Memory-efficient approximate three-dimensional beamforming

Nicholas R. Rypkema, Erin M. Fischell, and Henrik Schmidt

Citation: The Journal of the Acoustical Society of America **148**, 3467 (2020); doi: 10.1121/10.0002852

View online: https://doi.org/10.1121/10.0002852

View Table of Contents: https://asa.scitation.org/toc/jas/148/6

Published by the Acoustical Society of America

JASA ARTICLE

Memory-efficient approximate three-dimensional beamforming

Nicholas R. Rypkema,[1,a)] Erin M. Fischell,[1] and Henrik Schmidt[2]

[1] *Applied Ocean Physics & Engineering, Woods Hole Oceanographic Institution, Woods Hole, Massachusetts 02543, USA*

[2] *Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

ABSTRACT:

Localization of acoustic sources using a sensor array is typically performed by estimating direction-of-arrival (DOA) via beamforming of the signals recorded by all elements. Software-based conventional beamforming (CBF) forces a trade-off between memory usage and direction resolution, since time delays associated with a set of directions over which the beamformer is steered must be pre-computed and stored, limiting the number of look directions to available platform memory. This paper describes a DOA localization method that is memory-efficient for three-dimensional (3D) beamforming applications. Its key lies in reducing 3D look directions [described by azimuth/inclination angles (ϕ, θ) when considering the array as a whole] to a single variable (a conical angle, ζ) by treating the array as a collection of sensor pairs. This insight reduces the set of look directions from two dimensions to one, enabling computational and memory efficiency improvements and thus allowing direction resolution to be increased. This method is described and compared to CBF, with comparisons provided for accuracy, computational speedup, and memory usage. As this method involves the incoherent summation of sensor pair outputs, gain is limited, restricting its use to localization of strong sources—e.g., for real-time acoustic localization on embedded systems, where computation and/or memory are limited. © *2020 Acoustical Society of America*.

https://doi.org/10.1121/10.0002852

(Received 21 April 2020; revised 30 October 2020; accepted 10 November 2020; published online 7 December 2020)

[Editor: Karim G. Sabra] Pages: 3467–3480

I. INTRODUCTION

Beamforming or spatial filtering is a mature and widely used technique for direction-of-arrival (DOA) estimation using a spatially distributed array of sensors to sample the incoming signal and phase-shifting (or time-aligning) these signals into coherence. Originally developed for radar,[1] conventional beamforming (CBF)[2] has found wide application in communications,[3] imaging,[4] and, in particular, in SONAR source localization, tracking, and classification.[5–8]

CBF performs poorly in its ability to resolve two closely-spaced sources, and the desire to improve this angular resolution (this term is analogous to resolving power in optical systems) for target tracking applications drove the development of more computationally complex so-called adaptive or super-resolution beamforming methods. These include minimum variance distortionless-response (MVDR) beamforming,[9] which seeks to minimize the average output power while passing a signal from a given look direction; and the multiple signal classification (MUSIC) beamforming method,[10] which uses eigendecomposition of the autocorrelation matrix into signal and noise subspaces, and performs a weighted beamforming of these components. More recently, compressive sensing has been employed to improve angular resolution;[11–13] these methods solve the linear set of beamforming equations formed by collecting a set of steering vectors and performing convex optimization under a sparsity constraint imposed by minimization of the l_1-norm of the signal.

In contrast to the amount of work done on improving angular resolution, comparatively little has been done on improving beamforming efficiency. Since early CBF operated in the time-domain, and with the desire to steer the beamformer in small increments to increase direction resolution, high sampling frequencies were necessary—as such, early efficiency improvements looked to decrease the amount of sampled data to be stored, using techniques such as partial-sum, interpolation, and shifted-sideband (down-sampling) methods.[14] The shift to frequency-domain CBF superseded these approaches by relaxing the sampling rate to only satisfying the Nyquist criterion. More recent work has used the fact that sensor arrays that sample uniformly in space are analogous to uniform temporal sampling to exploit the computational efficiencies afforded by the fast Fourier transform; this was first proposed by Williams[15] who derived this spatial Fourier transform for a fast beamforming algorithm for line arrays. This work was extended to uniform circular arrays[16] and, finally, to uniform grid arrays.[17] A recent extension of this method using the chirp Z-transform (CZT) enables similarly efficient processing for wideband beamforming.[18] Each of these methods are limited by the need to use uniform (equispaced) arrays due to their use of the spatial Fourier transform; the use of the non-uniform fast Fourier transform (NUFFT) in recent work has overcome this limitation.[19] In each of the aforementioned methods, the phase-shifts associated with all look directions

[a)] Electronic mail: nrypkema@whoi.edu, ORCID: 0000-0003-0874-8980.

are stored—thus, they are computationally but not memory efficient.

This paper introduces a technique for improved memory and computational efficiency for approximate three-dimensional (3D) beamforming applications with small arrays, termed *sensor pair decomposition beamforming* (SPD-BF). As CBF typically requires the storage of phase-shifts for the look directions over which it is steered, in 3D beamforming this translates to a quadratic increase in memory usage with direction resolution since it must be steered over both azimuth and inclination. SPD-BF ties memory usage and computation time to the *number of pairs of sensors* in the array, enabling a memory-efficient increase in direction resolution. Since SPD-BF involves the incoherent summation of sensor pair outputs, its gain is fixed and does not increase with more array elements, resulting in degraded angular resolution performance in comparison to CBF. Note that "direction resolution" and "angular resolution" are distinct—the smaller the increment in steering angle for a set number of look directions, the higher the direction resolution, while angular resolution concerns how well a beamformer can resolve closely spaced sources. Thus, SPD-BF achieves gains in computation time, memory efficiency, and direction resolution at the expense of array gain and angular resolution. This makes this method well-suited for applications where high signal-to-noise ratio (SNR) is anticipated, e.g., beacon-based ultra-short baseline (USBL) or inverted USBL localization systems. This paper details this approximate beamforming method, its advantages and limitations, and its performance against CBF with simulated and real data.

II. CONVENTIONAL BEAMFORMING

Given an arbitrary array with N sensor elements, a signal originating from direction $\boldsymbol{a}$ with speed c will, under a far-field planar wave assumption, arrive at sensor i with time delay,

$$\tau_i = \frac{\boldsymbol{a}^T \boldsymbol{p}_i}{c}, \quad \text{where} \quad \boldsymbol{a} = \begin{pmatrix} -\sin(\theta)\cos(\phi) \\ -\sin(\theta)\sin(\phi) \\ -\cos(\theta) \end{pmatrix}. \tag{1}$$

CBF applies phase shifts to the measured signals with the goal of negating this natural array response to the incoming wave as defined by these time delays, for a beamformed output at a given look direction (θ, ϕ) and wave frequency ω of

$$Y[\omega; \theta, \phi] = \sum_{i=1}^{N} H_i[\omega; \theta, \phi] \cdot X_i[\omega], \quad \text{where} \quad H_i[\omega; \theta, \phi] = e^{j\omega\tau_i}, \tag{2}$$

and where $X_i[\omega]$ is the Fourier transformed signal received by sensor i. The top part of Fig. 1 shows this process. Often the objective of this beamforming operation in reception is DOA estimation, which in the case of wideband CBF is generally the power summed over all M frequencies,

$$|\tilde{Y}[\theta, \phi]|^2 = \frac{1}{M} \sum_{k=1}^{M} |Y[\omega_k; \theta, \phi]|^2. \tag{3}$$

This output is computed over a set of look directions (θ, ϕ), with estimated DOA selected as

$$\tilde{\theta}, \tilde{\phi} = \arg\max_{\theta, \phi} |\tilde{Y}[\theta, \phi]|^2. \tag{4}$$

Computation time for estimating DOA in this manner scales with the number of look directions, and since the phase shifts associated with the look directions are typically pre-computed and stored in memory to improve computation speed, memory usage also scales similarly. For 3D beamforming this is particularly problematic, since the number of look directions is $N_{angles} = N_\theta N_\phi$, where N_θ is the number of look inclinations and N_ϕ is the number of look azimuths. For simplicity, consider dividing the azimuth and inclination search space so as to have equal spacing, i.e., $N_\phi = 2N_\theta$. If we label $N_\theta = C$, then the number of gridded look directions is

$$N_{angles} = 2C^2. \tag{5}$$

Estimating DOA with Eq. (4) over this grid is thus quadratic in the parameter C ($\mathcal{O}(C^2)$); pre-computation and storage of the phase shifts drastically reduces the computation time for each look direction, but introduces a memory requirement also on the order of C^2. This scaling factor can severely restrict the direction resolution of beamforming in computationally and/or memory limited systems.

III. MEMORY-EFFICIENT APPROXIMATE 3D BEAMFORMING

A. Key insight and application to example array

Array steering in 3D CBF requires calculating time delays that are dependent on array geometry and the look direction using Eq. (1). However, when performing CBF with a 1D linear array, placing the sensor elements on the z–axis as is convention reduces this equation to

$$\tau_i = \frac{-\cos(\theta) z_i}{c}. \tag{6}$$

Thus, for linear arrays, any 3D look direction given by ϕ and θ can be solely described using its inclination angle θ; this insight manifests in the well-known conical beampattern of a linear array. In essence, this means that sets of 3D look directions that point along the surface of a cone induce the same time delays at the sensors of a linear array, allowing these look directions to be described by a single *conical angle* ζ.

Consider the 3-element right-angle planar array in the leftmost plot of Fig. 2, along with a desired set of 16 look directions to beamform at as red arrows. Now, instead of beamforming at each of these look directions, the array is *decomposed* into its unique pairs of sensors, as shown in the

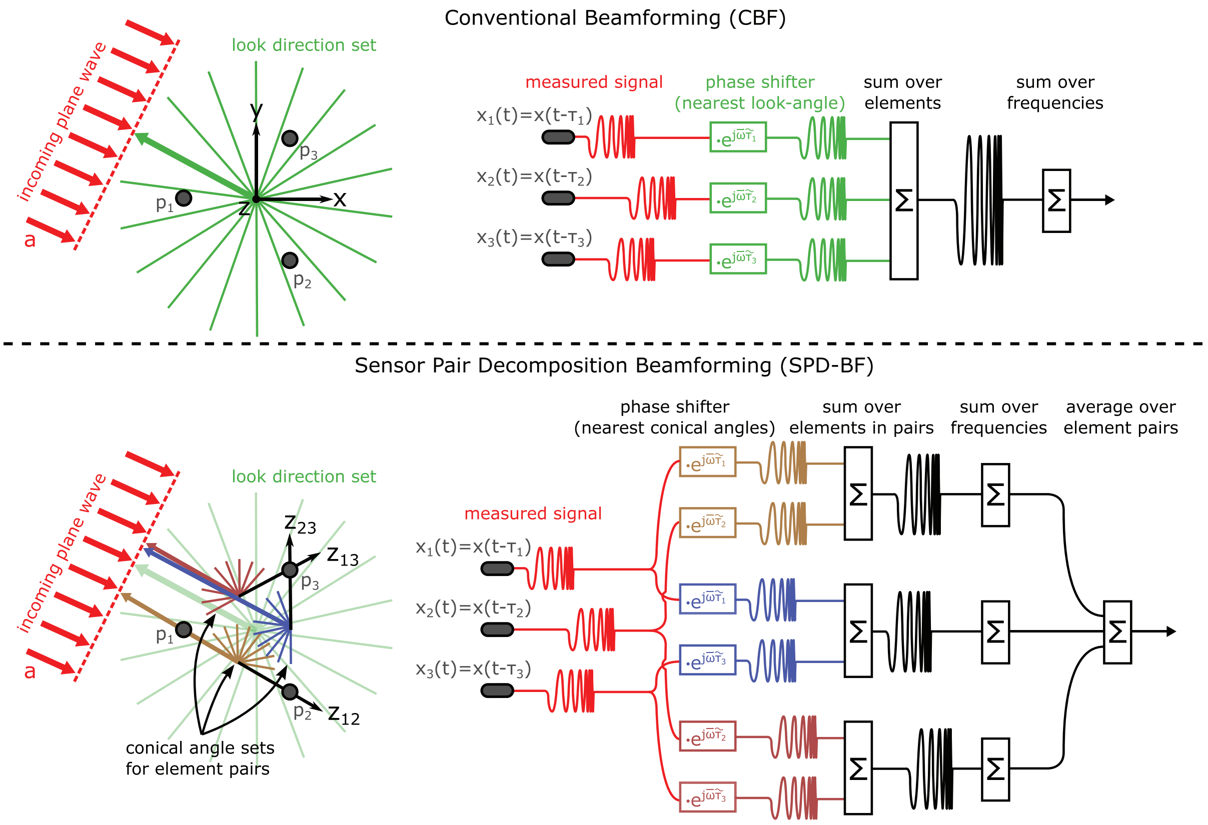

FIG. 1. (Color online) Top: In CBF, the output power is calculated over a set of look directions; when the look direction is pointing toward the incoming signal, measured signals are phase-shifted into alignment and the output power is maximum. Bottom: In SPD-BF, the array is decomposed into sensor pairs and CBF is performed over a set of *conical angles* for each pair; the output power of each pair is then summed to obtain the final output for improved memory and computation performance for 3D beamforming with small arrays.

next three plots of Fig. 2. This decomposition reveals that for the vertical sensor pair, the set of 16 look directions can be represented by just 4 conical angles, a reduction factor of 4. For the horizontal sensor pair, the look directions can be reduced to 6 conical angles, while for the final (diagonal) sensor pair, no reduction is possible and 16 conical angles remain. Using these conical angles, we must beamform at a total of $4 + 6 + 16 = 26$ angles, rather than the original 16 look directions; thus, in this example it is less efficient in terms of both computation and memory to do so as compared to 3D CBF, but this example provides an intuitive understanding of our approximate beamforming approach.

B. Approximate beamforming using conical angles

To demonstrate how our approach can improve efficiency, instead of finding conical angles from look directions as in Sec. III A, look directions can be determined from the set of conical angles of the three sensor pairs, as illustrated in Fig. 3. In the left-hand plots, the vertical sensor pair conical angle space is divided into 3 conical angles, as is done for the horizontal sensor pair; and for the diagonal sensor pair, the space is divided into 5 conical angles. In the rightmost plot, all three sets of conical angles and the full array are plotted—it is apparent that the three sets of cones from each sensor pair intersect at exactly 9 look directions, shown as arrows in the rightmost plot. Thus, in this

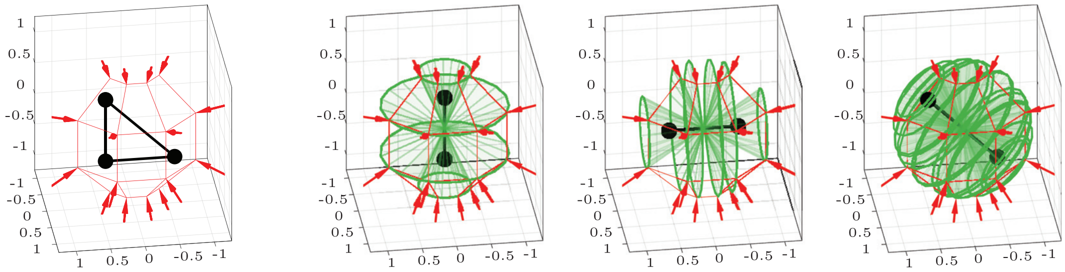

FIG. 2. (Color online) Look directions can be represented by conical angles in the space of the array's decomposition into sensor pairs—the 3-element array on the left is decomposed into its three unique sensor pairs, shown on the right; the set of 16 look directions can be collapsed to 4 conical angles for the vertical sensor pair, into 6 conical angles for the horizontal sensor pair, and remains as 16 conical angles for the diagonal sensor pair.

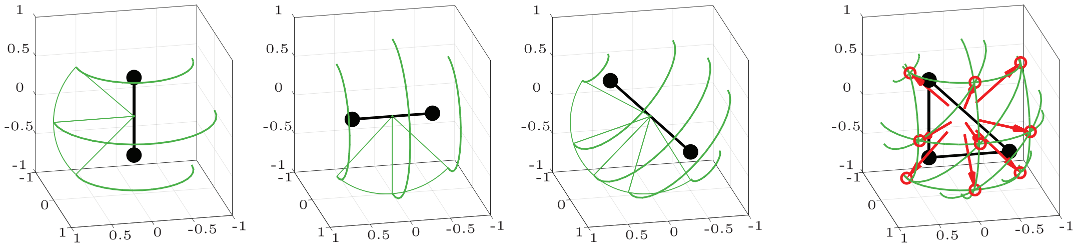

FIG. 3. (Color online) Look directions can be constructed by conical angles in the space of the array's decomposition into sensor pairs—each pair contributes a set of conical angles to the approximate beamforming output, as shown in the three leftmost plots; these conical angle sets intersect along nine look directions, as shown as red arrows on the right. The number of look directions scales quadratically with a linear increase in the number of conical angles per sensor pair.

example, $3+3+5=11$ conical angles have been used to represent a set of only 9 look directions, which is again less efficient. However, note that if the number of look directions in the constructed "grid" is continually increased, the corresponding number of conical angles to construct that grid scales at a significantly slower rate. For C conical angles in the vertical and horizontal sensor pairs of this array, the number of intersections scales as C^2, while the number of conical angles across all three pairs scales as

$$C_{total} = C + C + (2C - 1) = 4C - 1. \quad (7)$$

Thus, for this particular case, the total number of conical angles grows linearly, while the total number of constructed look directions grows quadratically—consequently, for a large number of look directions, our approximate beamforming method has a greater efficiency in terms of computation and memory usage.

Figure 3 highlights a couple of caveats with our beamforming approach. First, note that each sensor pair has its own phase center half-way between its two sensors—these phase centers are termed *pair phase centers*. Since pair phase centers are not aligned with the array phase center (as shown in the rightmost plot of Fig. 3), the output of our beamformer is only an *approximation* of the CBF output—this limitation is briefly expanded upon later in the paper. Second, notice that the construction of the look directions is dependent on the geometry of the array—depending on how conical angles are constructed, in general, intersections will only occur between two sensor pairs. For example, if the diagonal element pair in Fig. 3 had only a single conical angle at 90°, only three of the resulting look directions will intersect along the conical angles of all sensor pairs. The accuracy of the look direction construction is thus a complicated function of the number of array sensor pairs, array geometry, and the number of conical angles for each sensor pair.

These toy examples highlight the simple intuition behind our approximate beamforming method, the steps of which are illustrated conceptually at the bottom of Fig. 1. The three-element array in this figure is first decomposed into its three sensor pairs, each within its own coordinate system (z_{12}, z_{13}, z_{23}); this allows each sensor pair to be treated as a linear array, whose phase center is halfway between the sensors (the *pair phase center*). This is followed by performing CBF over a set of *conical angles* (shown in red, blue, and brown) for each of the sensor pairs. Conical angles for each pair are not selected such that they intersect exactly as in Fig. 3, but instead a single set of C conical angles is used for all pairs,

$$S_{conical} = \{\zeta_1, \zeta_2, \ldots, \zeta_C\}, \quad (8)$$

where $0 \leq \zeta_i < 180°$. Generally, the 180° domain of conical angles is divided uniformly to generate this set. Finally, to get the approximate beamforming output for a desired look direction (shown in green), the CBF output of the conical angle that is nearest to the look direction is found from each sensor pair, and these are averaged together.

An example illustration of the output resulting from our approximate beamforming method is shown in Fig. 4. For each sensor pair of the triangle array in this figure, CBF is performed along a set of conical angles with a half-degree resolution. The beamformed output for each pair is shown by projecting the output for each conical angle onto the sphere—the resulting rotationally symmetric "banding" illustrates the ambiguity associated with conical angles per sensor pair. The approximate beamforming output is obtained by summing the outputs of the array's pairs, disambiguating the direction of the incoming wave through constructive and destructive summation.

C. Memory efficiency

The computational and memory cost of CBF was shown previously to be proportional to the number of look directions, which is approximated in Eq. (5) as $2C^2$ for a grid covering the sphere. For our approximate beamforming method, if the same number C is taken as the number of conical angles for each sensor pair, then the computational and memory cost grows as the product of the number of array sensor pairs and the number of conical angles,

$$N_{angles} = PC, \quad (9)$$

where P is the number of sensor pairs. This PC value is significantly smaller than the $2C^2$ value obtained for CBF in the case of small arrays: where the total number of pair combinations is typically much smaller than the desired number of conical or inclination angles C. In addition, PC is an *upper bound* on memory usage because a further efficiency

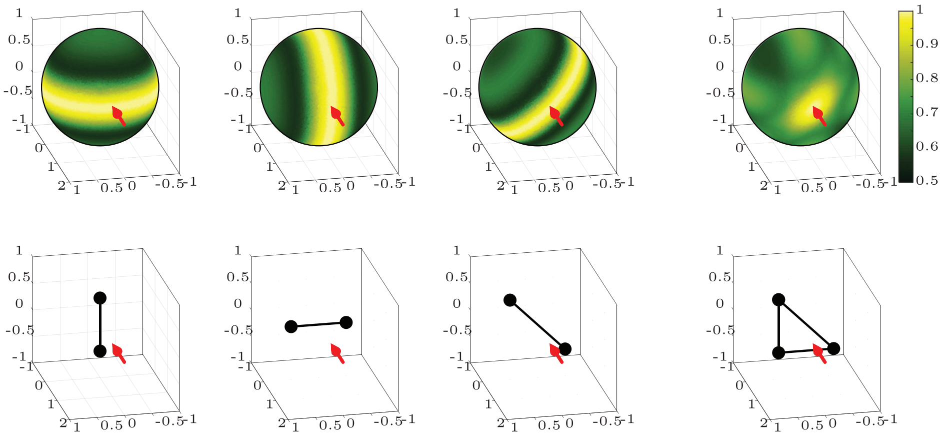

FIG. 4. (Color online) Illustration of how the output of SPD-BF is constructed from sensor pair CBF outputs—the full approximated output in the rightmost plot is generated by summation of the CBF outputs over the conical angles for each sensor pair on the left. The ambiguity associated with the CBF output of each pair is removed via constructive/destructive summation, enabling the array to detect the DOA as indicated by the red arrow.

can be exploited: if two sensor pairs within an array have elements that are separated by the same distance, and the same set of conical angles is used for both pairs, then these two pairs are identical linear arrays and the phase shifts stored for both are also identical; as a result, only one set of phase shifts has to be stored in memory for both sensor pairs. As an example, the memory savings for a regular tetrahedral array are enormous, since all six sensor pairs are separated by the same distance—the computational cost is approximately $6C$, while the memory cost is only C.

Since one of its core concepts is the decomposition of the array into sensor pairs, our approximate beamforming method is termed *sensor pair decomposition beamforming* (SPD-BF).

D. SPD-BF formulation

Given some arbitrary array defined by its N sensor positions $\boldsymbol{p}_i$, the array is first decomposed into its $N(N-1)/2$ sensor pairs. The unique combinations of pairs of elements are described by the elements above the diagonal of an $N \times N$ matrix, where the row indexed by i encodes the first element and the column indexed by j encodes the second element,

$$\begin{array}{cc} & \text{2nd el.}\,j \\ \text{1st el.}\,i & \begin{bmatrix} - & (\boldsymbol{p}_1,\boldsymbol{p}_2) & (\boldsymbol{p}_1,\boldsymbol{p}_3) & \cdots & (\boldsymbol{p}_1,\boldsymbol{p}_N)\\ - & - & (\boldsymbol{p}_2,\boldsymbol{p}_3) & \cdots & (\boldsymbol{p}_2,\boldsymbol{p}_N)\\ - & - & - & \ddots & \vdots \\ - & - & - & - & (\boldsymbol{p}_{N-1},\boldsymbol{p}_N)\\ - & - & - & - & - \end{bmatrix} \end{array}, \qquad (10)$$

$$S_{unique} = \{(\boldsymbol{p}_1,\boldsymbol{p}_2),\ldots,(\boldsymbol{p}_i,\boldsymbol{p}_j)\}$$
$$\text{such that} \quad i \neq j. \qquad (11)$$

For each sensor pair, the distance between elements is

$$D_{ij} = ||\boldsymbol{p}_i - \boldsymbol{p}_j||. \qquad (12)$$

The sensor pairs are constructed as a set of linear arrays in their own coordinate systems, with each pair phase center halfway between the elements,

$$S_{pairs} = \left\{\left[\frac{-D_{12}}{2}\,\frac{D_{12}}{2}\right]^T,\ldots,\left[\frac{-D_{ij}}{2}\,\frac{D_{ij}}{2}\right]^T\right\}$$
$$= \{\boldsymbol{P}_{12},\ldots,\boldsymbol{P}_{ij}\}, \qquad (13)$$

where

$$\boldsymbol{P}_{ij} = [P_i\,P_j]^T \quad \text{with} \quad P_i = \frac{-D_{ij}}{2}, \quad P_j = \frac{D_{ij}}{2}. \qquad (14)$$

This unique coordinate system in which a sensor pair linear array resides is referred to as the *conical angle space*. In the conical angle space, the time delay experienced by each element of the pair referenced to the pair phase center is given by

$$\boldsymbol{\tau}_{ij} = \frac{-\cos(\zeta)}{c}\boldsymbol{P}_{ij}, \qquad (15)$$

where ζ is the conical angle and c is the signal speed. The wideband CBF spatial filter for the sensor pair is given as

$$\boldsymbol{H}_{ij}[\boldsymbol{\omega};\zeta] = e^{j\boldsymbol{\omega}\boldsymbol{\tau}_{ij}} = \begin{bmatrix}\boldsymbol{H}_i[\boldsymbol{\omega};\zeta]\\ \boldsymbol{H}_j[\boldsymbol{\omega};\zeta]\end{bmatrix} = \begin{bmatrix} e^{j\boldsymbol{\omega}\tau_i} \\ e^{j\boldsymbol{\omega}\tau_j}\end{bmatrix}$$
$$= \begin{bmatrix} e^{j\omega_1\tau_i} & e^{j\omega_2\tau_i} & \ldots & e^{j\omega_M\tau_i} \\ e^{j\omega_1\tau_j} & e^{j\omega_2\tau_j} & \ldots & e^{j\omega_M\tau_j}\end{bmatrix}. \qquad (16)$$

The beamformed output of the sensor pair for a given conical angle is thus given by

$$\boldsymbol{Z}_{ij}[\boldsymbol{\omega};\zeta] = \frac{1}{2}\left(\boldsymbol{H}_i[\boldsymbol{\omega};\zeta]\odot\boldsymbol{X}_i[\boldsymbol{\omega}] + \boldsymbol{H}_j[\boldsymbol{\omega};\zeta]\odot\boldsymbol{X}_j[\boldsymbol{\omega}]\right), \qquad (17)$$

where $\odot$ is the element-wise multiplication operator, and $\boldsymbol{X}_i[\omega_k]$ is the kth FFT bin from element i. The frequency-averaged output for a given sensor pair at a particular conical angle is then

$$|\hat{Z}_{ij}[\zeta]| = \frac{1}{M}\sum_{k=1}^{M}|\boldsymbol{Z}_{ij}[\omega_k;\zeta]|. \qquad (18)$$

To average across all sensor pairs, a conversion must be done from the azimuth-inclination (ϕ,θ) space of look directions to the conical angle space (ζ). This is done by calculating the angle between the vector joining the two elements, and the vector represented by the look direction

$$\text{Let} \quad \boldsymbol{a}_{ij} = \frac{\boldsymbol{p}_i - \boldsymbol{p}_j}{||\boldsymbol{p}_i - \boldsymbol{p}_j||}, \quad \boldsymbol{b}(\phi, \theta) = \begin{bmatrix} \sin(\theta)\cos(\phi) \\ \sin(\theta)\sin(\phi) \\ \cos(\theta) \end{bmatrix} \tag{19}$$

then

$$\zeta = \arccos(\boldsymbol{a}_{ij} \bullet \boldsymbol{b}(\phi, \theta)). \tag{20}$$

Rewriting Eq. (18) in terms of the look direction

$$|\hat{Z}_{ij}[\phi, \theta]| = \frac{1}{M} \sum_{k=1}^{M} \left| \boldsymbol{Z}_{ij}\left[\omega_k; \arccos(\boldsymbol{a}_{ij} \bullet \boldsymbol{b}(\phi, \theta))\right] \right|. \tag{21}$$

The full approximate beamforming output for a given look direction is finally calculated by averaging over all sensor pairs,

$$|\hat{Z}[\phi, \theta]| = \frac{1}{M} \frac{2}{N(N-1)} \sum_{i,j}^{N-1,N} \sum_{k=1}^{M} \left| \boldsymbol{Z}_{ij}\left[\omega_k; \arccos(\boldsymbol{a}_{ij} \bullet \boldsymbol{b}(\phi, \theta))\right] \right|. \tag{22}$$

In principle, given a static grid of look directions, the required conical angles may be pre-calculated for each element pair to intersect the look directions precisely using Eq. (20). However, in practice, for more flexibility the phase shifts for a static set of conical angles are pre-computed per sensor pair, and Eq. (20) is used to find the nearest pre-computed conical angle in the set, providing the benefit of the greater memory savings due to identical pair separation distances. Thus, the output for a given look direction is the summation of the nearest neighbor conical angles for each sensor pair. If the grid of look directions is static, a look-up table can also be constructed to bypass the use of Eq. (20) during runtime, providing an additional speedup.

IV. SIMULATION COMPARISONS

To evaluate the performance of SPD-BF, it is compared to CBF in terms of fundamental properties such as its beampattern, accuracy, angular resolution, and processing speed. In this section simulations are used to provide a comparative analysis between the two approaches, using the base parameters listed in Table I.

TABLE I. Base parameters for comparative analysis of SPD-BF and CBF.

f_s	c	S_{angles}	$S_{conical}$
37.5 kHz	1480 ms^{-1}	$\theta = 0° : 1° : 179°$ $\phi = 0° : 1° : 359°$ (180 × 360 angles)	$\zeta = 0° : 1° : 179°$ (180 angles)

A. Beampattern

The outputs using the uniform line array (ULA) and regular cube array geometries listed in Table II were used to compare the beampatterns produced by CBF and SPD-BF. The 1D ULA was included for easy visualization of the 2D beamformed output, beamforming over a single inclination angle (θ) for this geometry.

An incoming acoustic plane wave was simulated in all cases using a 8.25–10.25 kHz, 20 ms linear frequency modulated (LFM) chirp, incident onto the array from a randomly selected azimuth-inclination (ϕ, θ) look direction; each element of the array records 8000 samples, which represents 213.3 ms of data; the center frequency of 9.25 kHz is chosen such that the minimum element spacing of the arrays in Table II is one half the wavelength of this center frequency. Representative examples of the normalized beamformed outputs from both approaches are shown in Figs. 5 and 6 for the line and cube arrays.

The beampatterns from both methods share similar features, and similar maxima positions. It is apparent that the maximums occur close to the true DOAs, indicating that both approaches share a similar level of accuracy. The increased "floor" of SPD-BF due to sensor pair summation is especially apparent in the beampatterns of the cube array, where the floor is approximately 0.4. This suggests a reduction in the resolving power of the array using SPD-BF for the same look-angle resolution, due to the increase in the half-power beamwidth.

B. Accuracy

To estimate the accuracy of both methods, the simulated measurement on each sensor is corrupted with white Gaussian noise. Increasing levels of noise were added to vary the signal-to-noise ratio (SNR) from 25 to 5 dB, and to 0 dB. Two hundred simulations were performed for each array and SNR level, and the angular difference between the true DOA and the arg-maximum value from the response of both CBF and SPD-BF was determined as

$$\epsilon = \arccos(\boldsymbol{v}_{max} \bullet \boldsymbol{v}_{true}), \tag{23}$$

where $\boldsymbol{v}_{true}$ is the unit vector pointing toward the true DOA and $\boldsymbol{v}_{max}$ is the unit vector pointing in the arg-

TABLE II. Array sensor positions (x, y, z in cm) to compare outputs of CBF and SPD-BF.

Element	ULA	Regular tetrahedral	Regular octagonal	Regular cube
1	(0, 0, 0)	(0,0,6.54)	(0,0,5.66)	(4, 4, 4)
2	(8, 0, 0)	(4.62,0,0)	(4, 4, 0)	(−4,4,4)
3	(16, 0, 0)	(−2.31,−4,0)	(−4,4,0)	(4,−4,4)
4	(24, 0, 0)	(−2.31,4,0)	(4,−4,0)	(−4,−4,4)
5	(32, 0, 0)	—	(−4,−4,0)	(4,4,−4)
6	(40, 0, 0)	—	(0,0,−5.66)	(−4,4,−4)
7	(48, 0, 0)	—	—	(4,−4,−4)
8	(56, 0, 0)	–	–	(−4,−4,−4)

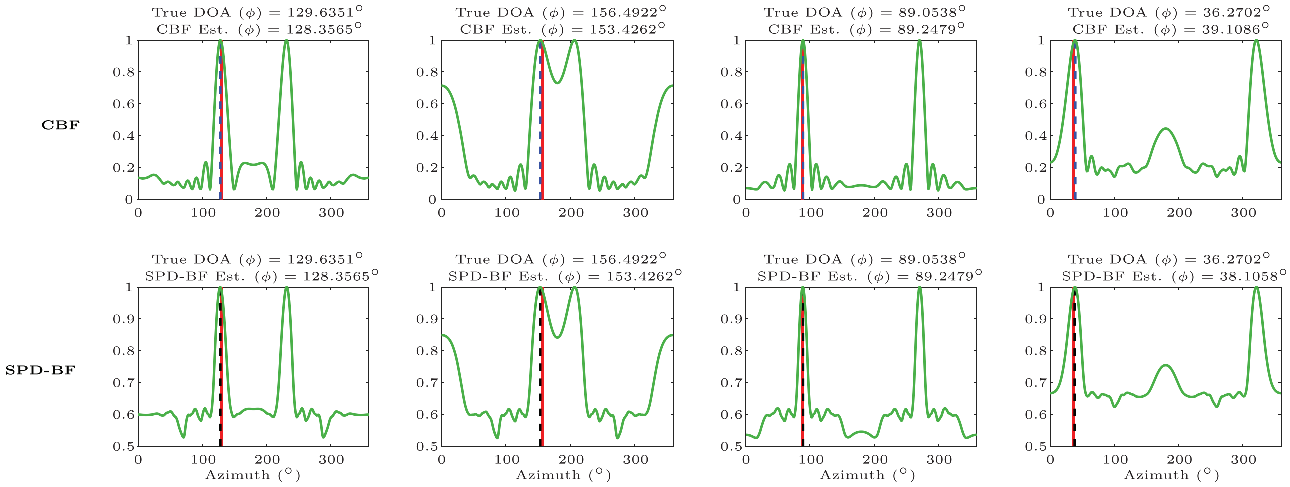

FIG. 5. (Color online) Sample beampatterns from CBF and SPD-BF for an 8 element uniform line array—the true DOA is indicated by the solid red line, and the estimates from maximums using CBF and SPD-BF are indicated by the dashed blue and black lines respectively. Top: beampattern from CBF. Bottom: beampattern from SPD-BF. Note the difference in scale.

maximum direction as determined by the two methods. The accuracy statistics resulting from these simulations are plotted in Fig. 7. These plots show the probability distribution of the errors (ϵ) over all SNRs on the left axis, and the empirical cumulative distribution function (CDF) of error for individual SNRs on the right axis. Two observations are consistent across both CBF and SPD-BF: the error distribution of accuracy improves as the number of array sensors increases, since the noise gets "averaged out" over more sensors and beamwidth reduces; and as the SNR increases, the accuracy improves. An interesting characteristic of the ULA is that its accuracy distribution has a long tail, with a wide variation; this is due to the degradation in the array response as the source moves from broadside to end-fire. Looking at the CDFs across all geometries, it is apparent that, in general, the accuracy of SPD-BF is slightly worse than that of CBF, likely due to larger beamwidths associated with the SPD-BF method as shown in Sec. IV C. In low SNR (high noise) situations, the accuracy of both methods are comparable, which is especially apparent at 0 dB across the three 3D arrays, and at all noise levels for the ULA. In contrast, at the high SNR of 25 dB, the CDFs for the 3D arrays show that the accuracy of CBF is significantly better than that of SPD-BF—this suggests that at low SNR, noise is the predominant cause of accuracy loss for both methods; however, high SNRs reveal limitations in accuracy of our method. It is likely that these limitations are due in large part to the number of conical angles used in SPD-BF for these simulations—indeed, we show in Sec. IV E that increasing the conical angle resolution significantly mitigates this loss in accuracy seen in the low-noise scenario.

C. Half-power beamwidth

The half-power beamwidth of the main lobe represents the -3 dB level from the peak, or about the 0.707 level in the array response of both approaches. This can be estimated by

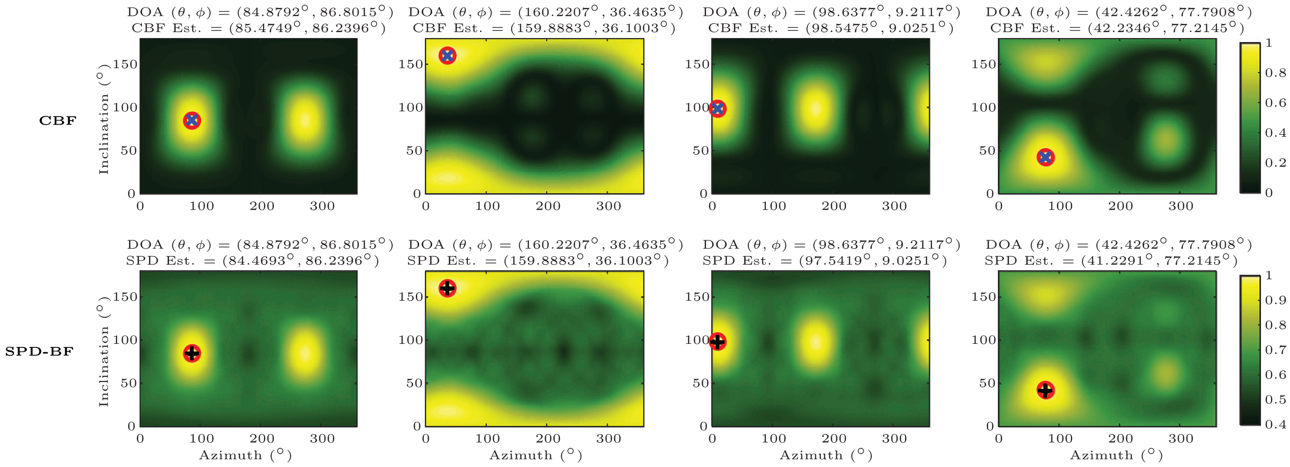

FIG. 6. (Color online) Sample beampatterns from CBF and SPD-BF for an 8 element regular cube array—the true DOA is indicated by the red circle, and the estimates from maximums using CBF and SPD-BF are indicated by the blue and black crosses respectively. Top: beampattern from CBF. Bottom: beampattern from SPD-BF. Note the difference in scale.

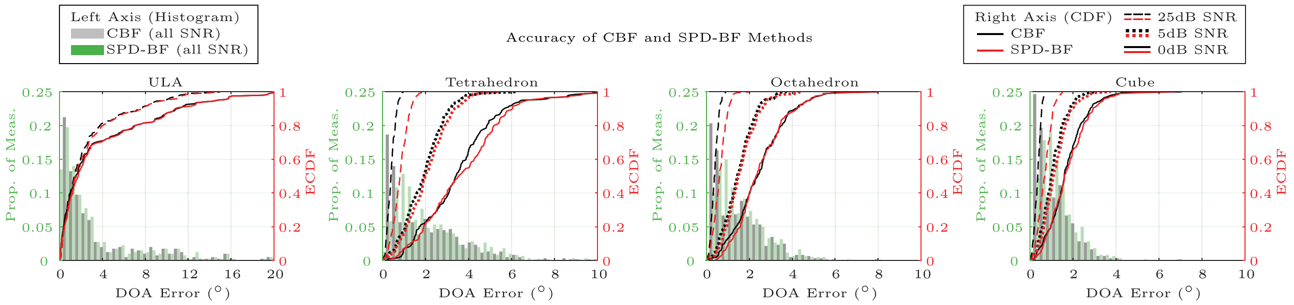

FIG. 7. (Color online) Comparison of accuracy of CBF and SPD-BF—accuracy is calculated as the angular difference between the true DOA and the arg-maximum value from both methods over 200 simulations for varying SNR; error distributions over all SNRs are on the left axis, while empirical CDFs for different SNR levels are shown on the right axis.

$$BW_{3\,\mathrm{dB}} = 2\Big(\min_{\boldsymbol{v}_{3\,\mathrm{dB}}\in S_{3\,\mathrm{dB}}}\big(\arccos(\boldsymbol{v}_{3\,\mathrm{dB}}\bullet\boldsymbol{v}_{true})\big)\Big), \qquad (24)$$

where $S_{3\,\mathrm{dB}}$ is the set of all directions that occur at the $-3\,\mathrm{dB}$ level, and $\boldsymbol{v}_{3\,\mathrm{dB}}$ are the corresponding unit vectors pointing in these directions. The statistics for the estimated beamwidth over 200 simulations with added white Gaussian noise are shown in Fig. 8, with distributions of beamwidths over all SNR levels on the left axis, and CDFs for individual SNRs on the right axis. As expected, the increase in the floor of the SPD-BF beampattern from incoherent summation of sensor pair outputs has increased its resulting half-power beamwidth. The SNR level was found to have a much larger effect on SPD-BF in comparison to CBF—as SNR decreases the beamwidth of SPD-BF increases. This is likely due to more noise signal being added to the floor of the beampattern at low SNR, resulting in a higher floor after summation and a wider beamwidth. This effect is not as evident for the ULA, since its geometry already results in a widely varying beamwidth that changes with steering direction.

D. Speed-up factor

The computation time of CBF and SPD-BF were measured on a standard laptop computer (equipped with 8 GB of memory, and an Intel Core i7–3630QM CPU @ 2.40 GHz × 8 processor). For the three 3D arrays, the beamformer was steered over a full grid of 180 × 360 look directions, with 180 conical angles used for SPD-BF per pair; for the uniform line array, 360 azimuths were used at a single inclination of 90°. Six sensor pairs describe the tetrahedral array; 15 sensor pairs describe the octahedral array, and 28 sensor pairs describe the 8-element line array and cube array. The statistics of the computation time for both approaches are shown as distributions in Fig. 9, with the left and right axes corresponding to the computation time of SPD-BF and CBF respectively. This figure shows that for 3D arrays SPD-BF achieves more than an order of magnitude decrease in computation time versus CBF when evaluating the output over the full grid of look directions. This speed up is due to the fact that CBF must evaluate $180 \times 360 = 64\,800$ look directions, while SPD-BF is evaluated only $6 \times 180 = 1080$ times, $15 \times 180 = 2700$ times, or $28 \times 180 = 5040$ times, for the tetrahedral, octahedral and cube arrays respectively. As expected, SPD-BF is slower than CBF for the ULA, since the beamformer is only steered in a single dimension, negating the efficiency advantages of our method.

E. Direction resolution, accuracy, speed, and memory usage

To better understand the relationship between the resolution of the grid of look directions and the accuracy of the subsequent DOA estimate from both CBF and SPD-BF, the grid resolution was varied, observing the impact on accuracy, computation time, and memory use. Configurations for these simulations are listed in Table III—the 360° azimuth and 180° inclination domains are divided equally into N_ϕ and N_θ angles, respectively, giving a $N_\theta \times N_\phi$ look direction

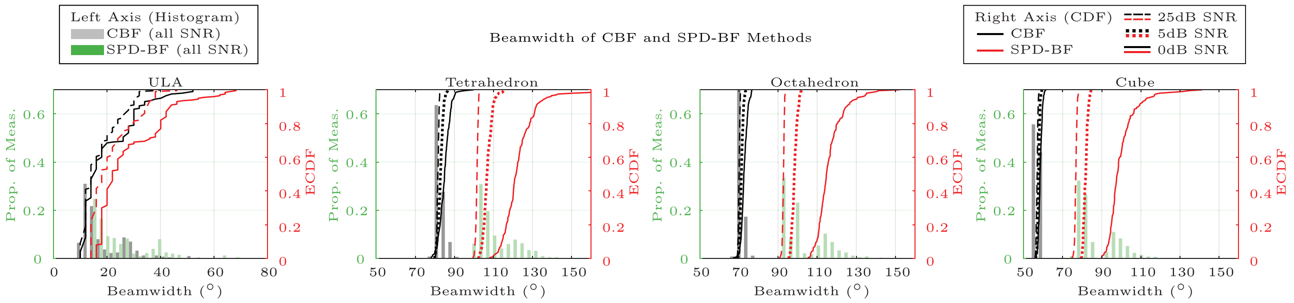

FIG. 8. (Color online) Comparison of the half-power beamwidth of CBF and SPD-BF—beamwidth is calculated as the angular difference between the arg-maximum direction and the nearest –3 dB direction from both methods over 200 simulations for varying SNR; beamwidth distributions over all SNRs are on the left axis, while empirical CDFs for different SNR levels are shown on the right axis.

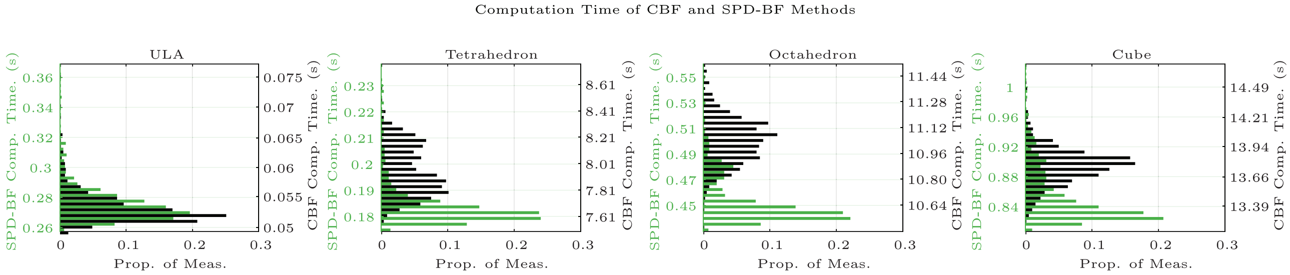

FIG. 9. (Color online) Comparison of the computation times of CBF and SPD-BF—computation time is measured on a standard laptop computer using both methods over 200 simulations for varying SNR; distributions of computation time per simulation are shown on the left and right axes for SPD-BF and CBF respectively. Note that the left and right axes have different ranges.

grid; the number of conical angles in the 180° conical angle space is set equal to the number of azimuthal angles. Section IV B noted that conical angle resolution significantly impacts accuracy at high SNR—the number of conical angles is doubled here in comparison to Sec. IV B, so as to reduce the accuracy gap between CBF and SPD-BF, since improvements in speed and memory use for a given level of accuracy is the primary interest here.

The LFM chirp described previously is corrupted by white Gaussian noise to achieve random SNRs anywhere between 0 and 25 dB. True DOA was selected randomly from the set of look directions in Table III for a given configuration. For each configuration listed, 200 simulations were completed for both beamforming methods, using the regular tetrahedral, octahedral and cube array geometries, with the root-mean-square of the angular errors (RMSE) calculated over all simulations. Mean computation time was also calculated. The plot of Fig. 10 shows the RMSE for both CBF and SPD-BF, and illustrates that these configurations achieve a similar level of accuracy between both methods, allowing computational and memory efficiency to be fairly compared between both.

Figure 11 plots the computation time averaged over all runs for each of the configurations for both methods: as expected SPD-BF is significantly more computationally efficient for the same accuracy because as the direction resolution increases, the computation time for CBF increases quadratically, while for SPD-BF the computation time increases linearly. The speedup factor for computing the various grid configurations is plotted in Fig. 12: SPD-BF is able to speed up the computation time of a 180 × 360 grid of directions by more than 10 times with negligible loss in accuracy.

TABLE III. Look-angle configurations for comparative analysis of speed and memory usage.

Configuration	S_{angles} $(N_\theta \times N_\phi)$	S_{conical} (N_ϕ)
1	$23 \times 45 = 1035$ look directions	45 conical angles
2	$45 \times 90 = 4050$ look directions	90 conical angles
3	$68 \times 135 = 9180$ look directions	135 conical angles
4	$90 \times 180 = 16200$ look directions	180 conical angles
5	$113 \times 225 = 25425$ look directions	225 conical angles
6	$135 \times 270 = 36450$ look directions	270 conical angles
7	$158 \times 315 = 49770$ look directions	315 conical angles
8	$180 \times 360 = 64800$ look directions	360 conical angles

Memory use was also evaluated for each beamforming approach. For CBF, the phase shifts over all M frequencies for each look direction in the grid and for all N elements in the array must be stored—this requires $N_\theta \times N_\phi \times M \times N$ complex doubles. In comparison, SPD-BF requires the storage of the phase shifts over all M frequencies for all conical angles belonging to all *unique* sensor pairs. For the regular tetrahedral array, there is only one unique pair, since all sensors have the same separation distance; for the regular octahedral array, there are two unique pairs; and for the regular cube array, there are three unique pairs. As such, our method requires a much smaller amount of memory, equal to storing $N_\phi \times M \times P_{unique}$ complex doubles, where P_{unique} is the number of unique sensor pairs. The plot of Fig. 13 illustrates the amount of memory required by both methods using $M = 512$. This figure illustrates the remarkable memory savings of SPD-BF, with the memory required being more than *two orders of magnitude less* than CBF for a 180 × 360 grid of look directions. This means SPD-BF provides a significant increase in direction resolution as compared to CBF for a given amount of memory.

F. Issues and limitations in array gain and array size

A well-known fact of the CBF is that its gain increases linearly with the number of elements, since it forms a coherent sum of the signal over all N elements, resulting in a $10 \log (N)$ dB SNR. This result is clearly shown in the top plot of Fig. 14, which illustrates simulated responses of both CBF and SPD-BF with a ULA of increasing size, using a single narrowband source at 90° and in the absence of

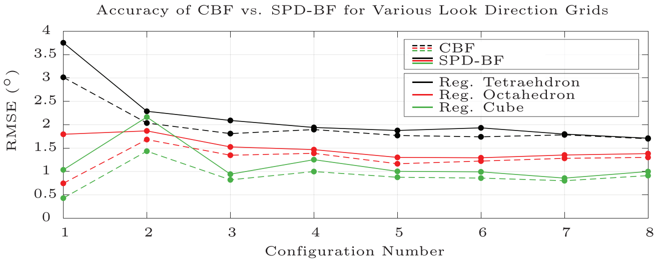

FIG. 10. (Color online) Root-mean-square of DOA estimate errors for CBF and SPD-BF with varying direction resolution from Table III.

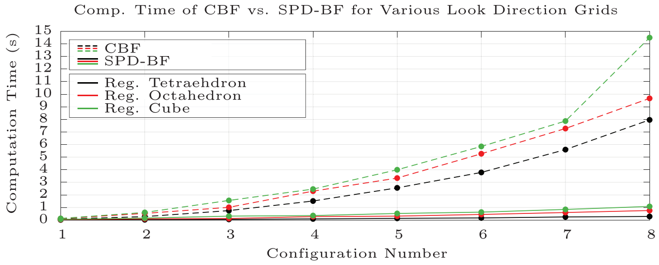

FIG. 11. (Color online) Computation time of CBF and SPD-BF with varying direction resolution as listed in Table III.

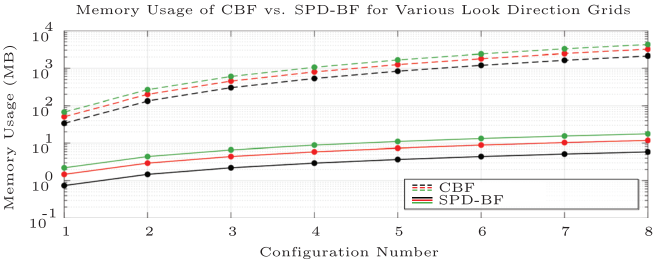

FIG. 13. (Color online) Memory usage of CBF and SPD-BF with varying direction resolution as listed in Table III.

noise—the main lobe of the CBF amplitude response has a uniform dB increase as the number of elements in the array is doubled. In contrast, the SPD-BF response shown in the lower plot of Fig. 14 illustrates that its gain is roughly fixed regardless of the number of elements in the array—in fact, the floor of its amplitude response appears to increase up to a limit as the array grows, suggesting a limited *decrease* in gain. This result is expected, since unlike CBF where a common origin is used to align and coherently sum the signal from all elements, SPD-BF uses a separate origin for each sensor pair, limiting coherent array gain to $10\log(2) = 3$ dB, followed by the incoherent summation of the magnitude outputs of each pair. This significant limitation in gain suggests that SPD-BF can only be used to detect strong sources, whose levels are within approximately 3 dB of each other. However, it is interesting to note from Fig. 14 that the main lobe for both CBF and SPD-BF experiences a similar rate of decrease in beamwidth with increasing array size. Thus, the increased beamwidths of SPD-BF obtained in Sec. IV C and Fig. 8 are primarily affected by its fixed gain, and may not accurately represent its ability to resolve multiple sources, as long as these sources are of similar power.

As elements are added to an array, the number of pair combinations grows with number of elements N,

$$P = \binom{N}{2} = \frac{N!}{2!(N-2)!} = \frac{N(N-1)}{2}. \qquad (25)$$

This factorial growth means that SPD-BF is most valuable when the number of array sensor pairs is small in comparison to the desired resolution of the output, i.e., $PC \ll C^2$. This limitation is illustrated in the approximated theoretical cost map of Fig. 15; the computational cost over a grid of look directions for CBF is roughly estimated as $2C^2$ (where C is the number of angles), while that of SPD-BF is estimated as PC—this figure illustrates that, for example, using a grid of 100×200 look directions in inclination and azimuth, respectively, SPD-BF will computationally outperform CBF for arrays with 20 elements or fewer. A feature of SPD-BF that we term *anytime stopping*, illustrated in Fig. 16, further extends the applicability of this technique to larger arrays. In anytime stopping, a subset of sensor pairs may be used to reduce computation cost in exchange for accuracy. Figure 16 shows that SPD-BF is able to estimate the DOA to within 1° after the summation of only 3 of the 6 sensor pairs in a regular tetrahedral array.

Another issue with SPD-BF is the use of nearest neighbor summation of sensor pair conical angles to construct the full output: full output accuracy is dependent on the number of conical angles used for each pair, the geometry of the array, and the set of desired look directions. Figure 16 illustrates the effect of the level of discretization of the conical angle space on SPD-BF accuracy—accuracy of SPD-BF improves in a non-trivial manner with the increase in the number of conical angles used.

The aforementioned use of separate origins or phase centers for each sensor pair in SPD-BF results in fixed coherent gain, and an increase in the floor of its response as more pairs are added—this is a result of the fact that the CBF magnitude response of individual sensor pairs is

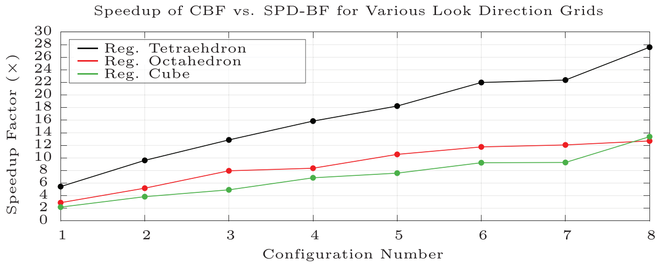

FIG. 12. (Color online) Speedup of SPD-BF over CBF with varying direction resolution as listed in Table III.

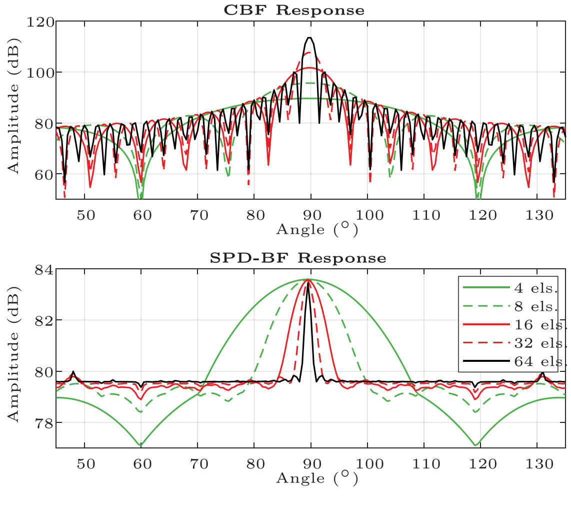

FIG. 14. (Color online) CBF and SPD-BF responses of a uniform line array with varying number of elements to a narrowband signal arriving from 90°, illustrating linear increase in gain of CBF, and fixed gain of SPD-BF.

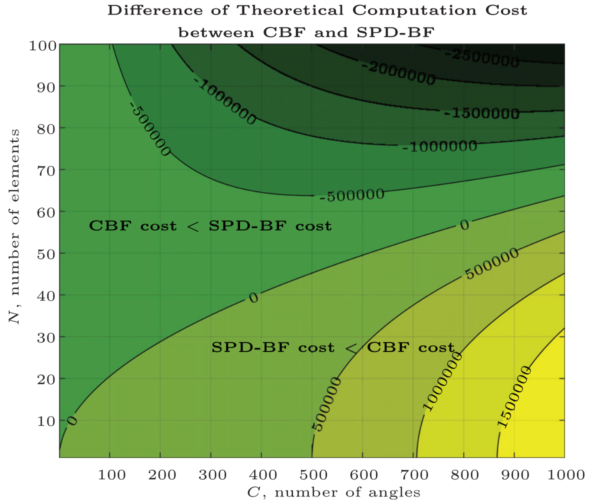

FIG. 15. (Color online) Approximated theoretical difference in computational cost of CBF and SPD-BF as a function of number of elements and number of angles for 3D beamforming; areas greater than 0 indicate that SPD-BF is less costly than CBF.

always greater than zero, raising the floor via incoherent summation, and is clearly visible in Fig. 14. To better understand the behavior of SPD-BF stemming from this incoherent summation, we simulate two narrowband sources at 40° and 100°, with the second source at a lower power than the first. The responses in the absence of noise of a ULA with increasing size for both CBF and SPD-BF are shown in Fig. 17. The CBF responses clearly indicate, again, the increase in gain with number of elements—for the 4, 8, 16, and 32 element ULAs, we expect gains of $10\log(N) \approx 6$, 9, 12, and 15 dB, respectively, and we clearly see that this is roughly the case: with 4 elements the second source merges with the sidelobe below –9 dB, and with 8 elements the second source is around the same level as the first sidelobe at –12 dB; the 16 and 32 element ULAs have a first sidelobe level of about –12 and –15 dB, respectively; thus, with the addition of more elements, the CBF is able to distinguish a second source of increasingly reduced power thanks to the linear increase in gain. Additionally, the amplitude responses of CBF accurately reflect the relative power of the two sources—the difference in peaks between the main lobe and the second source lobe is roughly equal to the reduction in power of the second source, something we see is not the case for the SPD-BF response. Looking at the SPD-BF responses in the lower plots of Fig. 17, we again see that the gain is essentially fixed, evidenced by the similar dynamic range across ULAs of different size. Interestingly, however, the second source is effectively visible in all arrays at a power level of –6 dB, and arguably even at –9 dB. Thus, even though the expectation is that individual sensor pairs would be unable to discern a secondary source 3 dB lower than the primary source, the incoherent summation of CBF outputs of all pairs together provide enough gain so as to enable the SPD-BF to discern a secondary source up to 9 dB lower. Incoherent summation across sensor pairs produces this surprising result, and indicates that the behavior of SPD-BF cannot be explained adequately within the classical beamforming framework—nulls and sidelobes from different sensor pair outputs combine and negate each other, leading to a very flat floor, which does not reflect the relative gain loss away from the main lobe. This is very clearly demonstrated in the relative levels of the main and second source lobes, as well as the level of the second source lobe as its power is decreased as shown in Fig. 17—the 3 dB step-changes in power of the second source are not accurately reflected in the SPD-BF responses. It is for this reason that we describe SPD-BF responses as having a floor rather than sidelobes, and why we do not usually display them on a dB scale. In addition, the standard rule used to determine an array's ability to resolve a secondary source, that in which the second source must be at least a half-power beamwidth away from the main source, is not applicable to SPD-BF—for example, the SPD-BF response of the 32 element ULA with a second source at –3 dB has a dynamic range less than 3 dB (indicating infinite beamwidth); however, the second source peak is clearly visible. The empirical results illustrated by these simulations demonstrate the strict limitations of SPD-BF, and indicate the need for in-depth theoretical analysis of its behavior, an analysis that is out of the scope of the current paper.

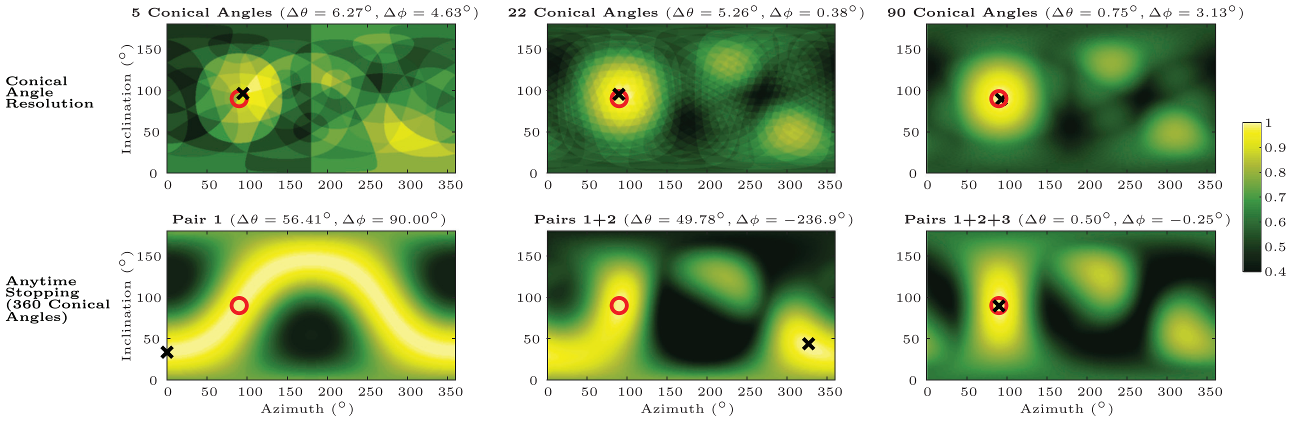

FIG. 16. (Color online) Top: the effect of conical angle resolution on SPD-BF output—the use of nearest neighbor summation means that the number of conical angles used significantly impacts SPD-BF output quality. Bottom: SPD-BF implicitly features anytime stopping by only summing subsets of sensor pairs; this allows computation time to be capped. True DOA is shown as the red circle at θ, $\phi = 90°$, and the maximum output of SPD-BF is shown as the black cross.

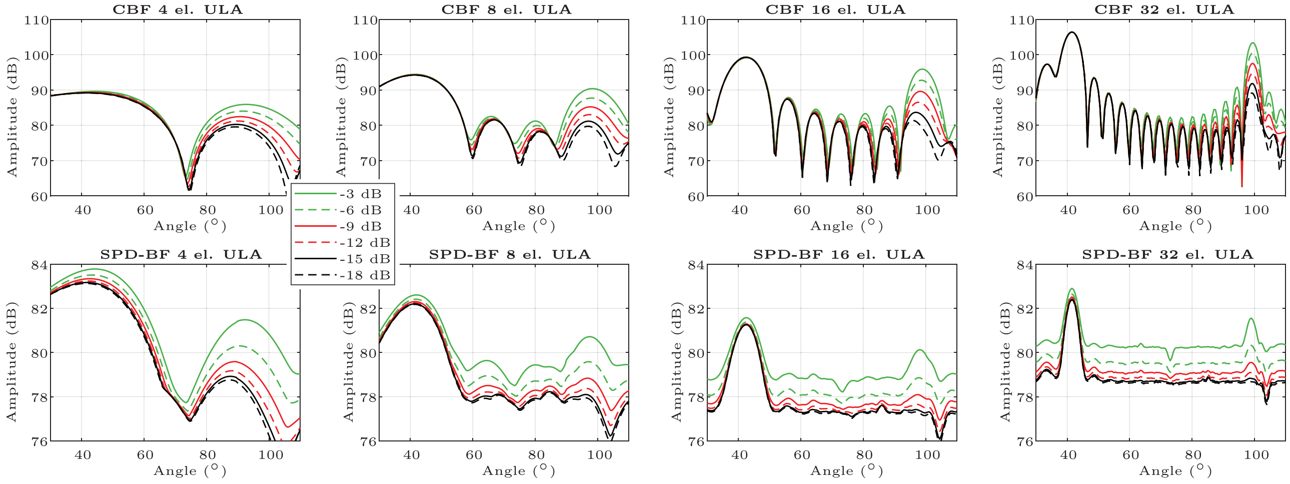

FIG. 17. (Color online) CBF and SPD-BF responses of a uniform line array with varying number of elements to two narrowband sources—a source at 40° and a second source of lower and varying power at 100°. Top: CBF response accurately reflects the relative power of both sources, while the SPD-BF response (bottom) does not.

The two most significant limitations of SPD-BF of an essentially fixed array gain and the restriction of applicability to small arrays, means that this method is not well-suited to the application of resolving or tracking multiple acoustic sources—a single source of much higher sound pressure level would effectively drown out other sources, and even if that were not the case, the wider beamwidths associated with smaller arrays would limit angular resolution. Thus, SPD-BF finds its greatest utility in tracking a single strong source, which we illustrate using experimental data in Sec. V.

V. EXPERIMENTAL COMPARISONS

A. Setup

Acoustic data were gathered by the WAM-V autonomous surface vehicle (ASV)[20] shown in Fig. 18, equipped with an 8-cm pyramidal hydrophone array as described in Ref. 21. The ASV is a 5 m long, differential drive platform, outfitted with a Hemisphere V102 DGPS receiver for navigation, providing an accuracy in positioning of 1 m or less and heading of 0.75°. The pyramidal array consists of five HTI-96-Min hydrophones with 8 cm edge spacing. The array is attached to the end of a 1.5 m port-side aluminum boom using a 3D-printed mount. Acoustic energy captured by this array is digitized using a USB-1608FS-Plus data acquisition (DAQ) device. The custom acoustic beacon for transmission consists of a GPS receiver, Arduino Uno, Wave Shield and Lubell 3400 underwater speaker, allowing transmission of user-defined signals.

An acoustic data set was collected by the ASV running on the Charles River by the MIT Sailing Pavilion in November 2017. The acoustic beacon was attached to the Pavilion dock at a 1 m depth, and set to transmit a 11–9 kHz, 20 ms LFM chirp every second as triggered by the pulse-per-second (PPS) signal from its GPS. The ASV drove the lawnmower track shown in Fig. 19, recording 16 000 acoustic samples every second from each hydrophone at 37.5 kS/s. Sampling was performed in sync with beacon

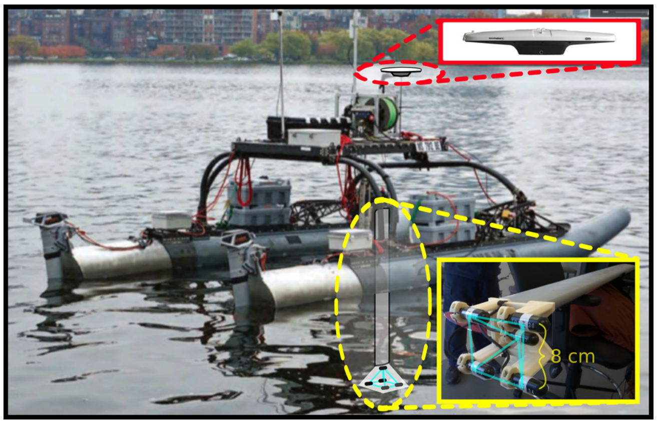

FIG. 18. (Color online) The WAM-V ASV equipped with 8-cm pyramidal hydrophone array (inset yellow), and Hemisphere V102 DGPS receiver (inset red).

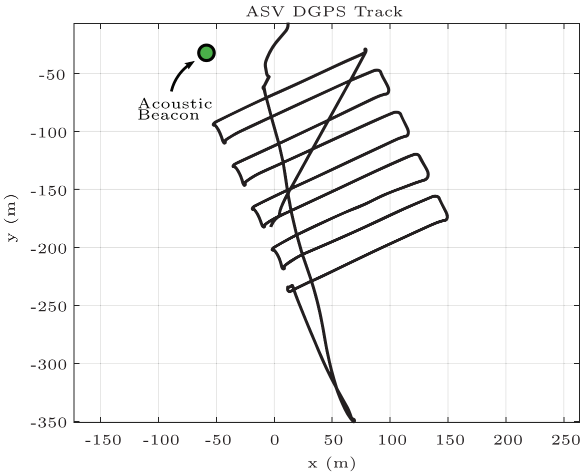

FIG. 19. (Color online) ASV DGPS track in a local coordinate frame in black, with beacon position as the green circle.

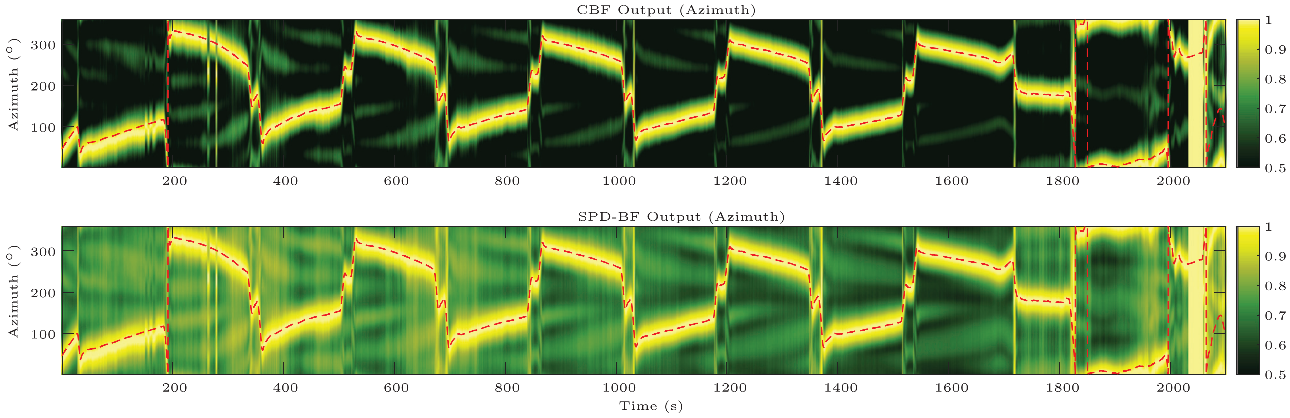

FIG. 20. (Color online) Outputs from CBF and SPD-BF using experimental acoustic data gathered by an ASV—outputs over azimuth are shown by selecting the azimuthal slice corresponding to the inclination along which the maximum occurs; red dashed line is the true DOA in azimuth calculated using DGPS position and heading.

transmission via triggering of the DAQ at the start of each second using GPS PPS.

B. Results

The collected acoustic data were processed using CBF and SPD-BF using configuration 8 in Table III. The data were pre-processed by normalizing each sample by the magnitude of its Fourier transform (the so-called phase-transform), then transformed into the frequency domain using the chirp Z-transform (CZT) with 2048 frequency components—this allows for the efficient time-to-frequency transformation of the signal, constraining the frequency domain to the subset of bins containing only the relevant frequencies between 9 and 11 kHz.

The plots in Fig. 20 illustrate the outputs from CBF at top, and that from SPD-BF at bottom, over the duration of the entire experiment. These plots show the outputs over azimuth (ϕ) only, with the azimuthal slice selected across the arg-maximum value of inclination (θ). The true DOA in azimuth is shown as the dashed red line, calculated using the position and heading recorded by the ASV's DGPS receiver. It is apparent from these plots that SPD-BF produces a very similar output to CBF. Qualitatively, the maximum for both methods follows the true azimuth well, indicating that both methods are able to track the acoustic source. As expected from simulation, the summation of the outputs of sensor pairs has resulted in a higher floor in the output of the SPD-BF method, clearly seen as the major difference between the two plots.

Taking the arg-maximum azimuths from the outputs of both methods, and comparing them to the true azimuths calculated using DGPS measurements, allows the probability distribution plot of azimuthal error in Fig. 21 to be generated. This figure shows that a major percentage of errors in azimuth are $<15°$, with about 92% of measurements from both CBF and SPD-BF falling into this category. Approximately 80% of measurements from both methods have an error of $<8°$. What is clearly apparent is that both CBF and SPD-BF have a comparable level of accuracy, as the profile of the error distributions for both methods are very similar. There is a 0° or 1° difference between the CBF and SPD-BF DOA estimates in 90% of the measurements, illustrating their similarity in output. The close match of results between techniques on real data, coupled with the computation and memory gains, shows the utility of SPD-BF for efficient DOA estimation of a single strong source.

VI. CONCLUSION

This paper describes a novel approximate beamforming method that is computationally and memory efficient for arrays with a small number of sensors, termed sensor pair decomposition beamforming (SPD-BF). The key insight of this method is that any given look direction for an arbitrary 3D array can be described in the conical angle space of the array's sensor pairs by a single conical angle rather than the usual two angles in azimuth and inclination. This insight allows the search space of a grid of look directions to be collapsed from two dimensions to one—rather than beamforming at every look direction on a grid, SPD-BF instead beamforms at every conical angle of each sensor pair, and constructs an approximation from these outputs on the original grid of look directions. This results in significant

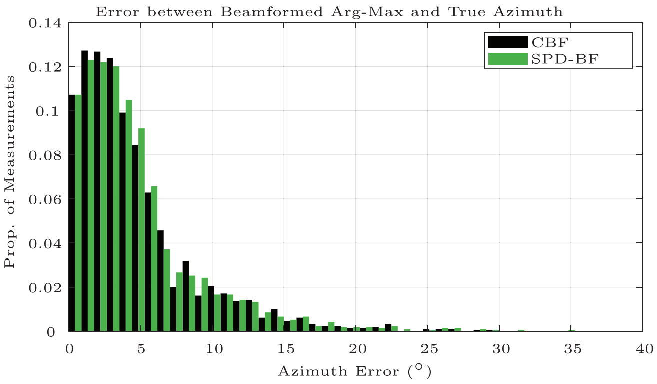

FIG. 21. (Color online) Probability distribution of error between true azimuth calculated from DGPS position/heading and the arg-maximum outputs from CBF and SPD-BF.

improvements in memory and computation efficiency for similar direction resolution and accuracy in 3D beamforming on small arrays. Results from simulations illustrated the advantages and limitations of SPD-BF in comparison to conventional beamforming (CBF), with the main advantage being an order of magnitude improvement in computation time and memory use in certain use-cases, and the main limitation being a fixed gain. Experimental results using acoustic data gathered by an autonomous surface vehicle (ASV) equipped with a five-element pyramidal hydrophone array demonstrated the utility of SPD-BF in processing sensor data more efficiently and at a similar level of accuracy as CBF for direction-of-arrival (DOA) estimation of a single strong source. These results indicate that SPD-BF may be particularly useful for high-resolution, real-time acoustic localization on low-power embedded systems, where computation and/or memory may be limited and where the memory usage of CBF may overwhelm available resources. For example, wideband CBF using a four-element array on a 180 × 360 grid of look directions with 512 FFT bins requires 180 × 360 × 512 × 4 16 byte complex doubles, equivalent to approximately 4.2 GB—this amount of memory is far beyond that available on a BeagleBone Black or a Raspberry Pi 3 embedded computer; in contrast, with 360 conical angles, SPD-BF requires only 360 × 512 × 4 × 16 bytes, or about 11.8 MB.

Future work may examine how the unique features of SPD-BF can be used to increase computational efficiency in specific applications. For example, SPD-BF with a low conical angle resolution and a small subset of an array's sensor pairs can be used to very quickly discard erroneous directions for DOA estimation; this can then be followed by an iterative refinement of the estimated DOA via an increase in the number of conical angles, or the addition of additional sensor pairs, to iteratively converge on the true DOA. Alternatively, once SPD-BF has been used to provide a quick and rough estimate of the true DOA, CBF can be performed over a smaller subset of look directions. The incorporation of other, more precise beamforming methods may be another direction of future work. By using conical angles to reduce the cost of computing/storing steering vectors while integrating weighting concepts from MVDR or MUSIC, an improvement in angular resolution and gain of SPD-BF may be possible while maintaining some of the benefits in speed and memory efficiency afforded by our method.

ACKNOWLEDGMENTS

The authors offer their sincere thanks to A. Baggeroer for feedback on this method, P. V. Teixeira for discussions, as well as to M. Sacarny and M. DeFilippo for assistance in collecting experimental data. Thanks are due to the anonymous reviewers for their critical comments. This work was partially supported by the Office of Naval Research, the Defense Advanced Research Projects Agency, and Lincoln Laboratory.

[1] E. Brookner, "Phased-array radar," Sci. Am. **252**, 94–102 (1985).
[2] H. L. Van Trees, *Optimum Array Processing* (Wiley, New York, 2002), pp. 17–89.
[3] J. Litva and T. K. Lo, *Digital Beamforming in Wireless Communications* (Artech House, Boston, 1996), pp. 93–114.
[4] T. L. Szabo, *Diagnostic Ultrasound Imaging: Inside Out* (Academic Press, New York, 2004), pp. 209–255.
[5] W. C. Knight, R. G. Pridham, and S. M. Kay, "Digital signal processing for sonar," Proc. IEEE **69**(11), 1451–1506 (1981).
[6] J. C. Chen, K. Yao, and R. E. Hudson, "Source localization and beamforming," IEEE Sign. Process. Mag. **19**(2), 30–39 (2002).
[7] J. Benesty, J. Chen, and Y. Huang, "Microphone array signal processing," Springer Top. Sign. Process. **1**, 39–65 (2008).
[8] J. H. Justice, N. L. Owsley, J. L. Yen, and A. C. Kak, *Array Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ, 1985), pp. 433.
[9] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," IEEE Proc. **57**(8), 1408–1418 (1969).
[10] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," IEEE Trans. Ant. Propag. **34**(3), 276–280 (1986).
[11] A. C. Gurbuz, J. H. McClellan, and V. Cevher, "A compressive beamforming method," in *IEEE International Conference on Acoustics, Speech and Signal Processing* (2008), pp. 2617–2620.
[12] G. F. Edelmann and C. F. Gaumond, "Beamforming using compressive sensing," J. Acoust. Soc. Am. **130**(4), EL232–EL237 (2011).
[13] A. Xenaki, P. Gerstoft, and K. Mosegaard, "Compressive beamforming," J. Acoust. Soc. Am. **136**(1), 260–271 (2014).
[14] R. Mucci, "A comparison of efficient beamforming algorithms," IEEE Trans. Acoust. Speech Sign. Process. **32**(3), 548–558 (1984).
[15] J. R. Williams, "Fast beam-forming algorithm," J. Acoust. Soc. Am. **44**(5), 1454–1455 (1968).
[16] D. R. Farrier, T. S. Durrani, and J. M. Nightingale, "Fast beamforming techniques for circular arrays," J. Acoust. Soc. Am. **58**(4), 920–922 (1975).
[17] V. Murino and A. Trucco, "Underwater 3D imaging by FFT dynamic focusing beamforming," in *IEEE Proceedings of 1st International Conference on Image Processing* (1994), pp. 890–894.
[18] M. Palmese and A. Trucco, "Three-dimensional acoustic imaging by chirp zeta transform digital beamforming," IEEE Trans. Instrum. Meas. **58**(7), 2080–2086 (2009).
[19] C. Chi, Z. Li, and Q. Li, "Fast broadband beamforming using nonuniform fast Fourier transform for underwater real-time 3-D acoustical imaging," IEEE J. Ocean. Eng. **41**(2), 249–261 (2016).
[20] A. Anderson, E. Fischell, T. Howe, T. Miller, A. Parrales-Salinas, N. Rypkema, D. Barrett, M. Benjamin, A. Brennen, M. DeFillipo, J. J. Leonard, L. Paull, H. Schmidt, N. Wang, and A. Yaari, "An overview of MIT-Olin's approach in the AUVSI RobotX Competition," in *Field and Service Robotics*, Vol. 113 of *Springer Tracts in Advanced Robotics* (Springer, Berlin, 2016), pp. 61–80.
[21] N. R. Rypkema and H. Schmidt, "Passive inverted ultra-short baseline (piUSBL) localization: An experimental evaluation of accuracy," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2019), pp. 7197–7204.