Fast Implementation of Sparse Iterative Covariance-based Estimation for Array Processing

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

***Abstract*—Fast implementations of the SParse Iterative Covariance-based Estimation (SPICE) algorithm are presented for source localization in passive sonar applications. 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 at the cost of a higher computational complexity compared to the conventional delay-and-sum beamforming method. It is shown in this paper that the computational complexity of the SPICE algorithm can be reduced by exploiting the Toeplitz structure of the array output covariance matrix using the Gohberg-Semencul factorization. The fast implementations for both the hydrophone uniform linear array (ULA) and the vector-sensor ULA scenarios are proposed and the computational gains are illustrated by numerical simulations.**

***Index Terms*—Sparse Iterative Covariance-based Estimation, direction-of-arrival, fast implementation, Gohberg-Semencul factorization, vector-sensor.**

I. INTRODUCTION

The direction-of-arrival (DOA) source localization problem is directly applicable to many fields including radar, sonar, communications, 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 hydrophones. The traditional delay-and-sum (DAS) beamformer suffers in performance due to high sidelobes, low resolution. Many high-resolution methods such as Capon and MUltiple SIgnal Classification (MUSIC), are typically sensitive to modeling errors, require a large number of snapshots, and can even fail if coherent sources are present [1].

Recently, a new semi-parametric algorithm referred to as SParse Iterative Covariance-based Estimation (SPICE), has been proposed [2], [3]. SPICE offers superior resolution and low sidelobe levels while retaining robustness against correlated sources [3]. It is also a user parameter-free method that is guaranteed to converge globally. SPICE suffers from higher computational complexity when compared with the DAS method.

In this paper, an efficient implementation of SPICE based on the Gohberg-Semencul (G-S) factorization is presented, which is inspired by previous fast implementations on Capon and Amplitude and Phase EStimation (APES) methods [4], [5], [6], [7], [8]. The Toeplitz matrix structure produced by the G-S factorization of the inverse of the SPICE covariance matrix $\mathbf{R}_M^{-1}$ is exploited while applying a Levinson-Durbin type algorithm (see e.g., [9]).

A typical vector-sensor array measures both acoustic particle velocities and acoustic pressure. It is capable of resolving bearing ambiguity of the linear hydrophone arrays and simultaneously estimating azimuth and elevation angles of signals of interest (SOIs) at the cost of a higher dimensionality (i.e., higher computational complexity, [10], [11]). The fast implementation of SPICE is also extended to the vector-sensor array scenario by exploiting the block Toeplitz structure (e.g., [12]) of the array output covariance matrix using G-S type factorization based on a generalized Levinson-Durbin algorithm (LDA) and fast Fourier transform (FFT).

Notation: Vectors and matrices are denoted by boldface lowercase and boldface uppercase letters, respectively, $\|\cdot\|_F$ denotes the Frobenius norm, $\otimes$ denotes the Kronecker matrix product, $\mathrm{Diag}(p_0,\ldots,p_{K-1})$ denotes a diagonal matrix with $p_0,\ldots,p_{K-1}$ as its diagonal elements. $(\cdot)^T$, $(\cdot)^*$ and $(\cdot)^H$ denote the transpose, complex conjugate and conjugate transpose of a vector or matrix, respectively. $\mathbf{I}_p$ and $\mathbf{J}_p$ denote identity matrix and exchange matrix of dimension $p \times p$, respectively.

II. DATA MODEL AND PROBLEM FORMULATION

Let a uniform linear array (ULA) of M omnidirectional sensors with half-wavelength inter-element spacing receive narrowband signals impinging from the sources with unknown locations. Let Ω denote the set of possible locations, and θ be a generic location parameter. 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., [2])

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

where $\mathbf{A}_M \triangleq [\mathbf{a}_M(\theta_0),\ldots,\mathbf{a}_M(\theta_{K-1})]$ is the steering matrix where each steering vector $\mathbf{a}_M(\theta_k) \triangleq [1, e^{j\pi\sin(\theta_k)},\ldots,e^{j\pi(M-1)\sin(\theta_k)}]^T$ is parameterized by the location parameter θ_k. The vector $\mathbf{x}(n) \triangleq [\mathbf{x}_0(n),\ldots,\mathbf{x}_{K-1}(n)]^T$ contains the K unknown complex signals, and $\mathbf{e}(n)$ is the noise term. We assume that $E\left[\mathbf{e}(n)\mathbf{e}^H(\bar{n})\right] = \sigma \mathbf{I}_M \delta_{n,\bar{n}}$, 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 independent, thus $E\left[\mathbf{x}(n)\mathbf{x}^H(\bar{n})\right] = \mathbf{P}_K \delta_{n,\bar{n}}$, where $\mathbf{P}_K \triangleq \mathrm{Diag}(p_0,\ldots,p_{K-1})$ and p_k denotes the unknown signal power at θ_k. This leads to the covariance matrix of $\mathbf{y}_M(n)$ (e.g., [2], [3])

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

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, and the SMART Fellowship Program.

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 $\mathbf{Y} \triangleq [\mathbf{y}_M(1), \ldots, \mathbf{y}_M(N)]$.

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 [2], [3]. In the particular case of spatially and temporally white uniform noise (i.e., $E\left[\mathbf{e}(n)\mathbf{e}^H(n)\right] = \sigma\mathbf{I}_{\mathrm{M}}$), the iterative steps of the SPICE method (SPICE+ as named in [3]) are summarized as follows:

Initialize $\{p_k^{(0)}\}_{k=0}^{K-1}$ using DAS. At the ith iteration,

1) Update $\mathbf{R}_M^{(i)}$ using 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 $\mathbf{P}_K^{(i-1)} \triangleq \mathrm{Diag}(p_0^{(i-1)}, \ldots, p_{K-1}^{(i-1)})$.

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

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

3) Estimate the noise power:

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

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

$$p_k^{(i)} = p_k^{(i-1)}\frac{\parallel \mathbf{a}_M^H(\theta_k){\mathbf{R}_M^{(i)}}^{-1}\hat{\mathbf{R}}_M^{1/2} \parallel_F}{\omega_k^{1/2}\rho^{(i)}}, \tag{6}$$

for $k = 0, 1, \ldots, K-1$. The quantities $\{\omega_k\}_{k=0}^{K+M-1}$ are constants over all iterations, and they are given by:

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

where $\hat{\mathbf{R}}_M^{-1}(k,k)$ denotes the element at kth row and kth column of the matrix $\hat{\mathbf{R}}_M^{-1}$. γ is given by $\gamma \triangleq \sum_{k=K}^{K+M-1}\omega_k$.

The terms $\parallel {\mathbf{R}_M^{(i)}}^{-1}\hat{\mathbf{R}}_M^{1/2} \parallel_F$ and $\parallel \mathbf{a}_M^H(\theta_k){\mathbf{R}_M^{(i)}}^{-1}\hat{\mathbf{R}}_M^{1/2} \parallel_F$ listed in the above iterative steps ((4) - (6)) have simpler forms:

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

and similarly,

$$\parallel \mathbf{a}_M^H(\theta_k){\mathbf{R}_M^{(i)}}^{-1}\hat{\mathbf{R}}_M^{1/2} \parallel_F^2 = \frac{1}{N}\sum_{n=1}^{N}|\mathbf{a}_M^H(\theta_k){\mathbf{R}_M^{(i)}}^{-1}\mathbf{y}_M(n)|^2. \tag{8}$$

IV. Fast implementation for Uniform Linear Arrays of hydrophones

Based on the G-S factorization (e.g., [13]), the inverse of 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 (7) and (8). Moreover, if the spatial frequency f ($f \triangleq \sin\theta$) 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 $\underline{\mathbf{R}}_M \triangleq \mathbf{A}_M\mathbf{P}_K\mathbf{A}_M^H$. Since 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:

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

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

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

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 (3)).

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

Once the first column of $\mathbf{R}_M$ is available, the vector $\mathbf{d}_M(n)$ involved in the linear equation

$$\mathbf{R}_M(n)\mathbf{d}_M(n) = \mathbf{y}_M(n), \tag{11}$$

appeared in (7) and (8) can be solved using the G-S factorization and the LDA.

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

The G-S formula (see e.g., [4], [14]) of $\mathbf{R}_M^{-1}$ is given by:

$$\mathbf{R}_M^{-1} = \mathcal{L}_M(\mathbf{t}_1, \mathbf{Z}_M)\mathcal{L}_M^H(\mathbf{t}_1, \mathbf{Z}_M) - \mathcal{L}_M(\mathbf{t}_2, \mathbf{Z}_M)\mathcal{L}_M^H(\mathbf{t}_2, \mathbf{Z}_M), \tag{14}$$

where $\mathbf{Z}_M$[1]is an $M \times M$ matrix with ones on the first sub-diagonal and zeros everywhere else, $\mathbf{t}_1 =$

[1]In MATLAB: $\mathbf{Z}_M$ is generated by diag (ones$(M,1), -1)$.

TABLE I
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 $t = 2, \ldots, M-1$
	$\psi_{t-1} = \mathbf{w}_{t-1}^T \mathbf{J}_{t-1} \mathbf{r}_{t-1} + r_t^*$
	$\mathbf{w}_t = \begin{bmatrix} \mathbf{w}_{t-1} \\ 0 \end{bmatrix} - \frac{1}{\alpha_{t-1}} \begin{bmatrix} \mathbf{J}_{t-1}\mathbf{w}_{t-1}^* \\ 1 \end{bmatrix} \psi_{t-1}$
	$\alpha_t = \alpha_{t-1} - \lvert\psi_{t-1}\rvert^2 / \alpha_{t-1}$

$\frac{1}{\sqrt{\alpha_{M-1}}}\left[1,\ \mathbf{w}_{M-1}\right]^T$, $\mathbf{t}_2 = \frac{1}{\sqrt{\alpha_{M-1}}}\left[0,\ \mathbf{J}_{M-1}\mathbf{w}_{M-1}^*\right]^T$, and $\mathcal{L}_M(\mathbf{t}, \mathbf{Z}) \triangleq \left[\mathbf{t}, \mathbf{Z}\mathbf{t}, \cdots, \mathbf{Z}^{M-1}\mathbf{t}\right]$ denotes a Krylov matrix. The recursive LDA algorithm to compute the generators $\mathbf{w}_{M-1}$ and α_{M-1} is summarized by Table I and proved in detail in [15].

Note that in (14) 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 (11) using FFT ([15]).

The fast implementation of SPICE for hydrophone ULAs is summarized as follows:

For each iteration of SPICE, do the following:

1) Given $\{p_k^{i-1}\}_{k=0}^{K-1}$ from the previous iteration, compute $\{r_m\}_{m=0}^{M-1}$ in (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 I with $\mathcal{O}(M^2)$ flops.
3) Given the generators, calculate $\{\mathbf{R}_M^{(i)^{-1}}\mathbf{y}_M(n)\}_{n=1}^N$ in (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 (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 $\|\ \mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\ \|_F$ in (7) and $\|\ \mathbf{a}_M^H(\theta_k)\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\ \|_F$ in (8) with $\mathcal{O}(MN)$ and $\mathcal{O}(NK)$ flops, respectively.
6) Given $\|\ \mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\ \|_F$ and $\|\ \mathbf{a}_M^H(\theta_k)\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\ \|_F$, calculate $\rho^{(i)}$ in $\mathcal{O}(K)$ flops.
7) Given $\|\ \mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\ \|_F$, calculate $\sigma^{(i)}$ in $\mathcal{O}(1)$ flops.
8) Given $\|\ \mathbf{a}_M^H(\theta_k)\mathbf{R}_M^{(i)^{-1}}\hat{\mathbf{R}}_M^{1/2}\ \|_F$, update $\{p_k^{(i)}\}_{k=0}^{K-1}$ in $\mathcal{O}(K)$ flops.

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

A. Computation of the Block Toeplitz Covariance Matrix in Vector-sensor ULA

A vector-sensor array provides extra directional motion information such as acoustic particle velocity or acceleration in addition to the acoustic pressure measurements. A typical measurement from a single vector-sensor consists of the acoustic pressure and the orthogonal directional velocities; and it can be denoted as a scalar-vector product $p \cdot \mathbf{h}$ (see, e.g., [10], [11], [16])), where

$$\mathbf{h} \triangleq \left[1,\ \cos\phi\cos\theta,\ \cos\phi\sin\theta,\ \sin\phi\right]^T, \tag{15}$$

and ϕ, θ denote the elevation and azimuth angles, respectively. In the general case, denote $\mathbf{h}$ in (15) as,

$$\mathbf{h}_D(\theta_i) \triangleq \left[1,\ h_2(\theta_i),\ \ldots\ h_D(\theta_i)\right]^T. \tag{16}$$

where $h_d(\theta_i)$ denotes the dth element in $\mathbf{h}_D(\theta_i)$, $d = 2, \ldots, D$, and θ_i denotes a generic bearing parameter (the vector (θ, ϕ) in the special case of (15)). The steering vector ($\mathbf{a}_{DM}$) of a vector-sensor array is related to the steering vector ($\mathbf{a}_M$) of the hydrophone array (which consists of only the hydrophones within the vector-sensor array) as:

$$\mathbf{a}_{DM}(\theta_i) = \mathbf{a}_M(\theta_i) \otimes \mathbf{h}_D(\theta_i). \tag{17}$$

Consider an M-element vector-sensor array, where each element consists of one hydrophone and $(D-1)$ acoustic velocity sensors. Each measurement from this array forms a $DM \times 1$ column vector $\mathbf{y}_{DM}$, and the array output covariance matrix is a $DM \times DM$ matrix $\tilde{\mathbf{R}}_{DM}$. Analogous to the hydrophone scenario, this covariance matrix is estimated by the sample covariance matrix $\hat{\mathbf{R}}_{DM}$. In the SPICE algorithm, the covariance matrix $\tilde{\mathbf{R}}_{DM}$ is updated during each iteration whenever new signal power estimates $\{p_k\}_{k=0}^{K-1}$ are generated,

$$\tilde{\mathbf{R}}_{DM} = \sum_{k=0}^{K-1} p_k \mathbf{a}_{DM}(\theta_k)\mathbf{a}_{DM}^H(\theta_k) + \sigma\mathbf{I}_{DM} \tag{18}$$

$$= \begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_1 & \mathbf{R}_2 & \ldots & \mathbf{R}_{M-1} \\ \mathbf{R}_1^* & \mathbf{R}_0 & \mathbf{R}_1 & \ldots & \mathbf{R}_{M-2} \\ \mathbf{R}_2^* & \mathbf{R}_1^* & \mathbf{R}_0 & \ldots & \ldots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{M-1}^* & \mathbf{R}_{M-2}^* & \ldots & \ldots & \mathbf{R}_0 \end{bmatrix} + \sigma\mathbf{I}_{DM}, \tag{19}$$

Analogous to (10), for $m = 0, \ldots, M-1$,

$$\mathbf{R}_m = \sum_{k=0}^{K-1} p_k \Big(\mathbf{h}_D(\theta_k)\mathbf{h}_D^T(\theta_k)\Big) e^{-j2\pi mk/K}, \tag{20}$$

From (16) and (20), each block $\mathbf{R}_m$ is a symmetric matrix, and it is possible to calculate its elements using FFT.

B. 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 $\bigtriangledown_{\mathbf{Z}_M, \mathbf{Z}_M^T}\tilde{\mathbf{R}}_{DM}^{-1}$.

According to the Toeplitz block structure of $\tilde{\mathbf{R}}_{DM}$ in (19), it can be expressed as

$$\tilde{\mathbf{R}}_{DM} = \begin{bmatrix} \mathbf{R}_0 & \bar{\mathcal{R}}_{D(M-1)}^H \\ \bar{\mathcal{R}}_{D(M-1)} & \tilde{\mathbf{R}}_{D(M-1)} \end{bmatrix} \tag{21}$$

$$= \begin{bmatrix} \tilde{\mathbf{R}}_{D(M-1)} & \hat{\mathcal{R}}_{D(M-1)} \\ \hat{\mathcal{R}}_{D(M-1)}^H & \mathbf{R}_0 \end{bmatrix}, \tag{22}$$

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

Application of the matrix inversion lemma (e.g., [17]) for partitioned matrices to (21) and (22) yields, respectively:

TABLE II
L-D-TYPE ALGORITHM FOR GENERATORS $\mathcal{B}_{Dm}$ AND $\mathbf{\Delta}_m$.

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

$$\tilde{\mathbf{R}}_{DM}^{-1} = \begin{bmatrix} \tilde{\mathbf{R}}_{D(M-1)}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \bar{\mathcal{B}}_{DM}\bar{\mathcal{B}}_{DM}^H \tag{23}$$

$$= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{R}}_{D(M-1)}^{-1} \end{bmatrix} + \bar{\mathcal{A}}_{DM}\bar{\mathcal{A}}_{DM}^H, \tag{24}$$

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

$$\bar{\mathcal{A}}_{DM} \triangleq \begin{bmatrix} \mathbf{I}_M \\ \mathcal{A}_{D(M-1)} \end{bmatrix}\mathbf{Q}_{M-1}^{-1/2}, \tag{25}$$

$$\bar{\mathcal{B}}_{DM} \triangleq \begin{bmatrix} \mathbf{I}_M \\ \mathcal{B}_{D(M-1)} \end{bmatrix}\mathbf{\Delta}_{M-1}^{-1/2}, \tag{26}$$

respectively, where

$$\mathcal{A}_{D(M-1)} = -\tilde{\mathbf{R}}_{D(M-1)}^{-1}\bar{\mathcal{R}}_{D(M-1)}, \tag{27}$$

$$\mathcal{B}_{D(M-1)} = -\tilde{\mathbf{R}}_{D(M-1)}^{-1}\hat{\mathcal{R}}_{D(M-1)}, \tag{28}$$

$$\mathbf{Q}_{M-1} = \mathbf{R}_0 + \bar{\mathcal{R}}_{D(M-1)}^H\mathcal{A}_{D(M-1)}, \tag{29}$$

$$\mathbf{\Delta}_{M-1} = \mathbf{R}_0 + \hat{\mathcal{R}}_{D(M-1)}^H\mathcal{B}_{D(M-1)}. \tag{30}$$

By the persymmetry property of $\tilde{\mathbf{R}}_{DM}$ (e.g., [18], [19]),

$$\mathbf{J}_{Dm}\tilde{\mathbf{R}}_{Dm}\mathbf{J}_{Dm} = \tilde{\mathbf{R}}_{Dm}^T,\ m = 1, \ldots, M, \tag{31}$$

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

$$\mathcal{A}_{D(M-1)} = \mathbf{J}_{D(M-1)}\mathcal{B}_{D(M-1)}^*\mathbf{J}_D, \tag{32}$$

$$\mathbf{Q}_{M-1} = \mathbf{J}_D\mathbf{\Delta}_{M-1}^T\mathbf{J}_D. \tag{33}$$

By using (23)–(33), the displacement representation (DR) of $\tilde{\mathbf{R}}_{DM}^{-1}$ takes the following form:

$$\nabla_{\mathbf{Z}_M\otimes\mathbf{I}_D,\mathbf{Z}_M^T\otimes\mathbf{I}_D}\tilde{\mathbf{R}}_{DM}^{-1} = \bar{\mathcal{A}}_{DM}\bar{\mathcal{A}}_{DM}^H - (\mathbf{Z}_M\otimes\mathbf{I}_D)\bar{\mathcal{B}}_{DM}\bar{\mathcal{B}}_{DM}^H(\mathbf{Z}_M^T\otimes\mathbf{I}_D). \tag{34}$$

Note that the generators $\mathcal{B}_{D(M-1)}$ and $\mathbf{\Delta}_{M-1}$ of $\tilde{\mathbf{R}}_{DM}^{-1}$ involved in (34) are all block matrices related to the forward predictor (23)–(33). These generators can be computed iteratively by using the L-D-type algorithm listed in Table II ([15]).

Once the DR of the matrix $\tilde{\mathbf{R}}_{DM}^{-1}$ is obtained, the G-S 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 $(\mathbf{Z}_M\otimes\mathbf{I}_D)\bar{\mathcal{B}}_{DM}$. Columns and rows in (34) can be expressed explicitly as follows:

$$\nabla_{\mathbf{Z}_M\otimes\mathbf{I}_D,\mathbf{Z}_M^T\otimes\mathbf{I}_D}\tilde{\mathbf{R}}_{DM}^{-1} \triangleq \sum_{i=1}^{2D}\sigma_i\mathbf{t}_{DM}^i\mathbf{t}_{DM}^{iH}, \tag{35}$$

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

Given (35), the G-S factorization of $\tilde{\mathbf{R}}_{DM}^{-1}$ takes the following form:

$$\begin{aligned}\tilde{\mathbf{R}}_{DM}^{-1} &= \sum_{j=0}^{M-1}(\mathbf{Z}_M\otimes\mathbf{I}_D)^j(\nabla_{\mathbf{Z}_M\otimes\mathbf{I}_D,\mathbf{Z}_M^T\otimes\mathbf{I}_D}\tilde{\mathbf{R}}_{DM}^{-1})(\mathbf{Z}_M^T\otimes\mathbf{I}_D)^j \\ &= \sum_{i=1}^{2D}\sigma_i\Big[\sum_{j=0}^{M-1}(\mathbf{Z}_M\otimes\mathbf{I}_D)^j(\mathbf{t}_{DM}^i\mathbf{t}_{DM}^{iH})(\mathbf{Z}_M^T\otimes\mathbf{I}_D)^j\Big] \\ &= \sum_{i=1}^{2D}\sigma_i\mathcal{L}_{DM}(\mathbf{t}_{DM}^i,\mathbf{Z}_M\otimes\mathbf{I}_D)\mathcal{L}_{DM}^H(\mathbf{t}_{DM}^i,\mathbf{Z}_M\otimes\mathbf{I}_D),\end{aligned} \tag{36}$$

where

$$\begin{aligned}&\mathcal{L}_{DM}(\mathbf{t}_{DM}^i,\mathbf{Z}_M\otimes\mathbf{I}_D) \\ &= \Big[\mathbf{t}_{DM}^i,\ (\mathbf{Z}_M\otimes\mathbf{I}_D)\mathbf{t}_{DM}^i,\ \cdots,\ (\mathbf{Z}_M\otimes\mathbf{I}_D)^{M-1}\mathbf{t}_{DM}^i\Big].\end{aligned} \tag{37}$$

Due to the fact that $\mathcal{L}_{DM}(\mathbf{t}_{DM}^i,\mathbf{Z}_M\otimes\mathbf{I}_D)$ 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 ([15]).

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

For each iteration of SPICE, do the following:

1) Given $\{p_k^{i-1}\}_{k=0}^{K-1}$ from the previous iteration, compute the auxiliary block matrix $\bar{\mathcal{R}}_{D(M-1)}$ using a series of FFT 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 $\mathbf{\Delta}_{M-1}$ 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 $\parallel \tilde{\mathbf{R}}_{DM}^{-1}\hat{\mathbf{R}}_{DM}^{1/2} \parallel_F$ 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 $\parallel \tilde{\mathbf{R}}_{DM}^{-1}\hat{\mathbf{R}}_{DM}^{1/2} \parallel_F$, calculate $\rho^{(i)}$ in $\mathcal{O}(K)$ flops.
7) Given $\parallel \tilde{\mathbf{R}}_{DM}^{-1}\hat{\mathbf{R}}_{DM}^{1/2} \parallel_F$, calculate $\sigma^{(i)}$ in $\mathcal{O}(1)$ flops.
8) Given $\mathbf{a}_{DM}(\theta_k)\tilde{\mathbf{R}}_{DM}^{-1}\hat{\mathbf{R}}_{DM}^{1/2}$, $k = 0, \ldots, K-1$, 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 fast implementations of the SPICE algorithm. Without any loss of generality, only the estimation of the azimuth angle is considered herein.

Consider five uncorrelated sources located at $\theta_1 = -44.4°$, $\theta_2 = -36.9°$, $\theta_3 = -32.7°$, $\theta_4 = 11.5°$ and $\theta_5 =$

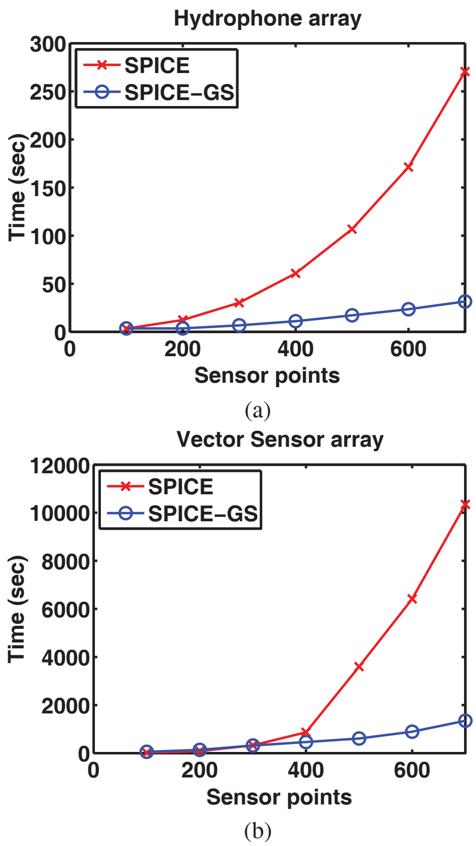

Fig. 1. Computational time comparison of the direct implementation of SPICE and fast implementations (SPICE-GS) for (a) hydrophone ULA and (b) vector-sensor ULA against the number of sensors M in the array, with $N = 10M$ available snapshots and $K = 10M$ angular scanning points.

53.1°, respectively. The corresponding 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 $n \in \{1, \dots, N\}$ denotes the index of the available snapshots. The phase values $\{\xi_1(n)\}, \dots, \{\xi_5(n)\}$ are independent and identically distributed random variables with uniform distribution over $[0, 2\pi)$. The noise terms are assumed to be circularly symmetric complex Gaussian random variables that are spatially and temporally white.

Figure 1 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 arrays with different number (M) of sensors. The simulations are implemented in MATLAB® using only one core of a workstation with an 8-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 Signal-to-noise ratio (SNR) is fixed at 5 dB and 40 SPICE iterations are performed. For a large value of M, the fast implementations provide significant time reduction compared with the direct implementations, which makes the fast implementation especially attractive for high resolution localization applications. Note that the computational complexity reduction in the vector sensor ULA scenario is less dramatic than that in the hydrophone ULA scenario. The displacement rank of the covariance matrix $\tilde{\mathbf{R}}_{DM}^{-1}$ of the vector-sensor ULA is $2D$ instead of 2 in $\mathbf{R}_M^{-1}$ as in the hydrophone ULA case.

VII. Conclusions

We have explored the DOA source localization problem for hydrophone ULAs and vector-sensor ULAs. The SPICE algorithm was reviewed and its fast implementations have been presented. The computational complexity of SPICE was reduced by exploiting the Toeplitz/block Toeplitz matrix structure in the SPICE covariance matrix and by utilizing FFT to compute spectral estimates for ULAs. In the simulations, it has been shown that a significant computational efficiency increase is obtained.

References

[1] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*, pp. 1194, 599–603. New York, NY: John Wiley & Sons, 2002.

[2] 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 Transactions on Signal Processing*, vol. 59, pp. 35–47, Jan 2011.

[3] P. Stoica, P. Babu, and J. Li, "SPICE: A Sparse Covariance-Based Estimation Method for Array Processing," *IEEE Transactions on Signal Processing*, vol. 59, pp. 629–638, Feb 2011.

[4] B. Musicus, "Fast MLM power spectrum estimation from uniformly spaced correlations," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, pp. 1333–1335, Oct 1985.

[5] Z.-S. Liu, H. Li, and J. Li, "Efficient implementation of Capon and APES for spectral estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, pp. 1314–1319, Oct 1998.

[6] T. Ekman, A. Jakobsson, and P. Stoica, "On the efficient implementation of the Capon spectral estimator," *Proceedings of European Signal Processing Conference,* Tampere, Finland, September 4-8 2000.

[7] A. Jakobsson, J. S. L. Marple, and P. Stoica, "Computationally efficient two-dimensional Capon spectrum analysis," *IEEE Transactions on Signal Processing*, vol. 48, pp. 2651–2661, Sep 2000.

[8] G.-O. Glentis, "A fast algorithm for APES and Capon spectral estimation," *IEEE Transactions on Signal Processing*, vol. 56, pp. 4207–4220, Sep 2008.

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

[10] A. Nehorai and E. Paldi, "Acoustic vector-sensor array processing," *IEEE Transactions on Signal Processing*, vol. 42, pp. 2481–2491, Sep 1994.

[11] M. Hawkes and A. Nehorai, "Acoustic vector-sensor beamforming and Capon direction estimation," *IEEE Transactions on Signal Processing*, vol. 46, pp. 2291 –2304, Sep 1998.

[12] H. Akaike, "Block Toeplitz matrix inversion," *SIAM Journal on Applied Mathematics*, vol. 24, no. 2, pp. 234–241, 1973.

[13] I. C. Gohberg and A. Semencul, "On inversion of finite-section Toeplitz matrices and their continuous analogues," *Mat. Issled. Kishinev*, pp. 201–224, 7:2 (1972) (In Russian).

[14] J. Jain, "An efficient algorithm for a large Toeplitz set of linear equations," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, pp. 612–615, Dec 1979.

[15] Q. Zhang, H. Abeida, M. Xue, W. Rowe, and J. Li, "Fast Implementation of Sparse Iterative Covariance-based Estimation for Source Localization," to appear in *the Journal of the Acoustical Society of America.*

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

[17] D. A. Harville, *Matrix Algebra from a Statistician's Perspective*, pp. 88 –103. New York, NY: Springer, Inc., 1997.

[18] M. Wax and T. Kailath, "Efficient inversion of Toeplitz-block Toeplitz matrix," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, pp. 1218–1221, Oct 1983.

[19] S. Zohar, "Toeplitz matrix inversion: The algorithm of W. F. Trench," *Journal of the Association for Computing Machinery*, vol. 16, pp. 592–601, Oct 1969.