperparameters and achieves better performance than the well-known methods such as MUSIC.[3,4] In the particular case of spatially and temporally white uniform noise (i.e., $E[\mathbf{e}(n)\mathbf{e}^H(n)] = \sigma \mathbf{I}_M$), the iterative steps of the SPICE method (SPICE + as named in Ref. 4) are summarized as follows:

Initialize $\{p_k^{(0)}\}_{k=0}^{K-1}$ using DAS and $\sigma^{(0)}$ with a small value (e.g., the average of the smallest M values in $\{p_k^{(0)}\}_{k=0}^{K-1}$). At the ith iteration the following should take place:

(1) Update $\mathbf{R}_M^{(i)}$ using the signal power estimates $\{p_k^{(i-1)}\}_{k=0}^{K-1}$ and the noise power estimate $\sigma^{(i-1)}$ from the $(i-1)$th iteration,

$$\mathbf{R}_M^{(i)} = \mathbf{A}_M \mathbf{P}_K^{(i-1)} \mathbf{A}_M^H + \sigma^{(i-1)} \mathbf{I}_M, \tag{3}$$

where $\mathbf{P}_K^{(i-1)} \triangleq \text{Diag}\left\{\mathbf{p}_K^{(i-1)}\right\}$ and $\mathbf{p}_K^{(i-1)} \triangleq \left[p_0^{(i-1)}, ..., p_{K-1}^{(i-1)}\right]^T$.

(2) Using the most recently obtained $\mathbf{R}_M^{(i)}$ in Eq. (3), noise power estimate $\sigma^{(i-1)}$ and signal power estimates $\{p_k^{(i-1)}\}_{k=0}^{K-1}$, compute the auxiliary variable $\rho^{(i)}$,

$$\rho^{(i)} = \sum_{k=0}^{K-1} \omega_k^{1/2} p_k^{(i-1)} \left\| \mathbf{a}_M^H(\theta_k) \mathbf{R}_M^{(i)^{-1}} \hat{\mathbf{R}}_M^{1/2} \right\|_F + \gamma^{1/2} \sigma^{(i-1)} \left\| \mathbf{R}_M^{(i)^{-1}} \hat{\mathbf{R}}_M^{1/2} \right\|_F. \tag{4}$$

(3) Estimate the noise power,

$$\sigma^{(i)} = \sigma^{(i-1)} \frac{\left\| \mathbf{R}_M^{(i)^{-1}} \hat{\mathbf{R}}_M^{1/2} \right\|_F}{\gamma^{1/2} \rho^{(i)}}. \tag{5}$$

(4) Update $\{p_k^{(i)}\}_{k=0}^{K-1}$ using $\rho^{(i)}$ and $\sigma^{(i)}$,

$$p_k^{(i)} = p_k^{(i-1)} \frac{\left\| \mathbf{a}_M^H(\theta_k) \mathbf{R}_M^{(i)^{-1}} \hat{\mathbf{R}}_M^{1/2} \right\|_F}{\omega_k^{1/2} \rho^{(i)}}, \quad k = 0, 1, ..., K-1, \tag{6}$$

until practical convergence,[15] where $\{\omega_k\}_{k=0}^{K+M-1}$ are constants over all iterations. They are given by

$$\omega_k \triangleq \mathbf{a}_M^H(\theta_k) \hat{\mathbf{R}}_M^{-1} \mathbf{a}_M(\theta_k)/N, \qquad k = 0, ..., K-1,$$
$$\omega_{K+k-1} \triangleq \hat{\mathbf{R}}_M^{-1}(k,k)/N, \qquad k = 1, ..., M,$$

where $\hat{\mathbf{R}}_M^{-1}(k,k)$ denotes the element at the kth row and the kth column of the matrix $\hat{\mathbf{R}}_M^{-1}$. γ is given by

$$\gamma \triangleq \sum_{k=K}^{K+M-1} \omega_k.$$

Note that during each iteration, we update $\mathbf{R}_M^{(i)}$ based on the latest estimates of signal powers $\{\mathbf{p}_K^{(i-1)}\}_{k=0}^{K-1}$ and noise term $\sigma^{(i-1)}$. The terms $\left\|\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\right\|_F$ and $\left\|\mathbf{a}_M^H(\theta_k)\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\right\|_F$ listed in the above-presented iterative steps [Eqs. (4)–(6)] have simpler forms,

$$\begin{aligned}\left\|\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\right\|_F^2 &= \frac{1}{N}\sum_{n=1}^{N}\mathrm{tr}\left[\mathbf{R}_M^{(i)^{-1}}\mathbf{y}_M(n)\mathbf{y}_M^H(n)\mathbf{R}_M^{(i)^{-1}}\right]\\ &= \frac{1}{N}\sum_{n=1}^{N}\left[\mathbf{y}_M^H(n)\mathbf{R}_M^{(i)^{-1}}\right]\left[\mathbf{R}_M^{(i)^{-1}}\mathbf{y}_M(n)\right]\\ &= \frac{1}{N}\sum_{n=1}^{N}\left|\mathbf{R}_M^{(i)^{-1}}\mathbf{y}_M(n)\right|^2 \end{aligned} \tag{7}$$

and similarly,

$$\begin{aligned}\left\|\mathbf{a}_M^H(\theta_k)\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\right\|_F^2 &= \frac{1}{N}\sum_{n=1}^{N}\mathrm{tr}\left[\mathbf{a}_M^H(\theta_k)\mathbf{R}_M^{(i)^{-1}}\right.\\ &\quad\left.\times\mathbf{y}_M(n)\mathbf{y}_M^H(n)\mathbf{R}_M^{(i)^{-1}}\mathbf{a}_M(\theta_k)\right]\\ &= \frac{1}{N}\sum_{n=1}^{N}\left|\mathbf{a}_m^H(\theta_k)\mathbf{R}_M^{(i)^{-1}}\mathbf{y}_M(n)\right|^2. \end{aligned} \tag{8}$$

IV. FAST IMPLEMENTATION FOR UNIFORM LINEAR ARRAYS OF HYDROPHONES

The direct implementation of SPICE neglects the Toeplitz structure of the Hermitian matrix $\mathbf{R}_M$. Based on the GS factorization (e.g., Refs. 16 and 17), the inverse of the SPICE covariance matrix $\mathbf{R}_M$ can be represented by a series of Toeplitz matrices. This factorization improves the implementation efficiency of the matrix-vector product [i.e., $\mathbf{R}_M^{(i)^{-1}}\mathbf{y}_M(n)$ in Eqs. (7) and (8). Moreover, if the spatial frequency $f\left(f \triangleq \sin\theta\right)$ is uniformly sampled, $\mathbf{A}_M$ is the upper part of an FFT matrix.

A. Fast computation of the covariance matrix $\mathbf{R}_M$

Define $\underline{\mathbf{R}}_M \triangleq \mathbf{A}_M\mathbf{P}_K\mathbf{A}_M^H$. As the steering matrix $\mathbf{A}_M$ is a Vandermonde matrix, $\underline{\mathbf{R}}_M$ is a Hermitian Toeplitz matrix and it is fully specified by its first column, which is given by

$$\begin{aligned}\underline{\mathbf{R}}_M = \mathbf{A}_M\mathbf{P}_K\mathbf{A}_M^H &= \sum_{k=0}^{K-1}p_k\mathbf{a}_M(\theta_k)\mathbf{a}_M^H(\theta_k)\\ &= \begin{bmatrix} r_0 & r_1 & \cdots & r_{M-1}\\ r_1^* & r_0 & \cdots & r_{M-2}\\ \vdots & \vdots & \ddots & \vdots\\ r_{M-1}^* & r_{M-2}^* & \cdots & r_0\end{bmatrix}. \end{aligned} \tag{9}$$

From Eq. (9), and assume that spatial frequency is uniformly sampled, each element in Eq. (9) is specified by

$$r_m = \sum_{k=0}^{K-1}p_k e^{-j2\pi mk/K}, \qquad m = 0,\ldots,M-1, \tag{10}$$

which indicates that $\{r_m\}_{m=0}^{M-1}$ are the first M elements of the K-point FFT of $\{p_k\}_{k=0}^{K-1}$.

By retaining the first M elements of the FFT result, $\underline{\mathbf{R}}_M$ can be computed using FFT within $\mathcal{O}[K\log_2(K)]$ flops. Consequently, the first column of $\mathbf{R}_M$ can be obtained by adding σ to r_0 [see Eq. (3)].

B. Fast computation of $\mathbf{R}_M^{-1}\mathbf{y}_M(n)$

Define the vector $\mathbf{d}_M(n)$ as

$$\mathbf{d}_M(n) \triangleq \mathbf{R}_M^{-1}\mathbf{y}_M(n), \qquad n = 1,\ldots,N. \tag{11}$$

$\mathbf{d}_M(n)$ appears in Eqs. (7) and (8) and can be solved using the GS factorization and LDA once the first column of $\mathbf{R}_M$ is available.

Consider the partitioning of $\mathbf{R}_M$:

$$\mathbf{R}_M = \begin{bmatrix} r_0+\sigma & \mathbf{r}_{M-1}^H\\ \mathbf{r}_{M-1} & \mathbf{R}_{M-1}\end{bmatrix} \tag{12}$$

$$= \begin{bmatrix} \mathbf{R}_{M-1} & \check{\mathbf{r}}_{M-1}^*\\ \check{\mathbf{r}}_{M-1}^T & r_0+\sigma\end{bmatrix}, \tag{13}$$

where $\mathbf{r}_{M-1} \triangleq \left[r_1^*, r_2^*, \ldots, r_{M-1}^*\right]^T$, and $\check{\mathbf{r}}_{M-1}$ denotes the reversed row ordering version of $\mathbf{r}_{M-1}$, i.e., $\check{\mathbf{r}}_{M-1} = \mathbf{J}_{M-1}\mathbf{r}_{M-1} = \left[r_{M-1}^*, \ldots, r_2^*, r_1^*\right]^T$.

Applying the matrix inversion lemma (e.g., Ref. 18) to Eqs. (12) and (13) yields

$$\mathbf{R}_M^{-1} = \begin{bmatrix} 0 & 0\\ 0 & \mathbf{R}_{M-1}^{-1}\end{bmatrix} + \frac{1}{\alpha_{M-1}}\begin{bmatrix} 1\\ \mathbf{w}_{M-1}\end{bmatrix}\left[1, \mathbf{w}_{M-1}^H\right] \tag{14}$$

and

$$\mathbf{R}_M^{-1} = \begin{bmatrix} \mathbf{R}_{M-1}^{-1} & 0\\ 0 & 0\end{bmatrix} + \frac{1}{\alpha_{M-1}}\begin{bmatrix} \check{\mathbf{w}}_{M-1}^*\\ 1\end{bmatrix}\left[\check{\mathbf{w}}_{M-1}^T, 1\right], \tag{15}$$

respectively, where $\mathbf{w}_{M-1} = -\mathbf{R}_{M-1}^{-1}\mathbf{r}_{M-1}$, $\check{\mathbf{w}}_{M-1} = \mathbf{J}_{M-1}\mathbf{w}_{M-1}$, and $\alpha_{m-1} = r_0 + \sigma - \mathbf{r}_{M-1}^H\mathbf{R}_{M-1}^{-1}\mathbf{r}_{M-1}$.

Based on Eqs. (14) and (15), the displacement representation (see, e.g., Ref. 19) of $\mathbf{R}_M^{-1}$ is given by

$$\begin{aligned}\nabla_{\mathbf{z}_M,\mathbf{z}_M^T}\mathbf{R}_M^{-1} &\triangleq \mathbf{R}_M^{-1} - \mathbf{Z}_M\mathbf{R}_M^{-1}\mathbf{Z}_M^T\\ &= \frac{1}{\alpha_{M-1}}\begin{bmatrix} 1\\ \mathbf{w}_{M-1}\end{bmatrix}\left[1, \mathbf{w}_{M-1}^H\right]\\ &\quad - \frac{1}{\alpha_{M-1}}\begin{bmatrix} 0\\ \mathbf{w}_{M-1}^*\end{bmatrix}\left[0, \mathbf{w}_{M-1}^T\right], \end{aligned} \tag{16}$$

where $\mathbf{Z}_M$[20] is an $M\times M$ matrix with ones on the first subdiagonal and zeros everywhere else. Hence the GS formula (see, e.g., Refs. 5 and 21) of $\mathbf{R}_M^{-1}$ is given by

$$\mathbf{R}_M^{-1} = \mathcal{L}_M(\mathbf{t}_1, \mathbf{Z}_M)\mathcal{L}_M^H(\mathbf{t}_1, \mathbf{Z}_M) - \mathcal{L}_M(\mathbf{t}_2, \mathbf{Z}_M)\mathcal{L}_M^H(\mathbf{t}_2, \mathbf{Z}_M), \tag{17}$$

where $\mathbf{t}_1 = \left(1/\sqrt{\alpha_{M-1}}\right)[1, \mathbf{w}_{M-1}]^T$, $\mathbf{t}_2 = \left(1/\sqrt{\alpha_{M-1}}\right)\left[0, \mathbf{J}_{M-1}\mathbf{w}_{M-1}^*\right]^T$, and $\mathcal{L}_M(\mathbf{t}, \mathbf{Z}) \triangleq \left[\mathbf{t}, \mathbf{Z}\mathbf{t}, \ldots, \mathbf{Z}^{M-1}\mathbf{t}\right]$ denotes a Krylov matrix

Based on Eqs. (16) and (17), $\mathbf{R}_M^{-1}$ only depends on $\mathbf{w}_{M-1}$ and α_{M-1}, both of which can be computed using the Levinson–Durbin equations detailed in Appendix A and summarized in Table II.

Note that the displacement, $\nabla_{\mathbf{z}_M, \mathbf{z}_M^T}\mathbf{R}_M^{-1}$ is of rank 2. This is in fact a property for all Toeplitz nonsingular matrices: The inverse shares the same displacement rank as the original matrix (e.g., Refs. 1, 5, and 21). Also note that in Eq. (17) both $\mathcal{L}_M(\mathbf{t}_1, \mathbf{Z}_M)$ and $\mathcal{L}_M(\mathbf{t}_2, \mathbf{Z}_M)$ are lower triangular Toeplitz matrices, which makes possible the computation of matrix vector product $\mathbf{R}_M^{-1}\mathbf{y}_M(n)$ in Eq. (11) using FFT (detailed in Appendix E).

The fast implementation of SPICE for hydrophone ULAs is summarized as follows: For each iteration of SPICE, do the following:

(1) Given $\left\{p_k^{i-1}\right\}_{k=0}^{K-1}$ from the previous iteration, compute $\{r_m\}_{m=0}^{M-1}$ in Eq. (10) using FFT with $\mathcal{O}(K \log_2 K)$ flops.
(2) Given $\{r_m\}_{m=0}^{M-1}$, compute the generators $\mathbf{w}_{M-1}$ and α_{M-1} using LDA detailed in Table II with $\mathcal{O}(M^2)$ flops.
(3) Given the generators, calculate $\{\mathbf{R}_M^{(i)^{-1}}\mathbf{y}_M(n)\}_{n=1}^N$ in Eq. (7) using FFT with $\mathcal{O}(MN \log_2 M)$ flops.
(4) Given $\{\mathbf{R}_M^{(i)^{-1}}\mathbf{y}_M(n)\}_{n=1}^N$, calculate $\{\mathbf{a}_M^H(\theta_k)\mathbf{R}_M^{(i)^{-1}}\mathbf{y}_M(n)\}_{n=1,k=0}^{N,K-1}$ in Eq. (8) using FFT with $\mathcal{O}(NK \log_2 K)$ flops.
(5) Given $\{\mathbf{R}_M^{(i)^{-1}}\mathbf{y}_M(n)\}_{n=1}^N$ and $\{\mathbf{a}_M^H(\theta_k)\mathbf{R}_M^{(i)^{-1}}\mathbf{y}_M(n)\}_{n=1,k=0}^{N,K-1}$ calculate $\left\|\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\right\|_F$ in Eq. (7) and $\left\|\mathbf{a}_M^H(\theta_k)\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\right\|_F$ in Eq. (8) with $\mathcal{O}(MN)$ and $\mathcal{O}(NK)$ flops, respectively.
(6) Given $\left\|\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\right\|_F$ and $\left\|\mathbf{a}_M^H(\theta_k)\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\right\|_F$, calculate $\rho^{(i)}$ in $\mathcal{O}(K)$ flops.
(7) Given $\left\|\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\right\|_F$, calculate $\sigma^{(i)}$ in $\mathcal{O}(1)$ flops.
(8) Given $\left\|\mathbf{a}_M^H(\theta_k)\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\right\|_F$, update $\left\{p_k^{(i)}\right\}_{k=0}^{K-1}$ in $\mathcal{O}(K)$ flops.

V. FAST IMPLEMENTATION OF SPICE FOR UNIFORM LINEAR ARRAYS OF VECTOR SENSORS

A. Source localization with ULA of vector sensors

A vector-sensor array consists of a series of vector-sensor elements, each of which incorporates one conventional hydrophone and a series of geophones. The set of geophones are typically mounted at the same location as the hydrophone and orthogonal to each other. The geophones directly collect directional information such as particle velocities or accelerations, which contributes to advantageous properties compared to 1D hydrophone arrays, such as better resolution for fixed aperture, less aliasing effects in spatially undersampled wavefield and absence of left/right ambiguities (e.g., Refs. 13, 22, and 23).

TABLE II. Levinson–Durbin alogrithm for generators $\mathbf{w}_t$ and α_t.

Initialization	$w_1 = -\frac{r_1^*}{r_0+\sigma}$ and $\alpha_1 = r_0 + \sigma + r_1 w_1$
For $t = 2, \ldots, M-1$	$\psi_{t-1} = \mathbf{w}_{t-1}^T\mathbf{J}_{t-1}\mathbf{r}_{t-1} + r_t^*$
	$\mathbf{w}_t = \begin{bmatrix}\mathbf{w}_{t-1}\\ 0\end{bmatrix} - \frac{1}{\alpha_{t-1}}\begin{bmatrix}\mathbf{J}_{t-1}\mathbf{w}_{t-1}^*\\ 1\end{bmatrix}\psi_{t-1}$
	$\alpha_t = \alpha_{t-1} - \lvert\psi_{t-1}\rvert^2/\alpha_{t-1}$

Signals collected from each vector-sensor element typically consists of both the acoustic pressure measurement p and the orthogonal directional velocities measurements u_x, u_y, u_z. Assume that the orientations of the geophones in each vector-sensor element are known, velocity measurements from all elements can be rotated accordingly to be aligned with the global coordinates. After this aligning and scaling procedure, the measurement of a single vector-sensor element can be denoted as a scalar-vector product $p \cdot \mathrm{h}_D(\theta_i)$, and the steering vector $\mathbf{a}_{DM}$ of the vector-sensor array is given by (e.g., Ref. 13)

$$\mathbf{a}_{DM}(\theta_i) = \mathbf{a}_M(\theta_i) \otimes \mathbf{h}_D(\theta_i), \tag{18}$$

where $\mathbf{a}_M(\theta_i)$ denotes the steering vector of the array that consists of only the hydrophones in the vector-sensor array, $\mathbf{h}_D(\theta_i)$ denotes the real-valued response of a single vector-sensor element and θ_i denotes a generic bearing parameter. In the general case, denote $\mathbf{h}(\theta_i)$ as

$$\mathbf{h}_D(\theta_i) \triangleq [1, h_2(\theta_i), \ldots, h_D(\theta_i)]^T, \tag{19}$$

where $h_d(\theta_i)$ denotes the dth element in $\mathbf{h}_D(\theta_i)$, $d = 2, \ldots, D$. From Eq. (18), the steering matrix $\tilde{\mathbf{A}}_{DM}$ of the vector-sensor array is given by

$$\tilde{\mathbf{A}}_{DM} \triangleq [\mathbf{a}_{DM}(\theta_0), \mathbf{a}_{DM}(\theta_1), \ldots, \mathbf{a}_{DM}(\theta_{K-1})]. \tag{20}$$

The noise term for the vector-sensor array measurements is much more complicated with nondiagonal elements in the covariance matrix even in the isotropic Gaussian noise field (e.g., Refs. 24–26, and references therein). However, a simplified nonuniform white Gaussian noise model in the following equation[27] can be applied here with only minor performance degradation, thanks to the robustness of the SPICE algorithm (see Refs. 3 and 4, which will also be illustrated numerically in Sec. VI). Consider an M-element vector-sensor array, where each element consists of one omnidirectional hydrophone and $(D-1)$ acoustic velocity sensors. Each measurement from this array forms a $DM \times 1$ column vector $\tilde{\mathbf{y}}_{DM}$:

$$\tilde{\mathbf{y}}_{DM}(n) = \tilde{\mathbf{A}}_{DM}\mathbf{x}(n) + \tilde{\mathbf{e}}_{DM}(n), \qquad n = 1, \ldots, N, \tag{21}$$

where $E\left[\tilde{\mathbf{e}}_{DM}(n)\tilde{\mathbf{e}}_{DM}^H(\bar{n})\right] = (\mathbf{I}_M \otimes \mathrm{Diag}(\sigma_1, \ldots, \sigma_D))\delta_{n,\bar{n}}$ (see, e.g., Ref. 13). Based on this assumption on noise, the $DM \times DM$ array output covariance matrix $\tilde{\mathbf{R}}_{DM}$ is given by

$$\tilde{\mathbf{R}}_{DM} = \tilde{\mathbf{A}}_{DM}\mathbf{P}_K\tilde{\mathbf{A}}_{DM}^H + \mathbf{I}_M \otimes \mathrm{Diag}(\sigma_1, \ldots, \sigma_D), \tag{22}$$

and it can be estimated by the sample covariance matrix $\hat{\mathbf{R}}_{DM}$. Reminiscent of the SPICE derivations in Ref. 4, the estimates of the signal powers $\{\mathbf{p}_k\}_{k=0}^{K-1}$ and noise powers $\{\sigma_d\}_{d=1}^{D}$ can be obtained by solving the problem in [Eq. (22) in Ref. 4]. Analogous to [Eq. (29) in Ref. 4] and by incorporating the noise model in Eq. (21), this minimization problem is simplified as

$$\sum_{k=1}^{K}\frac{\|\mathbf{c}_k\|^2}{p_k}+\sum_{d=1}^{D}\sum_{m=1}^{M}\frac{\left\|\mathbf{c}_{K+d+(m-1)D}\right\|^2}{\sigma_2}, \tag{23}$$

subject to

$$\sum_{k=1}^{K}\omega_k p_k+\sum_{d=1}^{D}\gamma_d\sigma_d=1, \tag{24}$$

where

$$\gamma_d \overset{\Delta}{=} \sum_{m=1}^{M}\omega_{K+d+(m-1)D}, \tag{25}$$

and $\{\mathbf{c}_k\}_{k=1}^{K+MD}$ are defined in Eq. (28) of Ref. 4. By the Cauchy–Schwarz inequality, we have

$$\begin{aligned}
&\left[\sum_{k=1}^{K}\omega_k^{1/2}\|\mathbf{c}_k\|+\sum_{d=1}^{D}\gamma_d^{1/2}\sqrt{\sum_{m=1}^{M}\|\mathbf{c}_{K+d+(m-1)D}\|^2}\right]^2\\
&=\left[\sum_{k=1}^{K}\omega_k^{1/2}p_k^{1/2}\frac{\|\mathbf{c}_k\|}{p_k^{1/2}}+\sum_{d=1}^{D}\gamma_d^{1/2}\sigma_d^{1/2}\right.\\
&\quad\left.\times\frac{\sqrt{\sum_{m=1}^{M}\|\mathbf{c}_{K+d+(m-1)D}\|^2]^2}}{\sigma_d^{1/2}}\right]\le\left[\sum_{k=1}^{K}\frac{\|\mathbf{c}_k\|^2}{p_k}\right.\\
&\quad\left.+\sum_{d=1}^{D}\frac{\sum_{m=1}^{M}\|\mathbf{c}_{K+d+(m-1)D}\|^2}{\sigma_d}\right]\left[\sum_{k=1}^{K}\omega_k p_k+\sum_{d=1}^{D}\gamma_d\sigma_d\right]^2\\
&=\sum_{k=1}^{K}\frac{\|\mathbf{c}_k\|^2}{p_k}+\sum_{d=1}^{D}\sum_{m=1}^{M}\frac{\|\mathbf{c}_{K+d+(m-1)D}\|^2}{\sigma_d}.
\end{aligned} \tag{26}$$

The previous equality holds if and only if

$$\frac{\sqrt{\sum_{m=1}^{M}\left\|\mathbf{c}_{K+d+(m-1)D}\right\|^2}}{\sqrt{\sigma_d}}=\rho\gamma_d^{1/2}\sigma_d^{1/2}, \tag{27}$$

and similarly for the signal powers:

$$\frac{\|\mathbf{c}_k\|}{p_k^{1/2}}=\rho\omega_k^{1/2}p_k^{1/2}. \tag{28}$$

Therefore, the solutions to Eq. (23) are

$$p_k=\frac{\|\mathbf{c}_k\|}{\omega_k^{1/2}\rho},\quad k=1,\ldots,K, \tag{29}$$

$$\sigma_d=\frac{\left[\sum_{m=1}^{M}\left\|\mathbf{c}_{K+d+(m-1)D}\right\|^2\right]^{1/2}}{\gamma_d^{1/2}\rho},\quad d=1,\ldots,D, \tag{30}$$

where

$$\rho\overset{\Delta}{=}\sum_{k=1}^{K}\omega_k^{1/2}\|\mathbf{c}_k\|+\sum_{d=1}^{D}\left\{\gamma_d^{1/2}\left[\sum_{m=1}^{M}\left\|\mathbf{c}_{K+d+(m-1)D}\right\|^2\right]^{1/2}\right\}. \tag{31}$$

Substituting Eqs. (29)–(31) into Eq. (23) of Ref. 4, the SPICE iterative steps for the vector-sensor array is given as follows:

Initialize signal power estimates using DAS and $\{\sigma_d^{(0)}\}_{d=1}^{D}$ with a series of small values (e.g., the average of the smallest M values in $\{p_k^{(0)}\}_{k=0}^{K-1}$). At the ith iteration,

(1) Update the $\tilde{\mathbf{R}}_{DM}(i)$ using Eq. (22),
(2) Compute the auxiliary variable $\rho(i)$:

$$\begin{aligned}
\rho(i)=&\sum_{k=1}^{K}\omega_k^{1/2}p_k^{(i-1)}\left\|\tilde{\mathbf{a}}_{DM}(\theta_k)^H\tilde{\mathbf{R}}_{DM}^{-1}(i)\tilde{\mathbf{R}}_{DM}^{1/2}\right\|.\\
&+\sum_{d=1}^{D}\gamma_d^{1/2}\sigma_d^{(i-1)}\left\|\left(\tilde{\mathbf{R}}_{DM}^{-1}(i)\hat{\mathbf{R}}_{DM}^{1/2}\right)\odot(\mathbf{1}_{M\times DM}\otimes\mathbf{u}_d)\right\|_F.
\end{aligned} \tag{32}$$

(3) Update the noise variance estimates,

$$\sigma_d^{(i)}=\sigma_d^{(i-1)}\frac{\left\|\left(\tilde{\mathbf{R}}_{DM}^{-1}(i)\hat{\mathbf{R}}_{DM}^{1/2}\right)\odot(\mathbf{1}_{M\times DM}\otimes\mathbf{u}_d)\right\|_F}{\gamma_d^{1/2}\rho(i)},\quad d=1,\ldots,D, \tag{33}$$

where $\mathbf{u}_d$ denotes the dth column of the identity matrix $\mathbf{I}_D$

(4) Update source power estimates:

$$p_k^{(i)}=p_k^{(i-1)}\frac{\left\|\tilde{\mathbf{a}}_{DM}(\theta_k)^H\tilde{\mathbf{R}}_{DM}^{-1}(i)\hat{\mathbf{R}}_{DM}^{1/2}\right\|}{\omega_k^{1/2}\rho(i)},\quad k=1,\ldots,K, \tag{34}$$

until practical convergence.

The term $\left\|\left(\tilde{\mathbf{R}}_{DM}^{-1}(i)\hat{\mathbf{R}}_{DM}^{1/2}\right)\odot(\mathbf{1}_{M\times DM}\otimes\mathbf{u}_d)\right\|_F$ can be obtained by first calculating the $DM\times DM$ dimensional matrix $\tilde{\mathbf{R}}_{DM}^{-1}(i)\hat{\mathbf{R}}_{DM}^{1/2}$, and then only selecting the elements at row d, $d+D$, $d+2D$, and computing their Frobenius norm subsequently.

B. Computation of the block Toeplitz covariance matrix in vector sensor arrays

As the number of sensors [DM in Eq. (22)] or the number of scanning angles (K) increases, the computational complexity of $\tilde{\mathbf{R}}_{DM}$ increases as well. However, the block Toeplitz matrix structure can be exploited to compute $\tilde{\mathbf{R}}_{DM}$ efficiently,

$$\tilde{\mathbf{R}}_{DM}=\sum_{k=1}^{K-1}p_k\mathbf{a}_{DM}(\theta_k)\mathbf{a}_{DM}^H(\theta_k)+\mathbf{I}_M\otimes\mathrm{Diag}(\sigma_1,\ldots,\sigma_D) \tag{35}$$

$$= \begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_1 & \mathbf{R}_2 & \cdots & \mathbf{R}_{M-1} \\ \mathbf{R}_1^* & \mathbf{R}_0 & \mathbf{R}_1 & \cdots & \mathbf{R}_{M-2} \\ \mathbf{R}_2^* & \mathbf{R}_1^* & \mathbf{R}_0 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{M-1}^* & \mathbf{R}_{M-2}^* & \cdots & \cdots & \mathbf{R}_0 \end{bmatrix} + \mathbf{I}_M \otimes \mathrm{Diag}(\sigma_1, ..., \sigma_D). \tag{36}$$

Analogous to Eq. (10),

$$\mathbf{R}_m = \sum_{k=0}^{K-1} p_k \left(\mathbf{h}_D(\theta_k)\mathbf{h}_D^T(\theta_k)\right) e^{-j2\pi mk/K}, \quad m = 0, ..., M-1. \tag{37}$$

From Eqs. (19) and (37), each block $\mathbf{R}_m$ is a symmetric matrix, and it is possible to calculate its elements using FFT. For the notational simplicity, we temporarily assume $D = 3$, and the block $\mathbf{R}_m$ has the following structure:

$$\mathbf{R}_m = \begin{bmatrix} r_{m,1} & r_{m,2} & r_{m,4} \\ r_{m,2} & r_{m,3} & r_{m,5} \\ r_{m,4} & r_{m,5} & r_{m,6} \end{bmatrix}. \tag{38}$$

Note that the elements of $\mathbf{R}_m$ defined in Eq. (38) can be computed from p_k and $\mathbf{h}_d(\theta_i)$ using FFT:

$$r_{m,1} = \sum_{k=0}^{K-1} p_k e^{-j2\pi mk/K},$$

$$r_{m,2} = \sum_{k=0}^{K-1} (p_k h_2(\theta_k)) e^{-j2\pi mk/K},$$

$$r_{m,3} = \sum_{k=0}^{K-1} \left(p_k h_2^2(\theta_k)\right) e^{-j2\pi mk/K},$$

$$r_{m,4} = \sum_{k=0}^{K-1} (p_k h_3(\theta_k)) e^{-j2\pi mk/K,}$$

$$r_{m,5} = \sum_{k=0}^{K-1} (p_k h_2(\theta_k) h_3(\theta_k)) e^{-j2\pi mk/K},$$

$$r_{m,6} = \sum_{k=0}^{K-1} \left(p_k h_3^2(\theta_k)\right) e^{-j2\pi mk/K}.$$

C. Displacement representation of $\tilde{\mathbf{R}}_{DM}^{-1}$

Given the block Toeplitz covariance matrix $\tilde{\mathbf{R}}_{DM}$, the generalized LDA can be performed to compute the generators of displacement $\nabla z_M, z_M^T \tilde{\mathbf{R}}_{DM}^{-1}$.

According to the Toeplitz block structure of $\tilde{\mathbf{R}}_{DM}$ in Eq. (36), it can be expressed as

$$\tilde{\mathbf{R}}_{DM} = \begin{bmatrix} \mathbf{R}_0 & \bar{\mathcal{R}}_{D(M-1)}^H \\ \bar{\mathcal{R}}_{D(M-1)} & \tilde{\mathbf{R}}_{D(M-1)} \end{bmatrix} \tag{39}$$

$$= \begin{bmatrix} \tilde{\mathbf{R}}_{D(M-1)} & \hat{\mathcal{R}}_{D(M-1)} \\ \hat{\mathcal{R}}_{D(M-1)}^H & \mathbf{R}_0 \end{bmatrix}, \tag{40}$$

where $\bar{\mathcal{R}}_{Dm}$ and $\hat{\mathcal{R}}_{Dm}$ denote block matrix of dimension $Dm \times D$: $\bar{\mathcal{R}}_{Dm} = \left[\mathbf{R}_1^H, ..., \mathbf{R}_m^H\right]^T$ and $\hat{\mathcal{R}}_{Dm} = \mathbf{J}_{Dm}\bar{\mathcal{R}}_{Dm}^* \mathbf{J}_D$, $m = 1, ..., M-1$.

Application of the matrix inversion lemma (e.g., Ref. 18) for partitioned matrices to Eqs. (39) and (40) yields, respectively:

$$\tilde{\mathbf{R}}_{DM}^{-1} = \begin{bmatrix} \tilde{\mathbf{R}}_{D(M-1)}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \bar{\mathcal{B}}_{DM}\bar{\mathcal{B}}_{DM}^H \tag{41}$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & \tilde{\mathbf{R}}_{D(M-1)}^{-1} \end{bmatrix} + \bar{\mathcal{A}}_{DM}\bar{\mathcal{A}}_{DM}^H, \tag{42}$$

where $\bar{\mathcal{A}}_{DM}$ and $\bar{\mathcal{B}}_{DM}$ are block matrices of dimension $DM \times M$, given by

$$\bar{\mathcal{A}}_{DM} \triangleq \begin{bmatrix} \mathbf{I}_M \\ \mathcal{A}_{D(M-1)} \end{bmatrix} \mathbf{Q}_{M-1}^{-1/2}, \tag{43}$$

$$\bar{\mathcal{B}}_{DM} \triangleq \begin{bmatrix} \mathbf{I}_M \\ \mathcal{B}_{D(M-1)} \end{bmatrix} \Delta_{M-1}^{-1/2}, \tag{44}$$

respectively, where

$$\mathcal{A}_{D(M-1)} = -\tilde{\mathbf{R}}_{D(M-1)}^{-1} \bar{\mathcal{R}}_{D(M-1)}, \tag{45}$$

$$\mathcal{B}_{D(M-1)} = -\tilde{\mathbf{R}}_{D(M-1)}^{-1} \hat{\mathcal{R}}_{D(M-1)}, \tag{46}$$

$$\mathbf{Q}_{M-1} = \mathbf{R}_0 + \bar{\mathcal{R}}_{D(M-1)}^H \mathcal{A}_{D(M-1)}, \tag{47}$$

$$\Delta_{M-1} = \mathbf{R}_0 + \hat{\mathcal{R}}_{D(M-1)}^H \mathcal{B}_{D(M-1)}. \tag{48}$$

By the persymmetry property of $\tilde{\mathbf{R}}_{DM}$ (e.g., Refs. 28 and 29), i.e.,

$$\mathbf{J}_{Dm}\tilde{\mathbf{R}}_{Dm}\mathbf{J}_{Dm} = \tilde{\mathbf{R}}_{Dm}^T, \qquad m = 1, \ldots, M \tag{49}$$

the matrices $\bar{\mathcal{B}}_{D(M-1)}$ and $\mathcal{A}_{D(M-1)}$, $\mathbf{\Delta}_{M-1}$, and $\mathbf{Q}_{M-1}$ are related through

$$\mathcal{A}_{D(M-1)} = \mathbf{J}_{D(M-1)}\mathcal{B}_{D(M-1)}^* \mathbf{J}_D, \tag{50}$$

$$\mathbf{Q}_{M-1} = \mathbf{J}_D \Delta_{M-1}^T \mathbf{J}_D. \tag{51}$$

By using Eqs. (41)–(51), the displacement representation (DR) of $\tilde{\mathbf{R}}_{DM}^{-1}$ takes the following form:

$$\nabla_{Z_M \otimes I_D, \mathbf{Z}_M^T \otimes I_D} \tilde{\mathbf{R}}_{DM}^{-1} = \bar{\mathcal{A}}_{DM}\bar{\mathcal{A}}_{DM}^H - (\mathbf{Z}_M \otimes \mathbf{I}_D) \times \bar{\mathcal{B}}_{DM}\bar{\mathcal{B}}_{DM}^H (\mathbf{Z}_M^T \otimes \mathbf{I}_D). \tag{52}$$

Note that the generators $\mathcal{B}_{D(M-1)}$ and $\mathbf{\Delta}_{M-1}$ of $\tilde{\mathbf{R}}_{DM}^{-1}$ involved in Eq. (52) are all block matrices related to the forward predictor [Eqs. (41)–(51)]. These generators can be computed

TABLE III. The *L–D*-type algorithm for generators $\mathcal{B}_{Dm}$ and Δ_m.

Initialization	$\mathcal{B}_{D1} = \mathbf{R}_0^{-1}\mathbf{R}_1^*$ and $\Delta_1 = \mathbf{R}_0 - \mathbf{R}_1^H\mathbf{R}_0^{-1}\mathbf{R}_1$
For $m = 2,\ldots, M-1$	$\mathbf{\Omega}_m = \Delta_{m-1}^{-*}\left[\mathbf{R}_m + \mathcal{B}_{D(m-1)}^T\bar{\mathcal{R}}_{D(m-1)}^*\right]\mathbf{J}_D$
	$\mathcal{B}_{Dm} = \begin{bmatrix} 0 \\ \mathcal{B}_{D(m-1)} \end{bmatrix} + \mathbf{J}_{Dm}\begin{bmatrix} \mathcal{B}_{D(m-1)}^* \\ \mathbf{I}_D \end{bmatrix}\mathbf{\Omega}_m$
	$\Delta_m = \Delta_{m-1} - \Omega_m^H\Delta_{m-1}^T\Omega_m$

thanks to the following lemma proved in Appendix B and summarized by Table III.

Lemma 1: *For* $m = 2,\ldots, M-1$,

$$\mathcal{B}_{Dm} = \begin{bmatrix} \mathbf{0} \\ \mathcal{B}_{D(m-1)} \end{bmatrix} + \mathbf{J}_{Dm}\begin{bmatrix} \mathcal{B}_{D(m-1)}^* \\ \mathbf{I}_D \end{bmatrix}\Omega_m, \tag{53}$$

where $\Omega_m \triangleq \Delta_{m-1}^{-*}\left[\mathbf{R}_m + \mathcal{B}_{D(m-1)}^T\bar{\mathcal{R}}_{D(m-1)}^*\right]\mathbf{J}_D$, and

$$\Delta_m = \Delta_{m-1} - \Omega_m^H\Delta_{m-1}^T\Omega_m. \tag{54}$$

Once the DR of the matrix $\tilde{\mathbf{R}}_{DM}^{-1}$ is obtained, the GS factorization of $\tilde{\mathbf{R}}_{DM}^{-1}$ can be subsequently computed. Let $\{\mathbf{t}_{DM}^i\}_{i=1}^D$ denote the D columns of the $\bar{\mathcal{A}}_{DM}$ and $\{\mathbf{t}_{DM}^i\}_{i=D+1}^{2D}$ denote the D columns of $(Z_M \otimes I_D)\bar{\mathcal{B}}_{DM}$. Columns and rows in Eq. (52) can be expressed explicitly as follows:

$$\nabla_{\mathbf{Z}_M\otimes\mathbf{I}_D,\mathbf{Z}_M^T\otimes\mathbf{I}_D}\tilde{R}_{DM}^{-1} \triangleq \sum_{i=1}^{2D}\sigma_i\mathbf{t}_{DM}^i\mathbf{t}_{DM}^{iH}, \tag{55}$$

where $\sigma_i = 1$ for $i = 1,., D$ and $\sigma_i = -1$ for $i = (D+1),\ldots,2D$.

Given Eq. (55), the GS factorization of $\tilde{\mathbf{R}}_{DM}^{-1}$ takes the following form:

$$\tilde{\mathbf{R}}_{DM}^{-1} = \sum_{j=0}^{M-1}(\mathbf{Z}_M\otimes\mathbf{I}_D)^j(\nabla_{\mathbf{Z}_M\otimes\mathbf{I}_D,\mathbf{Z}_M^T\otimes\mathbf{I}_D}\tilde{\mathbf{R}}_{DM}^{-1})(\mathbf{Z}_M^T\otimes\mathbf{I}_D)^j \tag{56}$$

$$= \sum_{i=1}^{2D}\sigma_i\left[\sum_{j=0}^{M-1}(\mathbf{Z}_M\otimes\mathbf{I}_D)^j(\mathbf{t}_{DM}^i\mathbf{t}_{DM}^{iH})(\mathbf{Z}_M^T\otimes\mathbf{I}_D)^j\right] \tag{57}$$

$$= \sum_{i=1}^{2D}\sigma_i\mathcal{L}_{DM}(\mathbf{t}_{DM}^i, \mathbf{Z}_M\otimes\mathbf{I}_D)\mathcal{L}_{DM}^H(t_{DM}^i, \mathbf{Z}_M\otimes\mathbf{I}_D), \tag{58}$$

where

$$\mathcal{L}_{DM}(\mathbf{t}_{DM}^i, \mathbf{Z}_M\otimes\mathbf{I}_D) = \left[\mathbf{t}_{DM}^i, (\mathbf{Z}_M\otimes\mathbf{I}_D)\mathbf{t}_{DM}^i, \ldots, (\mathbf{Z}_M\otimes\mathbf{I}_D)^{M-1}\mathbf{t}_{DM}^i\right]. \tag{59}$$

Due to the fact that $\mathcal{L}_{DM}\left(t_{DM}^i, Z_M\otimes I_D\right)$ is a $DM\times M$ lower block triangular Toeplitz matrix, the matrix-vector product $\tilde{\mathbf{R}}_{DM}^{-1}\mathbf{y}_{DM}$ can be efficiently computed using FFT (detailed in Appendices C and D).

The fast implementation of SPICE with vector-sensor ULAs is summarized as follows:

For each iteration of SPICE, do the following:

(1) Given $\left\{p_k^{i-1}\right\}_{k=0}^{K-1}$ from the previous iteration, compute the auxiliary matrix $\bar{\mathcal{R}}_{D(M-1)}$ using a series of FFTs with $\mathcal{O}(DK\log_2 K)$ flops.

(2) Given $\bar{\mathcal{R}}_{D(M-1)}$, compute the generators $\mathcal{B}_{D(M-1)}$ and Δ_{M-1} using the generalized LDA with $\mathcal{O}(DM^2)$ flops.

(3) Given the generators, calculate $\tilde{\mathbf{R}}_{DM}^{-1}\hat{\mathbf{R}}_{DM}^{1/2}$ using FFT with $\mathcal{O}(DM^2\log_2 M)$ flops.

(4) Given $\tilde{\mathbf{R}}_{DM}^{-1}\hat{\mathbf{R}}_{DM}^{1/2}$, calculate $\mathbf{a}_{DM}(\theta_k)\mathbf{R}_{DM}^{-1}\hat{\mathbf{R}}_{DM}^{1/2}$, for $k = 0,1,\ldots, K-1$, using FFT with $\mathcal{O}(DMK\log_2 K)$ flops.

(5) Given $\tilde{\mathbf{R}}_{DM}^{-1}\hat{\mathbf{R}}_{DM}^{1/2}$, calculate all $\left\|\left(\tilde{\mathbf{R}}_{DM}^{-1}\hat{\mathbf{R}}_{DM}^{1/2}\right)\odot(\mathbf{1}_{M\times DM}\otimes\mathbf{u}_d)\right\|_F$, $d = 1,\ldots, D$ in $\mathcal{O}(D^2M^2)$ flops.

(6) Given $\mathbf{a}_{DM}(\theta_k)\tilde{\mathbf{R}}_{DM}^{-1}\hat{\mathbf{R}}_{DM}^{1/2}$, $k = 0,\ldots, K-1$ and $\left\|\tilde{\mathbf{R}}_{DM}^{-1}\hat{\mathbf{R}}_{DM}^{1/2}\right\|_F$, calculate $\rho^{(i)}$ in $\mathcal{O}(K+D)$ flops.

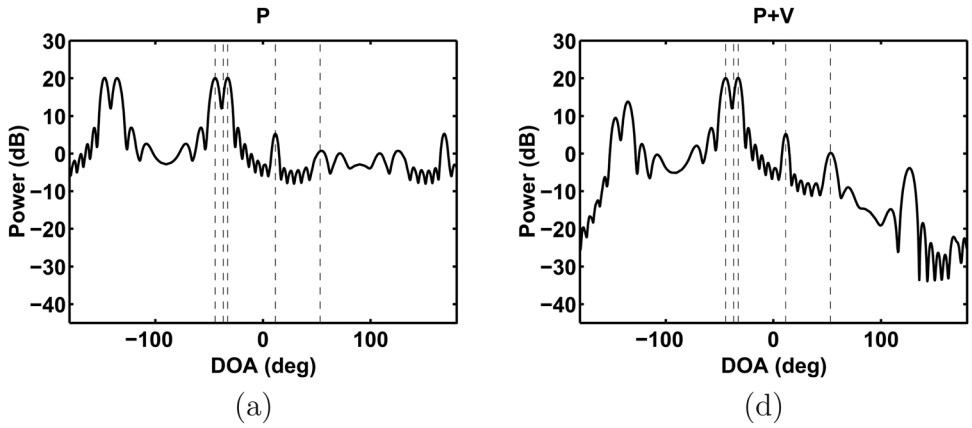
(7) Given $\left\|\left(\tilde{\mathbf{R}}_{DM}^{-1}\hat{\mathbf{R}}_{DM}^{1/2}\right)\odot(\mathbf{1}_{M\times DM}\otimes\mathbf{u}_d)\right\|_F$, $d = 1,\ldots, D$, calculate $\{\sigma_d^{(i)}\}_{d=1}^D$ in $\mathcal{O}(D)$ flops.

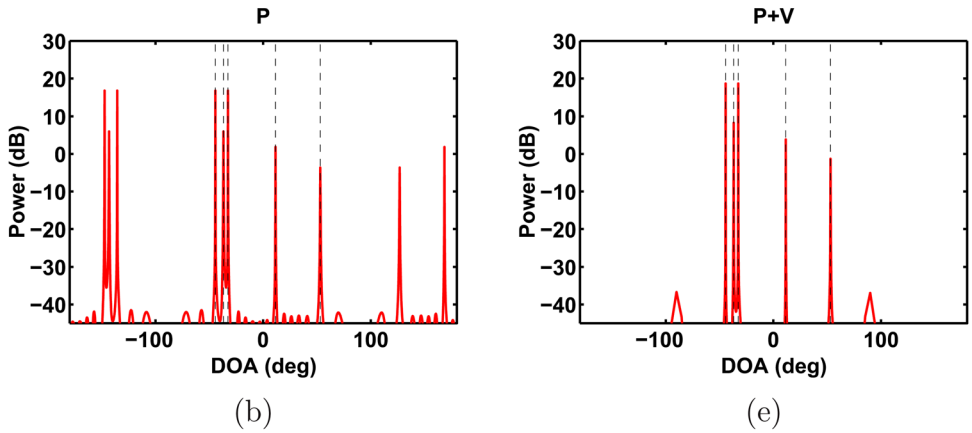
(8) Given $\mathbf{a}_{DM}(\theta_k)\tilde{R}_{DM}^{-1}\hat{R}_{DM}^{1/2}$, $k = 0\ldots K-1$, update $\{p_k^{(i)}\}_{k=0}^{K-1}$ in $\mathcal{O}(K)$ flops.

VI. NUMERICAL EXAMPLES

In this section, numerical examples are provided to compare the computational complexity of the direct implementation and the proposed fast implementations of SPICE. This section focuses on the DOA estimation problem of fixed sources with both hydrophone ULAs and vector-sensor ULAs. Without any loss of generality, only the estimation of the azimuth angle is considered herein.

In all of the following figures, five uncorrelated sources are present with the following true angles-of-arrival: $\theta_1 = -44.4°$, $\theta_2 = -36.9°$, $\theta_3 = -32.7°$, $\theta_4 = 11.5°$ and $\theta_5 = 53.1°$. The corresponding truth signals are $x_1(n) = 10e^{i\xi_1(n)}$, $x_2(n) = 3e^{i\xi_2(n)}$, $x_3(n) = 10e^{i\xi_3(n)}$, $x_4(n) = 1.8e^{i\xi_4(n)}$ and $x_5(n) = e^{i\xi_5(n)}$, where $n \in \{1,\ldots, N\}$ denotes the index of the available snapshots. The phase values $\{\xi_1(n)\},\ldots,$ $\{\xi_5(n)\}$ are independent and identically distributed random variables with uniform distribution over $[0, 2\pi)$. The ambient noise is assumed to be temporally isotropic white and the "diffuse" noise model in Ref. 26 is applied. For the hydrophone ULAs with half-wavelength interelement spacing, the noise term is circularly symmetric complex-valued white Gaussian random processes. For the vector-sensor ULAs with half-wavelength interelement spacing, the noise covariance matrix Q_{dm} has complicated structures.[25,26] In the following simulations, we consider a vector-sensor ULA with 2-D velocity measurements in the *x–y* plane, therefore, the measurements is given by $\tilde{\mathbf{y}}_{DM} = \left[p_{r1}, \bar{u}_{r1}^x, \bar{u}_{r1}^y, p_{r2}, \bar{u}_{r2}^x, \bar{u}_{r2}^y, \ldots\right]^T$, where p_{r1} denotes the acoustic pressure measurement at location $r1$, and $\bar{u}_{r1}^x$ and $\bar{u}_{r1}^y$ denote, respectively, directional velocities u_x and u_y scaled by the acoustic impedance (ρc). Assume that the noise power at the array origin is σ. The corresponding covariance matrix can be obtained by computing the interelement correlations (see Refs. 25 and 26):





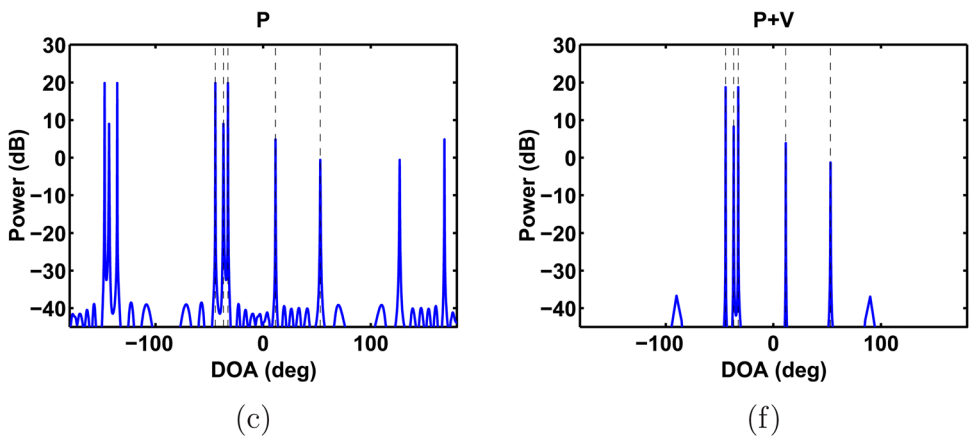


FIG. 1. (Color online) DOA estimation based on measurements of a hydrophone ULA (a)–(c) and a vector-sensor ULA (d)–(f). Estimates are obtained with (a) DAS, (b) direct implementation of SPICE, and (c) fast implementation of SPICE and (d) DAS, (e) direct implementation of SPICE, and (f) fast implementation of SPICE. Both ULAs have $M = 20$ sensors and collect $N = 10M$ snapshots. The vertical dashed lines indicate the true DOAs.

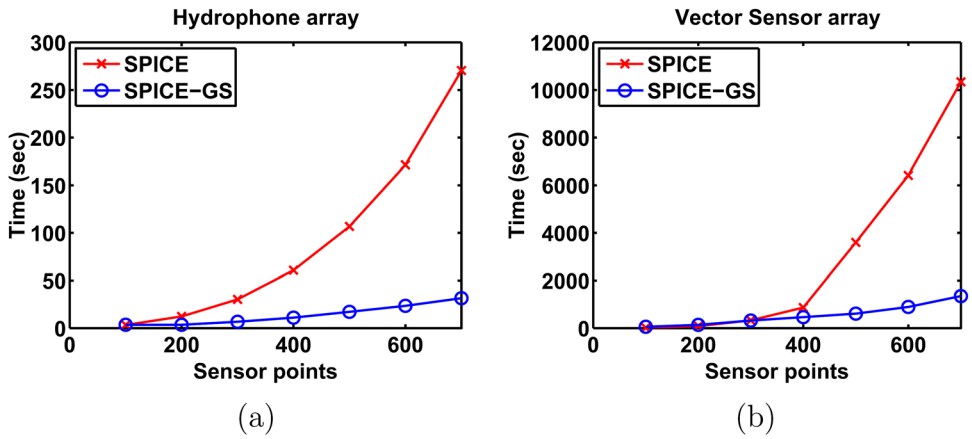


FIG. 2. (Color online) Computational time comparison of the direct implementation of SPICE and fast implementations (SPICE-GS) for (a) hydrophone ULA and (b) vector-sensor ULA against the number of sensors M in the array, with $N = 10M$ available snapshots and $K = 10M$ angular scanning points.

$$Q_{DM} = \begin{bmatrix} \sigma & 0 & 0 & \frac{\sin\kappa_{12}}{\kappa_{12}}\sigma & 0 & 0 & \cdots \\ 0 & \sigma/3 & 0 & 0 & Q_{\bar{u}^x_{r1},\bar{u}^x_{r2}} & 0 & \cdots \\ 0 & 0 & \sigma/3 & 0 & 0 & Q_{\bar{u}^y_{r1},\bar{u}^y_{r2}} & \cdots \\ \frac{\sin\kappa_{21}}{\kappa_{21}}\sigma & 0 & 0 & \sigma & 0 & 0 & \cdots \\ 0 & Q_{\bar{u}^x_{r1},\bar{u}^x_{r2}} & 0 & 0 & \sigma/3 & 0 & \cdots \\ 0 & 0 & Q_{\bar{u}^y_{r1},\bar{u}^y_{r2}} & 0 & 0 & \sigma/3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{60}$$

where $Q_{\bar{u}^x_{r1},\bar{u}^x_{r2}} = \frac{(\kappa_{12}^2-2)\sin\kappa_{12}+2\kappa_{12}\cos\kappa_{12}}{\kappa_{12}^3}\sigma$, $Q_{\bar{u}^y_{r1},\bar{u}^y_{r2}} = \frac{\sin\kappa_{12}-\kappa_{12}\cos\kappa_{12}}{\kappa_{12}^3}\sigma$, and $\kappa_{ij} = 2\pi d_{ij}/\lambda$, with d_{ij} denoting the distance between the ith and jth sensor, and λ denoting the wavelength.

In Fig. 1, we show a DOA estimation example using DAS (periodogram), direct implementation and fast implementation of the SPICE algorithm, with a 20-element hydrophone ULA and a 20-element vector-sensor ULA with half-wavelength intersensor spacing. Both arrays collect $N = 200$ snapshots and $K = 400$ scanning angular points are used. These angular scanning points are distributed uniformly across the spatial frequency f. A moderate signal-to-noise ratio (SNR) of 15 dB is assumed here and the true source locations are indicated by dashed vertical lines in Fig. 1.

In the hydrophone ULA scenario, the fast implementation in Fig. 1(c) offers identical results to the direct implementation in Fig. 1(b). Both implementations produced significantly enhanced angular resolution and attenuated sidelobe levels compared to DAS in Fig. 1(a). For instance, the second source at $\theta_2 = -36.9°$ is missed by the conventional DAS due to the method's severe power leakage and inadequate angular resolution, whereas both SPICE implementations were able to resolve the source. Note that ambiguity lobes exist in Figs. 1(a)–1(c), which is inevitable for 1-D hydrophone arrays.

In Figs. 1(d)–1(f), we show the DOA estimation results of the same sources as in Figs. 1(a)–1(c) using the vector-sensor ULA. Even with vector-sensor array measurements, the ambiguity lobes are not fully suppressed in Fig. 1(d), which implies that DAS is not effective for bearing ambiguity rejection. On the contrary, SPICE resolves all sources and eliminates the ambiguity lobes. Note that in spite of the fact that the covariance matrix in Eq. (60) is different from the approximate noise model in Eq. (21), the SPICE algorithm still provides superior resolution and sidelobe suppression, which also verifies its robustness in practical applications. Figure 1(f) again shows the fast implementation yields identical result as the direct implementation in Fig. 1(e).

Figure 2 compares the computational time required by the direct implementation and the fast implementation of SPICE on (a) conventional hydrophone ULA and (b) vector-sensor ULA for different number (M) of sensors. The simulations are implemented in MATLAB (The MathWorks, Inc., Natick, MA) using only one core of a workstation with an eight-core 2.83 GHz CPU and 8 GB RAM. Both the ULA and the vector-sensor ULA collect $N = 10M$ snapshots. The number of angular scanning points is $K = 10M$, the SNR is fixed to be 5 dB and 40 SPICE iterations[30] are performed in Figs. 2(a) and 2(b). For a large value of M, the fast implementations provide substantial time reduction compared with the direct implementations, which makes the fast implementations especially attractive for high resolution source localization applications. Note that the computational complexity reduction in the vector sensor ULA scenario is less dramatic than that in the hydrophone ULA scenario. This is due to the fact that the displacement rank of the covariance matrix $\tilde{\mathbf{R}}_{DM}^{-1}$ of the vector-sensor ULA is $2D$ instead of 2 for $\mathbf{R}_M^{-1}$ in the hydrophone ULA, which increases the computational costs for the generators of the inverse of SPICE covariance matrix and the covariance matrix-vector products.

VII. CONCLUSIONS

We have explored the DOA estimation problem using the SPICE algorithm with hydrophone ULAs and vector-sensor ULAs for the passive sonar applications. The SPICE algorithm is demonstrated in the simulations to possess advantageous properties, such as increased ability to resolve closely spaced sources and competence in rejection of bearing ambiguities in the vector-sensor array scenarios. Fast implementations for both hydrophones ULAs and vector-sensor ULAs are proposed: The computational complexity has been reduced by exploiting the Toeplitz/block Toeplitz matrix structure in the SPICE covariance matrix and by utilizing FFT/IFFT to compute spectral estimates. In the simulations, it has been shown that a significant computational efficiency increase is obtained with no loss in performance.

ACKNOWLEDGMENTS

This work was supported in part by the Office of Naval Research (ONR) under Grant No. N00014-10-1-0054, the National Science Foundation (NSF) under Grant No. ECCS-0729727, the National Aeronautics and Space Administration (NASA) under Grant No. NNX07AO15A, and the SMART Fellowship Program.

APPENDIX A: COMPUTING THE GENERATORS $\mathbf{w}_t$ AND α_t IN TABLE II

For $t = 2,\ldots, M-1$, plugging Eq. (15) into the definition of $\mathbf{w}_t$ yields:

$$\begin{aligned}\mathbf{w}_t &= -\mathbf{R}_t^{-1}\mathbf{r}_t \\ &= -\left\{\begin{bmatrix}\mathbf{R}_{t-1}^{-1} & 0\\ 0 & 0\end{bmatrix} + \frac{1}{\alpha_{t-1}}\begin{bmatrix}\breve{\mathbf{w}}_{t-1}^*\\ 1\end{bmatrix}\left[\breve{\mathbf{w}}_{t-1}^T, 1\right]\right\}\begin{bmatrix}\mathbf{r}_{t-1}\\ r_t^*\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{w}_{t-1}\\ 0\end{bmatrix} - \frac{\psi_{t-1}}{\alpha_{t-1}}\begin{bmatrix}\breve{\mathbf{w}}_{t-1}^*\\ 1\end{bmatrix}, \end{aligned} \tag{A1}$$

where $\psi_{t-1}\underline{\underline{\Delta}}\breve{\mathbf{w}}_{t-1}^T r_{t-1} + r_t^*$.

Similarly, plugging Eq. (A1) into the definition of α_t yields:

$$\begin{aligned}\alpha_t - \alpha_{t-1} &= \mathbf{r}_t\mathbf{w}_t - \mathbf{r}_{t-1}\mathbf{w}_{t-1} \\ &= \left[\mathbf{r}_{t-1}^H, r_{-t}\right]\left\{\begin{bmatrix}\mathbf{w}_{t-1}\\ 0\end{bmatrix} - \frac{1}{\alpha_{t-1}}\begin{bmatrix}\breve{\mathbf{w}}_{t-1}^*\\ 1\end{bmatrix}\psi_{t-1}\right\} \\ -\mathbf{r}_{t-1}\mathbf{w}_{t-1} &= -\frac{1}{\alpha_{t-1}}\left(\mathbf{r}_{t-1}^H\breve{\mathbf{w}}_{t-1}^* + r_{-t}\right)\psi_{t-1} \\ &= -\frac{1}{\alpha_{t-1}}\psi_{t-1}^H\psi_{t-1} = -|\psi_{t-1}|^2/\alpha_{t-1}. \end{aligned} \tag{A2}$$

Hence, the generators are computed iteratively as detailed in Table II.

APPENDIX B: PROOF OF LEMMA 1

1. Proof of Eq. (53)

Using Eq. (50), for $m = 2,\ldots, M-1$, we have

$$\bar{\mathcal{A}}_{Dm} = \mathbf{J}_{Dm}\bar{\mathcal{B}}_{Dm}^*\mathbf{J}_D. \tag{B1}$$

As $\mathbf{J}_D\mathbf{J}_D^T = \mathbf{I}_D$ the following equation holds:

$$\bar{\mathcal{A}}_{Dm}\bar{\mathcal{A}}_{Dm}^H = \mathbf{J}_{Dm}\bar{\mathcal{B}}_{Dm}\bar{\mathcal{B}}_{Dm}^T\mathbf{J}_{Dm}. \tag{B2}$$

From Eqs. (B2) and (42),

$$\tilde{\mathbf{R}}_{Dm}^{-1}\mathbf{J}_{Dm} = \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \tilde{\mathbf{R}}_{D(m-1)}^{-1}\mathbf{J}_{D(m-1)} & \mathbf{0}\end{bmatrix} + \mathbf{J}_{Dm}\bar{\mathcal{B}}_{Dm}^*\bar{\mathcal{B}}_{Dm}^T. \tag{B3}$$

From Eq. (46) and using the relation $\mathbf{J}_{Dm}\hat{\mathcal{R}}_{Dm}\mathbf{J}_D = \bar{\mathcal{R}}_{Dm}^*$,

$$\begin{aligned}\mathcal{B}_{Dm} &= -\tilde{\mathbf{R}}_{Dm}^{-1}\hat{\mathcal{R}}_{Dm} = -\tilde{\mathbf{R}}_{Dm}^{-1}\mathbf{J}_{Dm}(\mathbf{J}_{Dm}\hat{\mathcal{R}}_{Dm}\mathbf{J}_D)\mathbf{J}_D \\ &= -\tilde{\mathbf{R}}_{Dm}^{-1}\mathbf{J}_{Dm}\hat{\mathcal{R}}_{Dm}^*\mathbf{J}_D. \end{aligned} \tag{B4}$$

Substitution of Eqs. (B3) into (B4) and using the fact that $\bar{\mathcal{R}}_{Dm}^* = \begin{bmatrix}\bar{\mathcal{R}}_{D(m-1)}^H & \mathbf{R}_M^T\end{bmatrix}^T$,

$$\begin{aligned}\mathcal{B}_D &= \begin{bmatrix}\mathbf{0}\\ -\tilde{\mathbf{R}}_{D(m-1)}^{-1}\mathbf{J}_{D(m-1)}\bar{\mathcal{R}}_{D(m-1)}^*\mathbf{J}_D\end{bmatrix} \\ &\quad -\mathbf{J}_{Dm}\bar{\mathcal{B}}_{Dm}^*\bar{\mathcal{B}}_{Dm}^T\bar{\mathcal{R}}_{D(m-1)}^*\mathbf{J}_D, \\ &= \begin{bmatrix}\mathbf{0}\\ \mathcal{B}_{D(m-1)}\end{bmatrix} - \mathbf{J}_{Dm}\begin{bmatrix}\mathcal{B}_{D(m-1)}^*\\ \mathbf{I}_D\end{bmatrix}\Omega_m, \end{aligned} \tag{B5}$$

where $\mathbf{\Omega}_m = \mathbf{\Delta}_{m-1}^{-*}\left[\mathbf{R}_m + \mathcal{B}_{D(m-1)}^T\bar{\mathcal{R}}_{D(m-1)}^*\right]\mathbf{J}_D$. Hence Eq. (53) is proved.

2. Proof of Eq. (54)

From Eq. (48) and using Eq. (53),

$$\begin{aligned}\Delta_m &= \mathbf{R}_0 + \hat{\mathcal{R}}_{Dm}^H\mathcal{B}_{Dm} \\ &= \mathbf{R}_0 + \hat{\mathcal{R}}_{Dm}^H\begin{bmatrix}\mathbf{0}\\ \mathcal{B}_{D(m-1)}\end{bmatrix} - \hat{\mathcal{R}}_{Dm}^H\mathbf{J}_{Dm}\begin{bmatrix}\mathcal{B}_{D(\Delta ABCm-1)}^*\\ \mathbf{I}_D\end{bmatrix}\Omega_m \\ &= \mathbf{R}_0 + \hat{\mathcal{R}}_{D(m-1)}^H\mathcal{B}_{D(m-1)} - \mathbf{J}_D(\mathbf{R}_m^* + \bar{\mathcal{R}}_{D(m-1)}^T\mathcal{B}_{D(m-1)}^*)\Omega_m, \end{aligned} \tag{B6}$$

where the last equality uses the relations that $\hat{\mathcal{R}}_{Dm}^H\mathbf{J}_{Dm} = \mathbf{J}_D\hat{\mathcal{R}}_{Dm}^T$ and $\bar{\mathcal{R}}_{Dm} = \left[\bar{\mathcal{R}}_{D(m-1)}^T\mathbf{R}_m^H\right]^T$. Consequently,

$$\Delta_m = \Delta_{m-1} - \Omega_m^H\Delta_{m-1}^T\Omega_m. \tag{B7}$$

APPENDIX C: BLOCK TOEPLITZ MATRIX-VECTOR PRODUCT WITH $D \times 1$ BLOCKS

This part focuses on computing a matrix-vector product $\mathbf{Tz}$, where $\mathbf{T}$ is a $DM \times M$ block Toeplitz matrix with each block being a $D \times 1$ column vector and $\mathbf{z}$ is an arbitrary $M \times 1$ vector, $\mathbf{z} = [z_0, z_1,\ldots, z_{m-1}]^T$,

$$\mathbf{T}\underline{\underline{\Delta}}\begin{bmatrix}\mathbf{t}_0 & & & \\ \mathbf{t}_1 & \mathbf{t}_0 & & \\ \vdots & \vdots & \ddots & \\ \mathbf{t}_{M-1} & \mathbf{t}_{M-2} & \cdots & \mathbf{t}_0\end{bmatrix},$$

where $t_i \triangleq \left[t_{i,0}, t_{i,1}, \ldots, t_{i,D-1}\right]^T$, $i = 0\ldots D-1$.

Let $[\mathbf{r}_0, \mathbf{r}_1,., \mathbf{r}_{M-1}]^T$ denote the product:

$$\begin{bmatrix}\mathbf{r}_0\\ \mathbf{r}_1\\ \vdots\\ \mathbf{r}_{M-1}\end{bmatrix} = \begin{bmatrix}\mathbf{t}_0 & & & \\ \mathbf{t}_1 & \mathbf{t}_0 & & \\ \vdots & \vdots & \ddots & \\ \mathbf{t}_{M-1} & \mathbf{t}_{M-1} & \ldots & \mathbf{t}_0\end{bmatrix}\begin{bmatrix}z_0\\ z_1\\ \vdots\\ z_{M-1}\end{bmatrix}, \tag{C1}$$

where $r_i \triangleq \left[r_{i,0}, r_{i,1}, \ldots, r_{i,D-1}\right]^T$, $i = 0,.,M-1$.

The multiplication in Eq. (C1) can be decomposed into a series of Toeplitz matrix-vector products, which can be computed efficiently using FFT.

Select the jth element in each of the column vectors $\mathbf{r}_0,\ldots, \mathbf{r}_{M-1}$, and the jth element in each of the $\mathbf{t}_0,\ldots,\mathbf{t}_{M-1}$, $j = 0,\ldots, D-1$. From Eq. (C1),

$$\begin{bmatrix}r_{0,j}\\ r_{1,j}\\ \vdots\\ r_{M-1,j}\end{bmatrix} = \begin{bmatrix}t_{0,j} & & & \\ t_{1,j} & t_{0,j} & & \\ \vdots & \vdots & \ddots & \\ t_{M-1,j} & t_{M-2,j} & \ldots & t_{0,j}\end{bmatrix}\begin{bmatrix}z_0\\ z_1\\ \vdots\\ z_{M-1}\end{bmatrix} \tag{C2}$$

The right-hand side of Eq. (C2) is a lower triangular Toeplitz matrix-vector product, which can be computed efficiently by FFT (detailed in Appendix E).

APPENDIX D: BLOCK TOEPLITZ MATRIX-VECTOR PRODUCT WITH $1 \times D$ BLOCKS

Consider the matrix-vector product $\mathbf{S}^H\mathbf{y}$, where $\mathbf{S}^H$ is an $M \times DM$ block Toeplitz matrix with each block being a $1 \times D$ row vector and $\mathbf{y}$ is an arbitrary $DM \times 1$ column vector. Partition $\mathbf{y}$ into M D-dimensional column vectors: $\mathbf{y} \triangleq \left[\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{M-1}\right]^T$, where $\mathbf{y}_i \triangleq \left[y_{i,0}, y_{i,1}, \ldots, y_{i,D-1}\right]^T$, $i = 0,\ldots,M-1$.

The corresponding block Toeplitz matrix $\mathbf{S}^H$ is similarly partitioned as

$$\mathbf{S}^H \triangleq \begin{bmatrix} \mathbf{s}_0^H & \mathbf{s}_1^H & \cdots & \mathbf{s}_{M-1}^H \\ & \mathbf{s}_0^H & \cdots & \mathbf{s}_{M-2}^H \\ & & \ddots & \vdots \\ & & & \mathbf{s}_0^H \end{bmatrix}, \tag{D1}$$

where $\mathbf{s}_i^H$, $i = 0\ldots(M-1)$, is a row vector, $\mathbf{s}_i^H \triangleq \left[s_{i,0}, s_{i,j}, \ldots, s_{j,D-1}\right]$. Let $[z_0,., z_{M-1}]^T$ denote the matrix-vector product:

$$\begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{M-1} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_0^H & \mathbf{s}_1^H & \cdots & \mathbf{s}_{M-1}^H \\ & \mathbf{s}_0^H & \cdots & \mathbf{s}_{M-2}^H \\ & & \ddots & \vdots \\ & & & \mathbf{s}_0^H \end{bmatrix} \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{M-1} \end{bmatrix} \tag{D2}$$

Note that Eq. (D2) can be decomposed into a series of Toeplitz matrix-vector products: Select the jth element in each of the $\mathbf{s}_0,\ldots,\mathbf{s}_{M-1}$ and $\mathbf{y}_0,\ldots,\mathbf{y}_{M-1}$, and define $\left[z_{0,j}, z_{1,j}, \ldots, z_{M-1,j}\right]^T$, $j = 0, \ldots, D-1$, as

$$\begin{bmatrix} z_{0,j} \\ z_{1,j} \\ \vdots \\ z_{M-1,j} \end{bmatrix} = \begin{bmatrix} s_{0,j} & s_{1,j} & \cdots & s_{M-1,j} \\ & s_{0,j} & \cdots & s_{M-2,j} \\ & & \ddots & \vdots \\ & & & s_{0,j} \end{bmatrix} \begin{bmatrix} y_{0,j} \\ y_{1,j} \\ \vdots \\ y_{M-1,j} \end{bmatrix}. \tag{D3}$$

Note that the right-hand side of Eq. (D3) can be efficiently computed using the FFT (detailed in Appendix E). The results in Eq. (D2) can be obtained by a series of summations: $z_k = \sum_{j=0}^{D-1} z_{k,j}$, $k = 0,\ldots,M-1$.

APPENDIX E: TOEPLITZ MATRIX-VECTOR PRODUCT

A circulant matrix $\mathbf{C}$ is a special type of Toeplitz matrix and it can be diagonalized by a discrete Fourier transform (DFT) matrix $\mathbf{F}_n$, i.e., $\mathbf{C} = \mathbf{F}_n^*\mathrm{Diag}(\mathbf{F}_n\mathbf{c})\mathbf{F}_n$. where $\mathbf{c}$ is the first column of the circulant matrix, and $\mathbf{F}_n$ is the n-point DFT matrix. An important property of a circulant matrix is that the circulant matrix-vector product can be computed using FFT and IFFT. For example, the product $\mathbf{y} = \mathbf{C}\mathbf{x}$ can be computed as $y = \mathbf{F}_n^*((\mathbf{F}_n\mathbf{c}) \odot (\mathbf{F}_n\mathbf{x}))$.

Consequently, the Toeplitz matrix-vector product can be computed by first augmenting the Toeplitz matrix into a circulant matrix, then performing the FFT based multiplication and retaining only the relevant part.

[1] P. Stoica and R. L. Moses, *Spectral Analysis of Signals* (Prentice-Hall, Upper Saddle River, NJ, 2005), pp. 1–16.

[2] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*, Vol. 1194 (Wiley, New York, 2002), pp. 599–603.

[3] P. Stoica, P. Babu, and J. Li, "New method of sparse parameter estimation in separable models and its use for spectral analysis of irregularly sampled data," IEEE Trans. Signal Process. **59**, 35–47 (2011).

[4] P. Stoica, P. Babu, and J. Li, "SPICE: A sparse covariance-based estimation method for array processing," IEEE Trans. Signal Process. **59**, 629–638 (2011).

[5] B. Musicus, "Fast MLM power spectrum estimation from uniformly spaced correlations," IEEE Trans. Acoustics, Speech Signal Process. **33**, 1333–1335 (1985).

[6] Z.-S. Liu, H. Li, and J. Li, "Efficient implementation of Capon and APES for spectral estimation," IEEE Trans. Aerosp. Electron. Syst. **34**, 1314–1319 (1998).

[7] T. Ekman, A. Jakobsson, and P. Stoica, "On the efficient implementation of the Capon spectral estimator," Proceedings of the European Signal Processing Conference, Tampere, Finland, 2000.

[8] A. Jakobsson, J. S. L. Marple, and P. Stoica, "Computationally efficient two-dimensional Capon spectrum analysis," IEEE Trans. Signal Process. **48**, 2651–2661 (2000).

[9] G.-O. Glentis, "A fast algorithm for APES and Capon spectral estimation," IEEE Trans. Signal Process. **56**, 4207–4220 (2008).

[10] P. Stoica and R. L. Moses, *Spectral Analysis of Signals* (Prentice-Hall, Upper Saddle River, NJ, 2005), pp. 99–102.

[11] J. O. B. Wilson, S. N. Wolf, and F. Ingenito, "Measurements of acoustic ambient noise in shallow water due to breaking surf," J. Acoust. Soc. Am. **78**, 190–195 (1985).

[12] J. Charles R. Greene, M. W. McLennan, R. G. Norman, T. L. McDonald, R. S. Jakubczak, and W. J. Richardson, "Directional frequency and recording (DIFAR) sensors in seafloor recorders to locate calling bowhead whales during their fall migration," J. Acoust. Soc. Am. **116**, 799–813 (2004).

[13] M. Hawkes and A. Nehorai, "Acoustic vector-sensor beamforming and Capon direction estimation," IEEE Trans. Signal Process. **46**, 2291–2304 (1998).

[14] H. Akaike, "Block Toeplitz matrix inversion," SIAM J. Appl. Math. **24**, 234–241 (1973).

[15] Criteria such as $||\mathbf{p}_K^{(i)} - \mathbf{p}_K^{(i-1)}||/||\mathbf{p}_K^{(i-1)}|| < 10^{-4}$ can be applied.

[16] I. C. Gohberg and A. Semencul, "On inversion of finite-section Toeplitz matrices and their continuous analogues," Mat. Issled. Kishinev **2**, 201–233 (1972) (In Russian).

[17] T. Huckle, "Computation of Gohberg–Semencul formulas for a Toeplitz matrix," Technical Report, Universität Würzburg and Stanford University (1993).

[18] D. A. Harville, *Matrix Algebra from a Statistician's Perspective* (Springer, New York, 1997), pp. 88–103.

[19] M. Xue, L. Xu, and J. Li, "IAA spectral estimation: fast implementation using the Gohberg–Semencul factorization," IEEE Trans. Signal Process. **59**(7), 3251–3261 (2011).

[20] In MATLAB: Z_M is generated by "diag (ones(M, 1), – 1)".

[21] J. Jain, "An efficient algorithm for a large Toeplitz set of linear equations," IEEE Trans. Acoust. Speech Signal Process. **27**, 612–615 (1979).

[22] A. Nehorai and E. Paldi, "Acoustic vector-sensor array processing," IEEE Trans. Signal Process. **42**, 2481–2491 (1994).

[23] A. J. Poulsen, "Robust vector sensor array processing and performance analysis," Ph.D. dissertation, Massachusetts Institute of Technology, Boston, MA (2009), pp. 25–38.

[24] H. Cox, "Spatial correlation in arbitrary noise fields with application to ambient sea noise," J. Acoust. Soc. Am. **54**, 1289–1301 (1973).

[25] E. J. Sullivan, "A generalized Cramer–Rao lower bound for moving arrays," J. Acoust. Soc. Am. **125**, EL51–EL57 (2009).

[26] N. K. Naluai, "Acoustic intensity methods and their applications to vector sensor use and design," Ph.D. dissertation, The Pennsylvania State University, State College, PA (2006), pp. 61–93.

[27] Which retains the block Toeplitz matrix structure in $\tilde{\mathbf{R}}_{DM}$ and makes the fast implementation possible.

[28] M. Wax and T. Kailath, "Efficient inversion of Toeplitz-block Toeplitz matrix," IEEE Trans. Acoust. Speech Signal Process. **31**, 1218–1221 (1983).

[29] S. Zohar, "Toeplitz matrix inversion: The algorithm of W. F. Trench," J. Assoc. Comput. Mach. **16**, 592–601 (1969).

[30] Based on our empirical experience that the algorithm does not offer significant performance enhancement after 40 iterations in our examples.