Digital Signal Processing 69 (2017) 22–31

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

DOA estimation based on the difference and sum coarray for coprime arrays

Xinghua Wang, Zhenhong Chen, Shiwei Ren, Shan Cao *

School of Information and Electronics, Beijing Institute of Technology, 5 South Zhongguancun Street, Haidian District, Beijing, 100081, China

ARTICLE INFO

Article history:
Available online 20 June 2017

Keywords:
Array signal processing
Coprime array
DOA estimation
Virtual array

ABSTRACT

In this paper, we construct a novel coarray named as the difference and sum (diff–sum) coarray by exploiting an improved Conjugate Augmented MUSIC (CAM) estimator, which utilizes both the temporal information and the spatial information. The diff–sum coarray is the union of the difference coarray and the sum coarray. When taking the coprime array as the array model, we find that the elements of the sum coarray can fill up all the holes in the difference coarray. Besides, the sum coarray contains bonus uniform linear array (ULA) segments which extend the consecutive range of the difference coarray. As a result, the consecutive lags of the diff–sum coarray are much more than those of the difference coarray. For analysis, we derive the hole locations and consecutive ranges of the difference set and the sum set, discuss the complementarity of the two sets, and provide the analytical expression of the diff–sum virtual aperture. Simulations verify the effectivity of the improved method and show the high DOF of the diff–sum coarray.

1. Introduction

Direction-of-arrival (DOA) estimation of multiple narrowband signals is a major application of the antenna array [1–7]. In the past few decades, many high-resolution subspace-based methods have been proposed for direction finding, such as MUSIC [2], ESPRIT [8] etc. By using these conventional methods, the number of detectable sources impinging on a uniform linear array (ULA) with N sensors is at most $N-1$ [9]. Detecting more sources than the number of sensors has been more and more attractive in recent years [10]. Constructing sparse arrays based on the concept of the coarray is a useful and important way to increase the degrees of freedom (DOF). In [11], the minimum redundancy array (MRA) was introduced for this purpose. It is a kind of sparse array whose difference coarray has no holes in it. However, there is no closed form expression for the MRA. As a result, the DOF for a given number of sensors can not be obtained as well.

To resolve these problems, several sparse arrays have been proposed recently. In [12], a nested array, which can detect at most $(N^2+2N)/4-1$ (N is even) or $(N+1)^2/4-1$ (N is odd) sources with only N sensors, was proposed. This structure is obtained by systematically nesting two ULAs. Another sparse array named the coprime array was proposed in [13]. Two uniform linear subarrays, which share the first sensor, are used to form the sparse array. One is M sensors with spacing N units, and the other is N sensors with spacing M units, where integers M and N are coprime. Coprime arrays can detect as many as $O(MN)$ sources using only $M+N-1$ sensors. However, compared to a nested array, a coprime array requires more sensors to achieve the same DOF. Thus, how to increase the DOF of the coprime array has generated a new wave of interest. An extended coprime array which contains $M+2N-1$ sensors was proposed in [14]. Its difference coarray lags can reach consecutive integers from $-MN-N+1$ to $MN+N-1$, which has been somewhat extended with more sensors used. Authors in [15] proposed the generalized coprime array concept with two operations. The first operation is compressing the inter-sensor spacing of one subarray in the coprime array, which yields a coprime array with compressed inter-element spacing (CACIS). The second operation is displacing one subarray in the coprime array, which forms a coprime array with displaced subarrays (CADiS). Both the CACIS and CADiS can achieve a high number of DOF. In addition to optimizing the design of the coprime array configuration, several novel methods have also been proposed to increase the DOF of the coprime array. In [16], a sparsity enforced recovery technique for the coprime array was proposed. Taking the off-grid DOAs into account, the method can construct a larger difference coarray than that in [14] by utilizing the grid offset vector. However, the closed form expression for the DOF has not been summarized. In [17], a novel coarray interpolation method for the coprime array was

* Corresponding author.
E-mail addresses: 89811@bit.edu.cn (X. Wang), chenzhenhongbit@163.com (Z. Chen), renshiwei@bit.edu.cn (S. Ren), caoshan@bit.edu.cn (S. Cao).

http://dx.doi.org/10.1016/j.dsp.2017.06.013
1051-2004/

proposed. This method utilizes nuclear norm minimization to interpolate the missing samples or holes and is capable of achieving a high number of DOF. However, the actual freedom is still governed by the number of unique virtual sensors in the nonuniform difference coarray. Although all the geometries and methods mentioned above can correctly estimate more sources than the number of sensors, only the difference coarray is considered. The sum coarray cannot be utilized effectively.

Utilizing the difference and sum coarrays jointly to perform DOA estimation can further increase the DOF. The sum coarray usually arises as the virtual array in active sensing. With the transmitters illuminating the field of view and the receivers detecting the reflections from the targets, the received data can be considered as observations from the sum coarray. A sparsity-based method using active nonuniform arrays was proposed in [18] for extended aperture with both sum and difference coarrays. The vectorized covariance matrix of the sum coarray observations emulates the equivalent signals received from the difference coarray of the sum coarray. But one weakness of the method is that the sum coarray cannot perform sufficiently due to different transmitter and receiver arrays. In [19], a modified minimum redundancy monostatic multiple-input multiple-output (MIMO) configuration was proposed. As the transmitter and receiver arrays are identical, both the difference coarray and sum coarray can be used sufficiently. However, these methods constructing the sum coarray are restricted to active arrays. Therefore, proposing a novel method which can utilize the passive arrays to construct the difference and sum coarrays is significant.

In this paper, we exploit an improved Conjugate Augmented MUSIC (CAM) estimator [20], which we name as Vectorized Conjugate Augmented MUSIC (VCAM), to perform DOA estimation using passive arrays. By utilizing both the temporal information and the spatial information, we construct a conjugate augmented correlation vector based on the second-order statistics. Instead of calculating the fourth-order (FO) cumulants in [20], we vectorize the covariance matrix of the conjugate augmented correlation vector and get an equivalent received signal using the concept of Khatri–Rao (KR) product. The resulting coarray, which is named as the difference and sum (diff–sum) coarray in this paper, comprises not only the difference set but also the sum set. For the coprime array, it is at last verified that the elements of the sum coarray can fill up all the holes in the difference coarray. Besides, the sum coarray contains bonus ULA segments outside the consecutive range of the difference coarray. As a result, the virtual array can achieve much higher DOF than the difference coarray obtained by vectorized MUSIC [12]. Furthermore, its virtual aperture is more than twice the physical aperture, which will contribute to the decrease of the array size. In particular, we analyze the performance of the consisting sets of the diff–sum coarray and discuss the relationship between them. Based on the complementarity of the difference set and sum set, we give the expressions of the diff–sum coarray aperture for quantitative evaluation. Simulations verify the effectivity of the improved method.

The rest of the paper is organized as follows. In section 2, we first review the data model and the coprime array configuration. In section 3, we present the VCAM algorithm to acquire the diff–sum coarray. Then, in section 4, we derive the properties of the consisting difference coarray, the consisting sum coarray and the final diff–sum virtual array for the coprime geometry. Simulation results are provided in section 5 to numerically compare the performance of the nested array, the CACIS and the CADiS with vectorized MUSIC and the coprime array with VCAM method. Section 6 concludes the paper.

Notations. In this paper, vectors are denoted by italic boldface lowercase letters, e.g., $\boldsymbol{a}$. Matrices are denoted by italic boldface capital letters, e.g., $\boldsymbol{A}$. $(.)^H$ denotes conjugate transpose, whereas

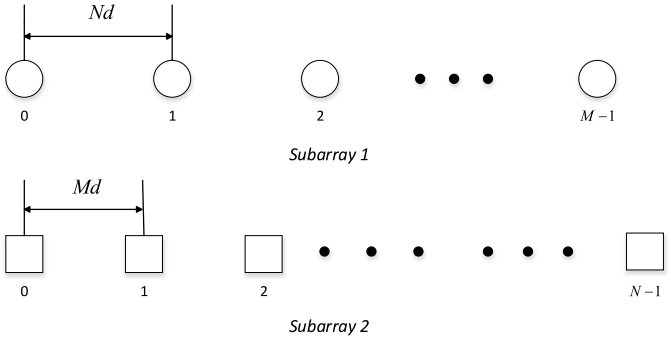

Fig. 1. The prototype coprime array.

$(.)^T$ and $(.)^*$ respectively denote transpose and conjugation. $vec(.)$ is used to denote vectorizing operation and the symbol $\otimes$ denotes the left Kronecker product.

2. System model

Denote $\boldsymbol{d} = \{d_1, \ldots, d_P\}$ as the positions of the array sensors with the first sensor as the reference, i.e., $d_1 = 0$. K uncorrelated narrowband plane wave sources impinge on the array from directions $\{\theta_1, \ldots, \theta_K\}$ with powers $\{\sigma_i^2, i = 1, 2, \ldots, K\}$. According to [20], we denote the ith source signal as $s_i(t) = u_i e^{j(\omega_c + \omega_i)t}$, where ω_c is the carrier frequency, u_i is the deterministic complex amplitude and ω_i is a small frequency offset. For different source signals, the frequency offsets are respectively different. After demodulation to IF, the ith signal becomes $s_i(t) = u_i e^{j\omega_i t}$. The additive noise is assumed to be white Gaussian with zero mean and variance σ_n^2, which is uncorrelated with the sources. Denote λ as the wavelength of the carrier. Then, the received signal can be expressed as

$$\boldsymbol{x}(t) = \sum_{k=1}^{K} \boldsymbol{a}(\theta_k) s_k(t) + \boldsymbol{n}(t) = \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{n}(t), \tag{1}$$

where $\boldsymbol{A} = [\boldsymbol{a}(\theta_1), \boldsymbol{a}(\theta_2), \ldots, \boldsymbol{a}(\theta_K)]$ with

$$\boldsymbol{a}(\theta_k) = [1, e^{j2\pi d_2 \sin(\theta_k)/\lambda}, \ldots, e^{j2\pi d_P \sin(\theta_k)/\lambda}]^T. \tag{2}$$

$\boldsymbol{s}(t) = [s_1(t), s_2(t), \ldots, s_K(t)]^T$ and $\boldsymbol{n}(t) = [n_1(t), n_2(t), \ldots, n_P(t)]^T$ are source vector and noise vector, respectively. Since the element in the ith row and jth column of the covariance matrix

$$\boldsymbol{R}_{xx} = E[\boldsymbol{x}(t)\boldsymbol{x}^H(t)] \tag{3}$$

corresponds to the equivalent signal received from the sensor located at $d_i - d_j$, vectorizing $\boldsymbol{R}_{xx}$ can obtain the equivalent received signal at the difference coarray. It's noted that, in practice, the covariance matrix $\boldsymbol{R}_{xx}$ is estimated from a finite (say, N_p) snapshots as

$$\overline{\boldsymbol{R}}_{xx} = \frac{1}{N_p} \sum_{a=1}^{N_p} \boldsymbol{x}(a\Delta t)\boldsymbol{x}^H(a\Delta t),$$

where Δt is the sampling interval for snapshots. Δt is set to satisfy the sampling theory.

The coprime array, as shown in Fig. 1, is the union of two uniform linear subarrays [13]. One is of M sensors with the inter-sensor spacing of N units. The other is of N sensors with the inter-sensor spacing of M units. M and N are coprime integers. Assume that $M < N$. Then, the number of sensors in the coprime array is $P = M + N - 1$. Denote d as the unit inter-element spacing. The array sensors are located at

$$L_{prototype} = \{Nmd \,|\, 0 \le m \le M-1\} \cup \{Mnd \,|\, 0 \le n \le N-1\}. \tag{4}$$

The difference coarray has virtual sensors at

$$L'_{diff} = \{d_i - d_j \,|d_i\ , d_j \in L_{prototype}, 1 \le i, j \le P\}. \tag{5}$$

For convenience, we ignore the inter-element spacing d. Then, the difference set can be simplified as

$$L_{diff} = \{l_{d_i} \,|l_{d_i} d \in L'_{diff}\}. \tag{6}$$

The maximum DOF of the difference coarray is determined by the number of distinct elements in L_{diff}. As mentioned in [15], the following relationship exists

$$L_{diff} = L_{sd} \cup L_{sd}^{-} \cup L_{cd} \cup L_{cd}^{-}, \tag{7}$$

where

$$L_{sd} = \{Mn \,|0 \le n \le N-1\} \cup \{Nm \,|0 \le m \le M-1\} \tag{8}$$

is the self-difference set and $L_{sd}^{-} = \{-Mn\} \cup \{-Nm\}$ is the corresponding mirrored version, whereas

$$L_{cd} = \{Mn - Nm \,|0 \le n \le N-1, 0 \le m \le M-1\} \tag{9}$$

is the forward cross-difference set and $L_{cd}^{-} = \{Nm - Mn\}$ is the backward cross-difference set.

As the consecutive lags of L_{diff} are more than those of $L_{prototype}$, the virtual array can detect more sources than the ULA and as a result achieve higher DOF. If the consecutive range can be extended, the detectable number of sources will be increased as well. The sum coarray is another related concept which can be used to extend the consecutive range. However, unlike the difference coarray which arises naturally in computation of the covariance matrix of the received signal, the sum coarray usually arises in active sensing.

3. Vectorized Conjugate Augmented MUSIC algorithm

In this section, we exploit the VCAM method, which is an improved CAM estimator [20], to generate the difference and sum coarrays in passive array. First, we utilize the method in [20] and [21] to construct the conjugate augmented correlation vector. Respectively collecting N_p samples of the first sensor output $x_1(t)$ and the mth $(1 \le m \le P)$ sensor output $x_m(t)$, the vector $\left[x_1(1), x_1(2), \ldots, x_1(N_p)\right]$ and the vector $[x_m(1+\tau), x_m(2+\tau), \ldots, x_m(N_p+\tau)]$ can be obtained. Then, the time average function of $x_1^*(t)$ and $x_m(t+\tau)$ can be expressed as

$$\begin{aligned}
R_{x_1^* x_m}(\tau) &= \frac{1}{N_p}\sum_{t=1}^{N_p} x_1^*(t)x_m(t+\tau) \\
&= \frac{1}{N_p}\sum_{t=1}^{N_p}\left\{\left[\sum_{k=1}^{K} a_1(\theta_k)s_k(t) + n_1(t)\right]^* \right. \\
&\quad \left. \times \left[\sum_{l=1}^{K} a_m(\theta_l)s_l(t+\tau) + n_m(t+\tau)\right]\right\} \\
&= \sum_{k=1}^{K}\sum_{l=1}^{K}\left\{a_1^*(\theta_k)a_m(\theta_l)\frac{1}{N_p}\sum_{t=1}^{N_p} s_k^*(t)s_l(t+\tau)\right\} \\
&\quad + \frac{1}{N_p}\sum_{t=1}^{N_p} n_1^*(t)n_m(t+\tau) \\
&= \sum_{k=1}^{K}\sum_{l=1}^{K}\left\{a_1^*(\theta_k)a_m(\theta_l)u_k^*u_l e^{j\omega_l\tau}\frac{1}{N_p}\sum_{t=1}^{N_p} e^{j(\omega_l-\omega_k)t}\right\} \\
&\quad + R_{n_1^* n_m}(\tau),
\end{aligned}$$

where $\tau \ne 0$. When N_p is sufficiently large and $\omega_l \ne \omega_k$, i.e., $l \ne k$, we can have $\sum_{t=1}^{N_p} e^{j(\omega_l-\omega_k)t}/N_p \approx 0$. Thus, $R_{x_1^* x_m}(\tau)$ can be expressed as

$$R_{x_1^* x_m}(\tau) = \sum_{i=1}^{K} a_1^*(\theta_i)a_m(\theta_i)R_{s_i^* s_i}(\tau) + R_{n_1^* n_m}(\tau), \tag{10}$$

where $R_{s_i^* s_i}(\tau) = \sum_{t=1}^{N_p} s_i^*(t)s_i(t+\tau)/N_p = |u_i|^2 e^{j\omega_i\tau}$. Combining (2) with the relationship

$$R_{n_1^* n_m}(\tau) = \sigma_n^2\delta(\tau)\delta(m-1) = 0, \tag{11}$$

we can further simplify (10) as

$$R_{x_1^* x_m}(\tau) = \sum_{i=1}^{K} e^{j2\pi d_m \sin(\theta_i)/\lambda} R_{s_i^* s_i}(\tau). \tag{12}$$

Since $R_{s_i^* s_i}(\tau) = |u_i|^2 e^{j\omega_i\tau}$ has the same form as the source signal $s_i(t) = u_i e^{j\omega_i t}$, $R_{s_i^* s_i}(\tau)$ can be treated as an equivalent signal coming from direction $\theta_i, i = 1, 2, \ldots, K$ with power of $|u_i|^4$. Vectorizing $\{R_{x_1^* x_m}(\tau), m = 1, 2, \ldots, P\}$, we have

$$\boldsymbol{r}_x(\tau) = \boldsymbol{A}\boldsymbol{r}_s(\tau), \tag{13}$$

where $\boldsymbol{r}_x(\tau) = [R_{x_1^* x_1}(\tau), \ldots, R_{x_1^* x_P}(\tau)]^T$ and $\boldsymbol{r}_s(\tau) = [R_{s_1^* s_1}(\tau), \ldots, R_{s_K^* s_K}(\tau)]^T$. If we change the argument of (13) from τ to $-\tau$ and take the conjugate, then the result will be

$$[\boldsymbol{r}_x(-\tau)]^* = \boldsymbol{A}^*\boldsymbol{r}_s(\tau). \tag{14}$$

In order to reduce the repetitive computations, we eliminate the first element of $\boldsymbol{a}(\theta)$ so that the new vector has the form $\boldsymbol{a}'(\theta) = [e^{j2\pi d_2 \sin(\theta)/\lambda}, \ldots, e^{j2\pi d_P \sin(\theta)/\lambda}]^T$. Then, (14) becomes

$$[\boldsymbol{r}'_x(-\tau)]^* = (\boldsymbol{A}')^*\boldsymbol{r}_s(\tau), \tag{15}$$

where $\boldsymbol{r}_{\boldsymbol{x}}{}'(\tau)$ is the same as $\boldsymbol{r}_x(\tau)$ except missing the first element, and $\boldsymbol{A}' = [\boldsymbol{a}'(\theta_1), \boldsymbol{a}'(\theta_2), \ldots, \boldsymbol{a}'(\theta_K)]$. Combining $[\boldsymbol{r}_{\boldsymbol{x}}{}'(-\tau)]^*$ and $\boldsymbol{r}_{\boldsymbol{x}}(\tau)$, we can construct the conjugate augmented correlation vector $\boldsymbol{r}(\tau)$ as

$$\boldsymbol{r}(\tau) = \begin{bmatrix} [\boldsymbol{r}_{\boldsymbol{x}}{}'(-\tau)]^* \\ \boldsymbol{r}_{\boldsymbol{x}}(\tau) \end{bmatrix} = \begin{bmatrix} (\boldsymbol{A}')^* \\ \boldsymbol{A} \end{bmatrix}\boldsymbol{r}_{\boldsymbol{s}}(\tau). \tag{16}$$

Denoting T_s and N_s as the pseudo sampling period and the number of pseudo snapshots, we can construct the following pseudo-data matrix of $\boldsymbol{r}(\tau)$ as [22]

$$\boldsymbol{Y} = [\boldsymbol{r}(T_s), \boldsymbol{r}(2T_s), \ldots, \boldsymbol{r}(N_sT_s)] = \boldsymbol{B}\boldsymbol{U}\boldsymbol{E}, \tag{17}$$

where

$$\boldsymbol{E} = \begin{bmatrix} e^{j\omega_1 T_s} & e^{j\omega_1 2T_s} & \ldots & e^{j\omega_1 N_s T_s} \\ e^{j\omega_2 T_s} & e^{j\omega_2 2T_s} & \ldots & e^{j\omega_2 N_s T_s} \\ \vdots & \vdots & \vdots & \vdots \\ e^{j\omega_K T_s} & e^{j\omega_K 2T_s} & \ldots & e^{j\omega_K N_s T_s} \end{bmatrix},$$

$\boldsymbol{B} = [\boldsymbol{b}(\theta_1), \boldsymbol{b}(\theta_2), \ldots, \boldsymbol{b}(\theta_K)]$ with $\boldsymbol{b}(\theta_i) = [(\boldsymbol{a}'(\theta_i))^H, \boldsymbol{a}(\theta_i)^T]^T$, and $\boldsymbol{U} = diag([|u_1|^2, \cdots, |u_K|^2])$. The pseudo sampling period T_s is set to satisfy the sampling theorem.

Instead of the process in [20] which applies FO cumulants, we vectorize the covariance matrix of the correlation vector and get an equivalent received signal using the concept of KR product. The covariance matrix of $\boldsymbol{r}(\tau)$ can be estimated by $\boldsymbol{Y}$ as

$$\boldsymbol{R}_{rr} = \frac{1}{N_s}\boldsymbol{Y}\boldsymbol{Y}^H = \boldsymbol{B}\underbrace{\boldsymbol{U}(\frac{1}{N_s}\boldsymbol{E}\boldsymbol{E}^H)\boldsymbol{U}^H}_{\overline{\boldsymbol{R}}_{ss}}\boldsymbol{B}^H, \tag{18}$$

where

$$\frac{1}{N_s}\boldsymbol{E}\boldsymbol{E}^H = \begin{bmatrix} 1 & \sum_{n=1}^{N_s}\frac{e^{j(\omega_1-\omega_2)nT_s}}{N_s} & \cdots & \sum_{n=1}^{N_s}\frac{e^{j(\omega_1-\omega_K)nT_s}}{N_s} \\ \sum_{n=1}^{N_s}\frac{e^{j(\omega_2-\omega_1)nT_s}}{N_s} & 1 & \cdots & \sum_{n=1}^{N_s}\frac{e^{j(\omega_2-\omega_K)nT_s}}{N_s} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{n=1}^{N_s}\frac{e^{j(\omega_K-\omega_1)nT_s}}{N_s} & \sum_{n=1}^{N_s}\frac{e^{j(\omega_K-\omega_2)nT_s}}{N_s} & \cdots & 1 \end{bmatrix}.$$

When $N_sT_s = 2q\pi/(\omega_a-\omega_b)$ (q is an arbitrary positive integer, $1 \le a, b \le K$ and $a \ne b$), we can have $\sum_{n=1}^{N_s} e^{j(\omega_a-\omega_b)nT_s}/N_s = 0$. Actually, when N_s is sufficiently large, the vector $[e^{j\omega_a T_s}, e^{j\omega_a 2T_s}, \cdots, e^{j\omega_a N_s T_s}]$ could be considered to be orthogonal to $[e^{j\omega_b T_s}, e^{j\omega_b 2T_s}, \cdots, e^{j\omega_b N_s T_s}]$, i.e., $\sum_{n=1}^{N_s} e^{j(\omega_a-\omega_b)nT_s}/N_s \approx 0$. Thus, choosing a proper value for N_s, we can have $\overline{\boldsymbol{R}}_{ss} = diag([|u_1|^4, \cdots, |u_K|^4])$. Vectorizing $\boldsymbol{R}_{rr}$ yields

$$\boldsymbol{z} = vec(\boldsymbol{R}_{rr}) = (\boldsymbol{B}^* \odot \boldsymbol{B})\boldsymbol{p}, \tag{19}$$

where $\odot$ denotes the KR product, $\boldsymbol{p} = [|u_1|^4, \cdots, |u_K|^4]^T$, and the ith column of $\boldsymbol{B}^* \odot \boldsymbol{B}$ has the form

$$\begin{aligned} \boldsymbol{b}^*(\theta_i) \otimes \boldsymbol{b}(\theta_i) &= \begin{bmatrix} (\boldsymbol{a}'(\theta_i))^* \\ \boldsymbol{a}(\theta_i) \end{bmatrix}^* \otimes \begin{bmatrix} (\boldsymbol{a}'(\theta_i))^* \\ \boldsymbol{a}(\theta_i) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{a}'(\theta_i) \otimes (\boldsymbol{a}'(\theta_i))^* \\ \boldsymbol{a}'(\theta_i) \otimes \boldsymbol{a}(\theta_i) \\ \boldsymbol{a}^*(\theta_i) \otimes (\boldsymbol{a}'(\theta_i))^* \\ \boldsymbol{a}^*(\theta_i) \otimes \boldsymbol{a}(\theta_i) \end{bmatrix} = \begin{bmatrix} diff1 \\ sum1 \\ sum2 \\ diff2 \end{bmatrix}, \end{aligned} \tag{20}$$

where any element in diff1, sum1, sum2, and diff2 can be respectively represented as $e^{j2\pi(d_{i_1}-d_{i_2})\sin(\theta_i)/\lambda}$ $(d_{i_1}, d_{i_2} \in \boldsymbol{d},\ 2 \le i_1, i_2 \le P)$, $e^{j2\pi(d_{i_1}+d_{i_2})\sin(\theta_i)/\lambda}$ $(d_{i_1}, d_{i_2} \in \boldsymbol{d},\ 2 \le i_1 \le P,\ 1 \le i_2 \le P)$, $e^{-j2\pi(d_{i_1}+d_{i_2})\sin(\theta_i)/\lambda}$ $(d_{i_1}, d_{i_2} \in \boldsymbol{d},\ 1 \le i_1 \le P,\ 2 \le i_2 \le P)$ and $e^{j2\pi(d_{i_1}-d_{i_2})\sin(\theta_i)/\lambda}$ $(d_{i_1}, d_{i_2} \in \boldsymbol{d},\ 1 \le i_1, i_2 \le P)$. It is obvious that $\boldsymbol{z}$ behaves like the equivalent signals received from a virtual array with manifold $\boldsymbol{B}^* \odot \boldsymbol{B}$. According to (20), we can find that the virtual array includes not only the difference result but also the sum result. Specifically, *diff1* and diff2 correspond to the manifold of the difference coarray. *sum1* and *sum2* correspond to the manifold of the sum coarray. As the rank of $\boldsymbol{p}$ is one, the subspace based DOA estimation algorithms, such as MUSIC, fail to get the correct estimations. To overcome this problem, we apply the spatial smoothing method proposed in [14] to restore the rank. Since this kind of method requires the equivalent virtual array consecutive, it is necessary to remove the repeated lags in (20) and sort them in ascending order. Extracting all the consecutive lags of $\mathbf{z}$, which correspond to the elements in the range $[-l_s, l_s]$ of the virtual array, we then obtain a new vector $\boldsymbol{z}_1$ as

$$\boldsymbol{z_1} = \boldsymbol{B_1}\boldsymbol{p}, \tag{21}$$

where $\boldsymbol{B_1}$ is a $(2l_s+1) \times K$ matrix equivalent to the manifold of a ULA with $2l_s+1$ sensors located from $-l_s d$ to $l_s d$. Divide this virtual array into l_s+1 subarrays, $\mathbf{z_{1i}}, i = 1, \ldots, l_s+1$, each with l_s+1 elements located from $(-i+1)d$ to $(-i+1+l_s)d$. Then we have

$$\boldsymbol{R}_i = \boldsymbol{z}_{1i}\boldsymbol{z}_{1i}^H. \tag{22}$$

Taking the average of $\boldsymbol{R}_i$ over i, we can obtain the full rank matrix

$$\boldsymbol{R_{zz}} = \frac{1}{l_s+1}\sum_{i=1}^{l_s+1}\boldsymbol{R}_i. \tag{23}$$

MUSIC can be applied directly on $\mathbf{R}_{zz}$ for DOA estimation.

Remarks. Different from the vectorized MUSIC which utilizes the spatial information to obtain the covariance matrix of the received data, the VCAM method first utilizes both the temporal information and the spatial information to obtain the delay correlations between the first received data $x_1(t)$ and any received data $x_m(t)$ $(1 \le m \le P)$, and then a conjugate augmented correlation vector can be constructed. As the conjugate augmented procedure extends the array manifold matrix from $\boldsymbol{A}$ to $\boldsymbol{B}$, the covariance matrix of the conjugate augmented correlation vector contains more information. The additional information is contained in the sum coarray after the vectorizing operation. Thus, different from the vectorized MUSIC method constructing the difference coarray, the VCAM method is an improved method which can form the diff–sum coarray.

4. The properties of the diff–sum coarray

In this section, we will first introduce the concept of the diff–sum coarray, and then give the properties of the consisting difference coarray and sum coarray, as well as the relationship between them. Furthermore, we derive the analytical expression of the virtual aperture so that the achievable DOF can be obtained easily.

4.1. The concept of the diff–sum coarray

As described in (20), the coarray obtained from VCAM method is the union of the following four subsets. They are $L_{diff1} = \{d_i - d_j | d_i, d_j \in \boldsymbol{d}, 2 \le i, j \le P\}$, $L_{diff2} = \{d_i - d_j | d_i, d_j \in \boldsymbol{d}, 1 \le i, j \le P\}$, $L_{sum1} = \{d_i + d_j | d_i, d_j \in \boldsymbol{d}, 2 \le i \le P, 1 \le j \le P\}$ and $L_{sum2} = \{-(d_i + d_j) | d_i, d_j \in \boldsymbol{d}, 2 \le i \le P, 1 \le j \le P\}$ respectively. As L_{diff2} contains all the elements in L_{diff1}, the latter one can be ignored. By adding 0 to the two sum sets, we can therefore get the coarray locations as

$$L'_{DS} = \{d_i - d_j\} \cup \{\pm(d_i + d_j)\}, \forall i, j = 1, 2, \ldots, P. \tag{24}$$

We name the above set as the diff–sum set. With the inter-element spacing d ignored, the diff–sum set becomes

$$L_{DS} = \{l_{ds_i} \,|\, l_{ds_i} d \in L'_{DS}\}. \tag{25}$$

It is not difficult to find that L_{DS} consists of three parts. One is the difference set, which has been discussed in section 2. The other two are non-negative sum set and non-positive sum set, respectively. Based on the coprime array model, the non-negative sum set is defined as

$$L'_{sum} = \{d_i + d_j \,|d_i\ , d_j \in L_{prototype}, 1 \le i, j \le P\}. \tag{26}$$

Denote L_{sum} as the set consisting of the elements in L'_{sum} with d eliminated. Then it can be considered as the union of the self-sum set and the cross-sum set. The self-sum has positions

$$L_{ss} = \{Mn \,|0 \le n \le 2(N-1)\} \cup \{Nm \,|0 \le m \le 2(M-1)\}, \tag{27}$$

whereas, the cross-sum has positions

$$L_{cs} = \{Mn + Nm \,|0 \le n \le N-1, 0 \le m \le M-1\}. \tag{28}$$

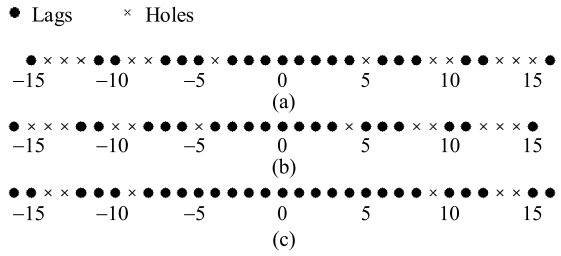

Fig. 2. An example of the difference coarray for the coprime array with $(M, N) = (4, 5)$. (a) The set L_{cd}. (b) The set L_{cd}^{-}. (c) The set L_{diff}.

The non-positive sum set is the mirrored version of L_{sum} with elements located at

$$L_{sum}^{-} = \{-l_s \,|l_s \in L_{sum}\}. \tag{29}$$

It consists of the mirrored self-sum set $L_{ss}^{-} = \{-l_{ss} \,|l_{ss} \in L_{ss}\}$ and the cross-sum set $L_{cs}^{-} = \{-l_{cs} \,|l_{cs} \in L_{cs}\}$.

Combined with (7) and the fact $L_{sum} = L_{ss} \cup L_{cs}$, $L_{sum}^{-} = L_{ss}^{-} \cup L_{cs}^{-}$, the diff–sum set L_{DS} is consequently given by

$$\begin{aligned} L_{DS} &= L_{diff} \cup L_{sum} \cup L_{sum}^{-} \\ &= L_{sd} \cup L_{sd}^{-} \cup L_{cd} \cup L_{cd}^{-} \cup L_{ss} \cup L_{cs} \cup L_{ss}^{-} \cup L_{cs}^{-}. \end{aligned} \tag{30}$$

4.2. Properties of the diff–sum coarray

To exam the properties of the diff–sum set for the coprime array, we exploit the inner structures of the difference set and sum set first. By analyzing the relationships between them, we give the consecutive range of the diff–sum coarray.

The following proposition summarizes the properties of L_{cd} and L_{diff} for the coprime array.

Proposition 1. *Set L_{cd} and L_{diff} have the following properties:*

(a) Holes falling in the positive and negative range of L_{cd} are located at $L_{h1} = \{aM + bN \,|a \geq 0, b > 0, 0 < aM + bN < M(N-1)\}$ and $L_{h2} = \{-(cM + dN) \,|c > 0, d \geq 0, 0 < cM + dN < N(M-1)\}$, respectively.

(b) L_{diff} contains all the consecutive lags in the range $[-(M + N - 1), M + N - 1]$.

Proof. See Appendix A. □

Since the first sensor of the coprime array is located at 0, the relationship $(L_{sd}^{-} \cup L_{sd}) \subseteq (L_{cd}^{-} \cup L_{cd})$ holds. The difference set as a result can be rewritten as $L_{diff} = L_{cd} \cup L_{cd}^{-}$. In Fig. 2, an example is illustrated for $(M, N) = (4, 5)$. It's obvious that part of the holes in L_{cd} are aligned with the elements in L_{cd}^{-}, which introduces the extension of consecutive range of L_{diff}. The consecutive lags are consequently located in the range $[-8, 8]$. Based on property (a) and Eq. (28), we can also find that L_{h1} is actually the subset of L_{cs}. That is to say, any hole in the positive range of L_{cd} can be filled up by the element in L_{cs}. The same situation occurs between L_{h2} and L_{cs}^{-}. As L_{cd}^{-} is the mirrored version of L_{cd}, holes in it can be totally filled up by L_{cs} and L_{cs}^{-} as well. Therefore, the difference set becomes continuous after united with the cross sum sets L_{cs} and L_{cs}^{-}. It means lags of L_{DS} in $[-M(N-1), M(N-1)]$ are consecutive.

Properties of the non-negative sum set and the cross-sum set for the coprime array are listed in Proposition 2. The non-positive set and the mirrored cross-sum set have the similar conclusions, which can be easily obtained by reversing the following results.

Proposition 2. *The following facts hold for L_{cs} and L_{sum}:*

(a) The cross-sum set L_{cs} contains all the consecutive lags in the range $[(M-1)(N-1), MN-1]$.

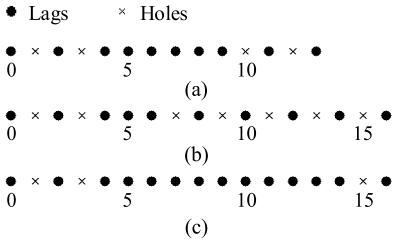

Fig. 3. An example of the non-negative sum coarray for the coprime array with $(M, N) = (2, 5)$. (a) The set L_{cs}. (b) The set L_{ss}. (c) The set L_{sum}.

(b) In the range $[0, M(N-1)]$, holes of L_{cs} are located at $L_{h3} = \{Mn_1 - Nm_1 \,|0 < Mn_1 - Nm_1 \leq M(N-1), 0 < n_1 \leq N-1, 0 < m_1 \leq M-1\}$. In the range $(M(N-1), MN + N - 1]$ for $M = 2$, holes are positioned at $L_{h4} = \{Mn_2 \,|N \leq n_2 \leq (MN + N - 1)/M\} \cup \{Nm_2 \,|M \leq m_2 \leq (MN + N - 1)/N\}$. The result is $L_{h5} = \{Mn_3 \mid N \leq n_3 \leq (MN + M + N - 1)/M\} \cup \{Nm_3 \,|M \leq m_3 \leq (MN + M + N - 1)/N\}$ for holes of L_{cs} within the range $(M(N-1), MN + M + N - 1]$ with $M > 2$.

(c) When $M = 2$, L_{sum} contains all the consecutive lags in the range $[(M-1)(N-1), MN + N - 1]$. When $M > 2$, the result will be $[(M-1)(N-1), MN + M + N - 1]$.

Proof. See Appendix B. □

Compared with Eq. (27) and the form of L_{h4}, L_{h5} given in the above proposition, we can conclude that the hole positions of L_{cs} in the range $(M(N-1), MN + N - 1]$ for $M = 2$ and $(M(N-1), MN + M + N - 1]$ for $M > 2$ form the subset of L_{ss}. In this case, L_{cs} and L_{ss} are complementary. Lags of L_{sum} in the corresponding range are continuous. Figs. 3 and 4 give the examples of the non-negative sum coarray in the cases of $(M, N) = (2, 5)$ and $(4, 5)$. It is clear that the self-sum set L_{ss} and cross-sum set L_{cs} are complementary in the range $[9, 14]$ and $[17, 28]$. Lags of L_{sum} in this range are continuous. It is also demonstrated from Fig. 2(c) and Fig. 4(a) that L_{cs} can fill up the holes of L_{diff} in the positive part. As a result, lags of L_{DS} in the range $[0, M(N-1)]$ are consecutive. Combined with L_{cs}^{-}, L_{DS} are continuous in the range $[-M(N-1), M(N-1)]$, which is identical to the conclusion derived from Proposition 1 and can be seen clearly from the simulations in the next section.

According to Eq. (30) and the complementarities deduced from the above two propositions, we give the expression of the diff–sum coarray aperture by the following proposition.

Proposition 3. *The consecutive lags of L_{DS} lie in the range $[-(MN + N - 1), MN + N - 1]$ when $M = 2$. The result is $[-(MN + M + N - 1), MN + M + N - 1]$ when $M > 2$.*

Proof. Denote L_{DS}^{+} as the positive part of the diff–sum set. Then, based on Eq. (30), we have $L_{DS}^{+} = L_{diff}^{+} \cup L_{cs} \cup L_{ss}$, where L_{diff}^{+} is the positive set of L_{diff}. Since holes in L_{diff}^{+} are aligned with the elements in L_{cs} located in $[0, M(N-1)]$, L_{DS}^{+} in this range is continuous. Thus, based on property (c) in Proposition 2 and the fact that lags of L_{diff}^{+} are located in $[0, M(N-1)]$, we deduce that the consecutive range of L_{DS}^{+} is $[0, MN + N - 1]$ for $M = 2$ and $[0, MN + M + N - 1]$ for $M > 2$. The reversed result holds for the negative part of L_{DS}. Taking the continuous range of L_{DS}^{+} and L_{DS}^{-} together, then we get the conclusion given by Proposition 3. □

Remarks. There are some differences between the sum and difference coarray in active sensing and the diff–sum coarray formed by the VCAM method. The sum coarray in active sensing is obtained by jointly using multiple transmit and receive elements.

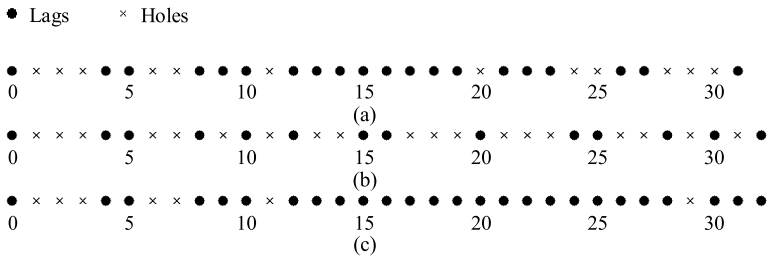

Fig. 4. An example of the non-negative sum coarray for the coprime array with $(M, N) = (4, 5)$. (a) The set L_{cs}. (b) The set L_{ss}. (c) The set L_{sum}.

However, the VCAM method is applied in passive sensing and constructs the diff–sum coarray by obtaining a conjugate augmented correlation vector $\boldsymbol{r}(\tau)$ with manifold matrix $[(\boldsymbol{A}')^H, \boldsymbol{A}^T]^T$. In addition, the sum coarray defined in active sensing only contains the non-negative sum set, while our sum coarray defined in passive sensing contains not only the non-negative sum set but also the non-positive sum set. The final virtual array used in active sensing is the difference coarray of the non-negative sum coarray. However, our final virtual array consists of the difference and sum coarrays of the passive array. These differences determine that the optimum design and performance analysis could be different between the passive array and the active array.

5. Simulation results

In this section, we consider the sensor number as $P = 8$ and the inter-element spacing as $d = \lambda/2$. For the coprime array, there are two possible pairs of M and N in this condition, which are $(M, N) = (2, 7)$ and $(4, 5)$.

5.1. DOF comparison

Fig. 5 and Fig. 6 depict the difference set L_{diff}, the full sum set $L_{sum} \cup L_{sum}^{-}$ and the diff–sum set L_{DS}, respectively for $(M, N) = (2, 7)$ and $(4, 5)$. From the results, we can see that the consecutive range of the difference coarray is $[-8, 8]$ no matter $M = 2$ or $M > 2$. While the sum sets and the diff–sum sets are different. As depicted in Fig. 5, all the holes in L_{diff} (-11, -9, 9 and 11) can be filled up by $L_{sum} \cup L_{sum}^{-}$. Besides, holes of the full sum set in the range $[-12, 12]$ (-5, -3, -1, 1, 3 and 5) can be filled up by the difference set. The consecutive lags of $L_{sum} \cup L_{sum}^{-}$ lie in the range $[-20, -6] \cup [6, 20]$. Thus, the consecutive range of L_{DS}, which is the union of L_{diff} and $L_{sum} \cup L_{sum}^{-}$, is $[-20, 20]$. In Fig. 6, the result is similar. L_{diff} and $L_{sum} \cup L_{sum}^{-}$ are complementary with each other in $[-16, 16]$. The consecutive range of $L_{sum} \cup L_{sum}^{-}$ is wider, which is $[-28, -12] \cup [12, 28]$. Consequently, the diff–sum coarray acquires much more continuous lags lying in the range $[-28, 28]$. As the spatial smoothing technique is applied, the DOF is roughly half the continuous lags of the coarray. Therefore, the DOF of the difference coarray is 8. The number can achieve 20 and 28, respectively for the two coprime arrays with the diff–sum coarray used. It is clear that the diff–sum coarray can achieve much higher DOF than the difference coarray due to the complementarity of the difference and sum set. We can also find that, when the number of sensors is fixed, the diff–sum coarray with the maximum number of consecutive lags is the one with M and N as close as possible. Thus, the diff–sum co-array with $M = 2$ achieves a less number of DOF than that with $M > 2$.

Fig. 7 shows the virtual coarrays obtained from the CACIS, the nested array, the CADiS and the coprime array. The first three coarrays are realized with the vectorized MUSIC used. There are only difference coarrays in the result. As for the last virtual coarray, the VCAM algorithm is used to obtain the diff–sum coarray. We consider the inter compression factor of Subarray 1 in the CACIS and

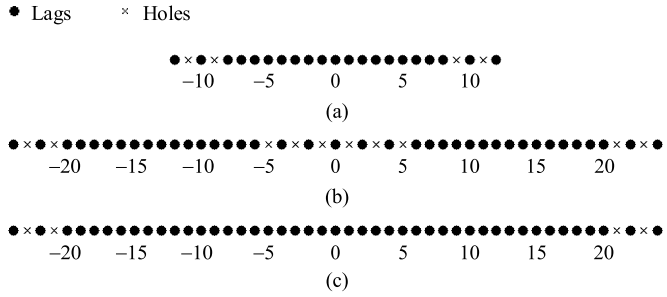

Fig. 5. Diff–sum set and the consisting subsets of the coprime array with $(M, N) = (2, 7)$. (a) The difference set L_{diff}. (b) The full sum set $L_{sum} \cup L_{sum}^{-}$. (c) The diff–sum set L_{DS}.

CADiS as $1/N$, i.e., the inter-element spacing of their Subarray 1 become $\widetilde{N} = 1$. In this circumstance, we consider the displacement between the two subarrays in CADiS as $L = 1 + M$ so that the largest number of consecutive lags can be achieved. As $M = 2$ is also a particular case of the CADiS and the corresponding properties haven't been summarized in [15], we just consider $(M, N) = (4, 5)$ in the comparison. After normalized by d, the sensor locations of the CACIS, the nested array and the CADiS can be specified as $S_1 = \{0, 1, 2, 3, 4, 8, 12, 16\}$, $S_2 = \{0, 1, 2, 3, 4, 9, 14, 19\}$ and $S_3 = \{0, 1, 2, 3, 8, 12, 16, 20\}$. As shown in Fig. 7, the difference coarrays of the CACIS, the nested array and the CADiS have 33, 39 and 41 consecutive lags, respectively. While 57 continuous lags can be obtained from the diff–sum coarray of the coprime array. It's obvious that our diff–sum coarray offers a higher number of DOF than the difference coarrays of the other three configurations. It should be noted that, when the number of DOF is fixed, the size of the coprime array utilized to construct the diff–sum coarray is smaller than those of the other three configurations utilized to construct the difference coarray. This fact could contribute to decreasing the physical aperture of the array applied to the aircraft and satellites.

5.2. DOA estimation

Fig. 8 presents the DOA estimations of the four configurations. The same parameters of the structures as in Fig. 7 are considered in this simulation. Vectorized MUSIC method is used for the CACIS, the nested array and the CADiS. VCAM method is used for the coprime array. Here, we consider snapshots $N_p = 1024$, pseudo snapshots $N_s = 600$ and input SNR $= 10$ dB. We suppose 25 narrowband sources distributed uniformly between $-60°$ and $60°$.

From Fig. 8(d), it is obvious that the coprime array with VCAM method used can detect all the sources, because the DOF of the diff–sum coarray, which is 28, is more than 25. However, the other three configurations with vectorized MUSIC method used fail to get the correct DOA estimations according to Figs. 8(a), 8(b) and 8(c). The DOFs of the difference coarrays of the first three configurations are respectively 16, 19 and 20, which are roughly half the continuous lags of the difference coarrays in Fig. 7. Therefore,

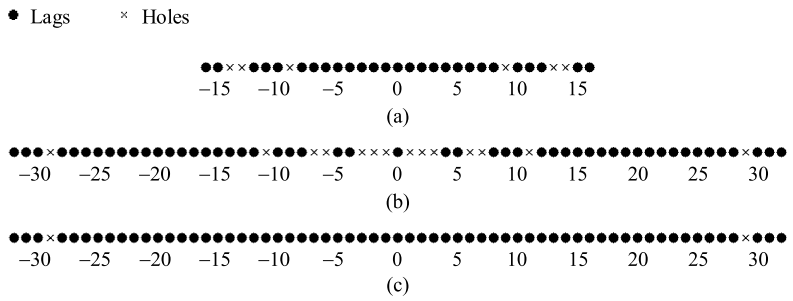

Fig. 6. Diff–sum set and the consisting subsets of the coprime array with $(M, N) = (4, 5)$. (a) The difference set L_{diff}. (b) The full sum set $L_{sum} \cup L_{sum}^{-}$. (c) The diff–sum set L_{DS}.

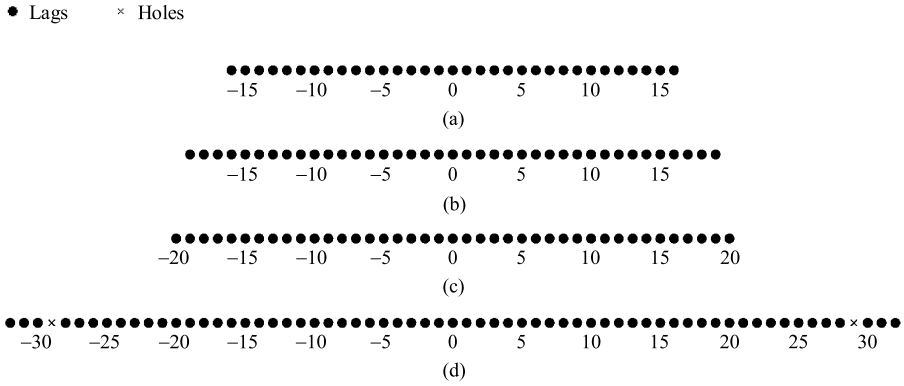

Fig. 7. The virtual coarrays for four configurations ($P = 8$). (a) The CACIS difference coarray. (b) The difference coarray for the nested array. (c) The CADiS difference coarray. (d) The diff–sum coarray for the coprime array.

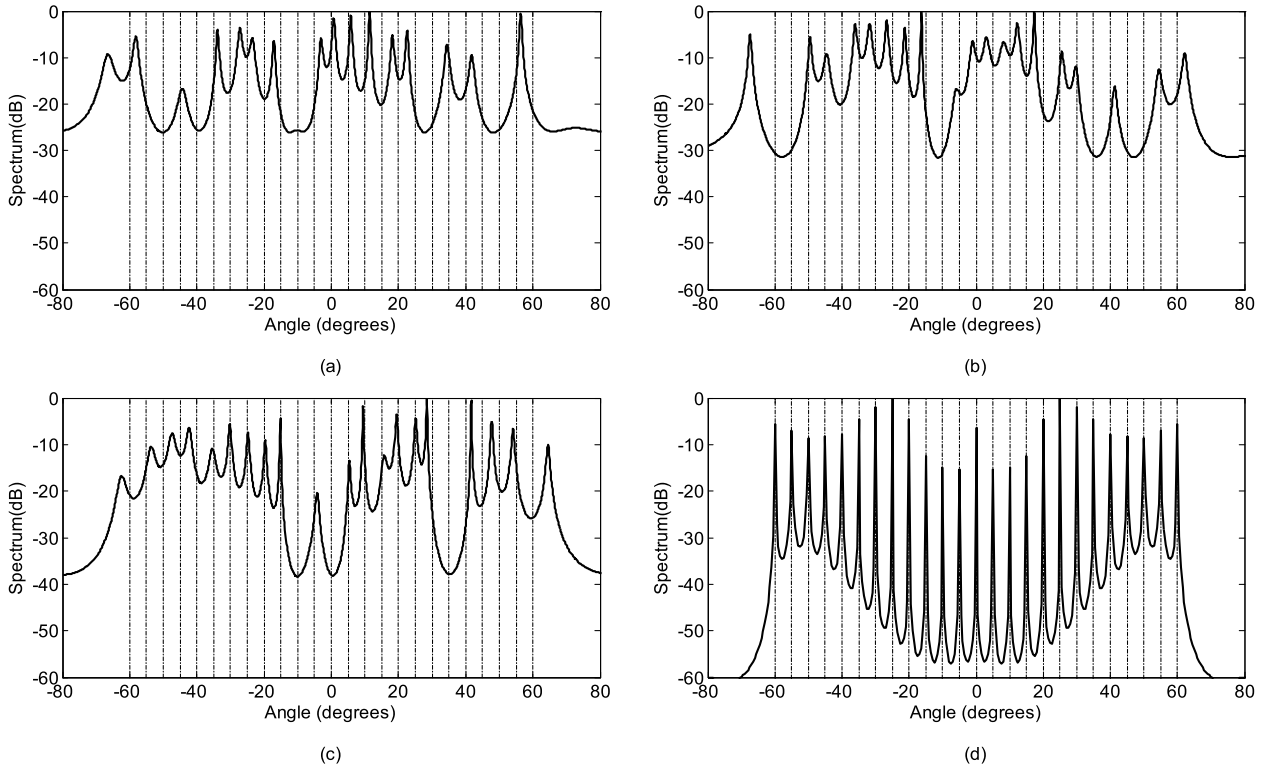

Fig. 8. Spatial spectra ($K = 25$ and SNR $= 10$ dB). (a) The CACIS with vectorized MUSIC used. (b) The nested array with vectorized MUSIC used. (c) The CADiS with vectorized MUSIC used. (d) The coprime array with VCAM used.

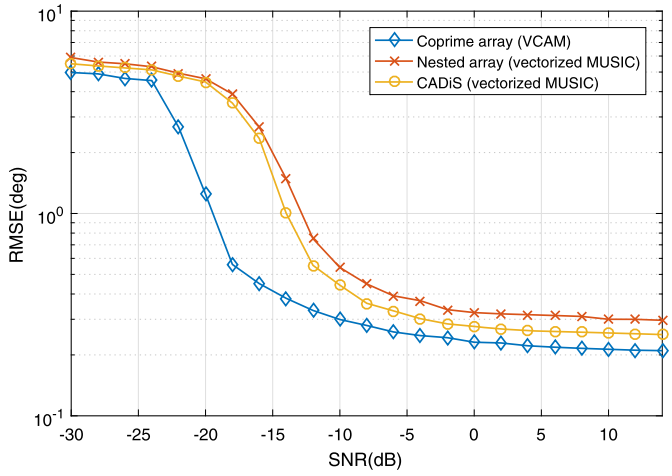

Fig. 9. RMSE versus SNR, $K = 16$ and $N_p = N_s = 1000$.

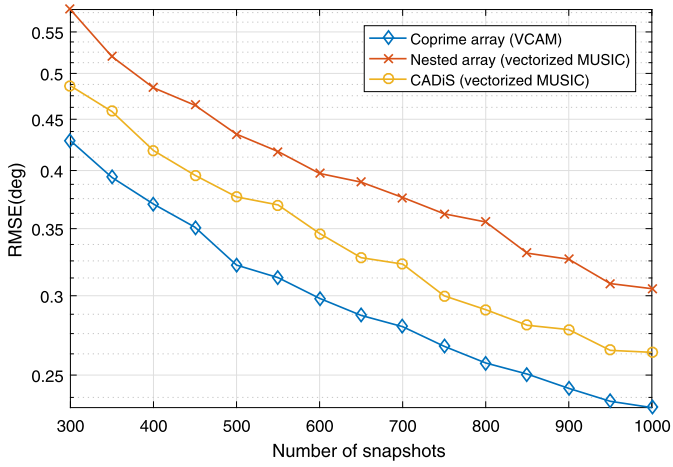

Fig. 10. RMSE versus the number of snapshots, $K = 16$ and SNR $= 0$ dB.

with the number of sensors P fixed, the coprime array with VCAM algorithm used achieves a higher DOF than the other three configurations with vectorized MUSIC method used.

5.3. Root mean square error versus SNR and snapshots

In this subsection, we further conduct Monte Carlo simulations to compare the empirical performance of the nested array with vectorized MUSIC used, the CADiS with vectorized MUSIC used and the coprime array with VCAM used in terms of the root mean square error (RMSE) of the estimated DOAs. The RMSE is expressed as

$$\text{RMSE} = \sqrt{\frac{1}{IK}\sum_{i=1}^{I}\sum_{k=1}^{K}\left(\hat{\theta}_k(i) - \theta_k\right)^2},$$

where $\hat{\theta}_k(i)$ is the estimate of θ_k for the ith trial, $i = 1, \ldots, I$. The parameters of the structures are the same as in Fig. 7. The reason why we don't compare the RMSE of the CACIS is that authors in [15] have proved the RMSE of the CACIS with vectorized MUSIC used is always larger than that of the CADiS with vectorized MUSIC used when the compression factor of Subarray 1 is $1/N$ in both the two structures. We consider $K = 16$ uncorrelated narrowband sources and use $I = 500$ independent trails in all simulations. In Fig. 9, we consider $N_p = N_s = 1000$ to compare the RMSE performance as a function of the input SNR. It's obvious the coprime array with VCAM used outperforms the other two structures with vectorized MUSIC used due to the diff–sum coarray containing more consecutive lags. We next fix SNR $= 0$ dB and compare the performance with respect to the number of snapshots in Fig. 10. We consider $N_s = N_p$ in this simulation. It can be seen the coprime array with VCAM used still shows a significantly better performance compared with the other two structures with vectorized MUSIC used and there is a gap even when the number of snapshots is large. The lower RMSE results also suggest that when the number of sensors is fixed, the diff–sum coarray obtained from the coprime array with VCAM used achieves a higher number of DOF than the difference coarray obtained from the other two structures with vectorized MUSIC used.

6. Conclusion

We have proposed VCAM estimator to generate a novel coarray named as diff–sum coarray, which is the union of the difference coarray and the sum coarray. Taking the coprime array as the model, the difference and sum coarrays are complementary. The DOF of the diff–sum coarray is much higher compared with that of the difference coarray. The properties of the consisting difference set and sum set were analyzed. The analytical expression of the coarray aperture was derived as well for quantitative evaluation and comparison. The structures of the consisting subsets of the diff–sum coarray and the performance of the DOA estimation were numerically studied and evaluated.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61271113).

Appendix A. Proof of Proposition 1

(a) To prove the property, we must demonstrate the sufficiency and the necessity of the property. Firstly, we use contradiction to prove the sufficiency, i.e., there exist holes in L_{cd} which can be represented as $L_{h1} \cup L_{h2}$.

Sufficiency: We suppose $Mn_4 - Nm_4 = aM + bN$ holds for some integers $n_4 \in [0, N-1]$ and $m_4 \in [0, M-1]$, where integers $a \geq 0$, $b > 0$. Then, we can obtain

$$\frac{M}{N} = \frac{m_4 + b}{n_4 - a}. \tag{A.1}$$

As $n_4 - a < N$, Eq. (A.1) doesn't hold due to M and N being coprime integers. Thus, we can conclude L_{h1} represents positions of partial holes in L_{cd}. The process of proving L_{h2} also represents positions of partial holes in L_{cd} is the same. Next we will prove the necessity, i.e., there are no other holes in L_{cd} except $L_{h1} \cup L_{h2}$.

Necessity: Combining L_{h1} and L_{cd} in the range $[0, M(N-1)]$, we can express the two sets as $Mn_5 + Nm_5$ restricted to the range $[0, M(N-1)]$ with $0 \leq n_5 \leq N-1$ and $-(M-1) \leq m_5 \leq M-1$. Denote an arbitrary inter l_u in $L_{h1} \cup L_{cd}$ satisfying $0 \leq l_u \leq M(N-1)$. We have proved L_{cd} and L_{h1} are mutual exclusive in the range $[0, M(N-1)]$ before. If we can prove there exist integers $0 \leq n_5 \leq N-1$ and $-(M-1) \leq m_5 \leq M-1$ such that $l_u = Mn_5 + Nm_5$ holds, it will be shown that all the holes falling in the positive range of L_{cd} can be represented as L_{h1}. Since $0 \leq n_5 \leq N-1$, we have $0 \leq Mn_5 \leq M(N-1)$. As $Nm_5 = l_u - Mn_5$, we can have $-M(N-1) \leq Nm_5 \leq M(N-1)$, i.e., $-(M-1) \leq m_5 \leq M-1$. This justifies L_{h1} standing for all the holes in the positive range of L_{cd}. The process of proving L_{h2} stands for all the holes in the negative range of L_{cd} is the same.

(b) According to property (a) and the definition of L_{cd}^{-}, it can be concluded the holes falling in the positive range of L_{cd}^{-} are located at $\{cM + dN \mid c > 0, d \geq 0, 0 < cM + dN < N(M-1)\}$ and

the holes falling in its negative range are located at $\{-(aM+bN)\,|a\geq 0, b>0, 0<aM+bN<M(N-1)\}$. It's obvious that the first same hole of L_{cd} and L_{cd}^{-} in the positive range is $M+N$ and the first same hole in the negative range is $-(M+N)$. That is to say that all the holes in the range $[-(M+N-1), M+N-1]$ can be filled up by uniting L_{cd} and L_{cd}^{-}. Since [15] has proved the self-difference set is a subset of the cross-difference set, L_{diff} contains all the consecutive lags in the range $[-(M+N-1), M+N-1]$.

Appendix B. Proof of Proposition 2

(a) Given an arbitrary integer L_{csi} in L_{cs} satisfying $(M-1)(N-1)\leq L_{csi}\leq MN-1$, we need to show there exist integers $n\in[0, N-1]$ and $m\in[0, M-1]$ such that $L_{csi}=Mn+Nm$ holds. According to [23], we know there exist integers n' and m' such that $L_{csi}=Mn'-Nm'$. We rewrite the equation as $L_{csi}=M(n'-lN)+N(lM-m')$, where l is an arbitrary integer. Defining $n=n'-lN$ and $m=lM-m'$, we then have $L_{csi}=Mn+Nm$. With l appropriately chosen, n will be in the range $[0, N-1]$. Since $(M-1)(N-1)\leq L_{csi}\leq MN-1$ and $0\leq Mn\leq M(N-1)$, we have $1-N\leq Nm=L_{csi}-Mn\leq MN-1$, i.e., $0\leq m\leq M-1$. Thus, the cross-sum set L_{cs} contains all the consecutive lags in the range $[(M-1)(N-1), MN-1]$.

(b) We have proved in property (a) of Proposition 1 that L_{h1} represents all the holes in the positive range of L_{cd}. Thus, it's easy to know L_{cd} and L_{cs} intersect at $\{Mn\,|0\leq n\leq N-1\}$ of which the elements are the positions of original sensors in subarray 2 of Fig. 1, and $L_{cs}\cup L_{cd}$ contains all the consecutive lags in $[0, M(N-1)]$. Then we can conclude that the holes of L_{cs} in the range $[0, M(N-1)]$ are located at $L_{h3}=\{Mn_1-Nm_1\,|0<Mn_1-Nm_1\leq M(N-1), 0<n_1\leq N-1, 0<m_1\leq M-1\}$.

Now, let's prove the holes of L_{cs} in the range $(M(N-1), MN+M+N-1]$ with $M>2$ are located at L_{h5}. In this range, elements in L_{cs} form the set $L_{cs1}=\{Mn_1'+Nm_1'\,|M(N-1)<Mn_1'+Nm_1'\leq MN+M+N-1, 0\leq n_1'\leq N-1, 0\leq m_1'\leq M-1\}$ and L_{h5} can be divided into two subsets: $L_{ss1}=\{Mn\,|N\leq n_3\leq (MN+M+N-1)/M\}$, $L_{ss2}=\{Nm\,|M\leq m_3\leq (MN+M+N-1)/N\}$. Similar to the proof of property (a) of Proposition 1, we need to demonstrate the sufficiency and the necessity of the property. Firstly, we will use contradiction to prove the sufficiency, i.e., there exist holes in L_{cs1} which can be represented as L_{h5}.

Sufficiency: Denote Mn_i and Nm_j as two arbitrary integers respectively in L_{ss1} and L_{ss2}, where $N\leq n_i\leq (MN+M+N-1)/M$ and $M\leq m_j\leq (MN+M+N-1)/N$. Had $Mn_i=Mn_1'+Nm_1'$ been held, where $0\leq m_1'\leq M-1$ and $0\leq n_1'\leq N-1$, we would have

$$\frac{M}{N}=\frac{m_1'}{n_i-n_1'}. \tag{B.1}$$

Since $0\leq m_1'\leq M-1$, (B.1) doesn't hold due to M and N being coprime integers. That is to say L_{ss1} doesn't intersect with L_{cs1}. The proof that L_{ss2} doesn't intersect with L_{cs1} is the same.

Necessity: If we can prove the number of the integers in L_{h5} is equal to the number of the holes in L_{cs1}, we will prove there are no other holes falling in the set L_{cs1} except L_{h5}. Given an integer x satisfying $xM\leq N-1$ *and* $(x+1)M>N-1$. Due to M and N being coprime integers, we can obtain

$$\frac{N-1}{M}\geq x \quad \text{and} \quad \frac{N}{M}<x+1. \tag{B.2}$$

For the set L_{ss1}, the argument n_3 is in the range $[N, (MN+M+N-1)/M]$. Thus, we have $N\leq n_3\leq N+1+x$. It's obvious the number of the integers in L_{ss1} is $x+2$. For the set L_{ss2}, the argument m_3 is in the range $[M, (MN+M+N-1)/N]$, i.e., $M\leq m_3\leq M+1$. Then, the number of integers in L_{ss2} is 2. Since the first integer of L_{ss1} and L_{ss2} is MN at the same time and there are no other same integers, the number of integers in L_{h5} is $x+3$.

Now, we will calculate the number of the holes in L_{cs1} to verify it is equal to $x+3$. According to property (a) of Proposition 2, the range $(M(N-1), MN+M+N-1]$ can turn into the range $[MN, MN+M+N-1]$ when considering the holes in L_{cs1}. We have also proved MN is a hole in L_{cs1} so that we will further consider the range $[MN+1, MN+M+N-1]$ we called range1. Since the length of range1 is equal to that of $[(M-1)(N-1), MN-1]$ we called range2 and the two ranges are adjacent, all of the integers in range1 are equal to the corresponding integers in range2 plus $M+N$. That is to say range1 can be seen as range2 panning right $M+N$. However, as the arguments m and n of L_{cs} are in the range $0\leq m\leq M-1$ and $0\leq n\leq N-1$, the cross-sum integers in range2 with $m=M-1$ or $n=N-1$ cannot plus $M+N$ any more. Since L_{cs} contains all the consecutive integers in range2, the number of the holes of L_{cs1} in range1 is equal to that of the cross-sum integers in range2 with $m=M-1$ or $n=N-1$. For the cross-sum set in range2, i.e., $L_{cs2}=\{Mn_2'+Nm_2'\,|(M-1)(N-1)\leq Mn_2'+Nm_2'\leq MN-1, 0\leq n_2'\leq N-1, 0\leq m_2'\leq M-1\}$, the cross-sum integers $Mn_a+Nm_a=M(N-1)$ and $Mn_b+Nm_b=N(M-1)$ are achieved with $n_a=N-1$, $m_a=0$, $n_b=0$ and $m_b=M-1$ so that the two integers cannot pan right $M+N$ to range1. When $n_a=N-1$ and $m_a=1$, the integer Mn_a+Nm_a is equal to $MN-M+N$ which is larger than $MN-1$. Thus in L_{cs2}, there is only one cross-sum integer with $n_a=N-1$ and it cannot pan right $M+N$. Now let's calculate the number of the integers Mn_b+Nm_b with $m_b=M-1$ in L_{cs2}. Given any integer n_b satisfying $(M-1)(N-1)\leq N(M-1)+Mn_b\leq MN-1$, we have $0\leq n_b\leq (N-1)/M$, i.e., $0\leq n_b\leq x$ according to (B.2). Thus, $1+x$ cross-sum integers with $m_b=M-1$ cannot pan right $M+N$. It can be concluded there are $2+x$ cross-sum integers in range2 which cannot pan right to range1. Adding the hole MN in L_{cs1}, the number of the holes in L_{cs1} is $3+x$ which is equal to the number of all the integers in L_{h5}.

When $M=2$, the last integer of the self-sum set $\{Nm\,|0\leq m\leq 2(M-1)\}$ of subarray 1 is $2N(M-1)=MN$. The integer $N(M+1)$ disappears so that the range which we discuss becomes $(M(N-1), MN+N-1]$. Thus, when the integer $M=2$, the holes of L_{cs} in the range $(M(N-1), MN+N-1]$ are located at L_{h4}. The proof is similar to the procedure above.

(c) Actually, the set L_{h4} and L_{h5} are subsets of the self-sum set L_{ss}. Since the sum set L_{sum} is the union set of L_{ss} and L_{cs}, L_{sum} contains all the consecutive lags in the range $[(M-1)(N-1), MN+N-1]$ for $M=2$ and $[(M-1)(N-1), MN+M+N-1]$ for $M>2$.

References

[1] H. Krim, M. Viberg, Two decades of array signal processing research: the parametric approach, IEEE Signal Process. Mag. 13 (4) (1996) 67–94, http://dx.doi.org/10.1109/79.526899.

[2] R. Schmidt, Multiple emitter location and signal parameter estimation, IEEE Trans. Antennas Propag. 34 (3) (1986) 276–280, http://dx.doi.org/10.1109/TAP.1986.1143830.

[3] Z. Weng, P.M. Djurić, A search-free DOA estimation algorithm for coprime arrays, Digit. Signal Process. 24 (2014) 27–33, http://dx.doi.org/10.1016/j.dsp.2013.10.005.

[4] W. Wang, S. Ren, Y. Ding, H. Wang, An efficient algorithm for direction finding against unknown mutual coupling, Sensors 14 (11) (2014) 20064–20077, http://dx.doi.org/10.3390/s141120064.

[5] Z. He, Z. Shi, L. Huang, Covariance sparsity-aware DOA estimation for nonuniform noise, Digit. Signal Process. 28 (2014) 75–81, http://dx.doi.org/10.1016/j.dsp.2014.02.013.

[6] A.M. Elbir, T.E. Tuncer, 2-D DOA and mutual coupling coefficient estimation for arbitrary array structures with single and multiple snapshots, Digit. Signal Process. 54 (2016) 75–86, http://dx.doi.org/10.1016/j.dsp.2016.03.011.

[7] M.-Y. Cao, L. Huang, C. Qian, J.-Y. Xue, H.C. So, Underdetermined DOA estimation of quasi-stationary signals via Khatri–Rao structure for uniform circu-

lar array, Signal Process. 106 (2015) 41–48, http://dx.doi.org/10.1016/j.sigpro.2014.06.012.
[8] R. Roy, T. Kailath, Esprit-estimation of signal parameters via rotational invariance techniques, IEEE Trans. Acoust. Speech Signal Process. 37 (7) (1989) 984–995, http://dx.doi.org/10.1109/29.32276.
[9] L. Harry, V. Trees, Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory, John Wiley & Sons, New York, 2002.
[10] R. Hoctor, S. Kassam, The unifying role of the coarray in aperture synthesis for coherent and incoherent imaging, Proc. IEEE 78 (4) (1990) 735–752, http://dx.doi.org/10.1109/5.54811.
[11] A. Moffet, Minimum-redundancy linear arrays, IEEE Trans. Antennas Propag. 16 (2) (1968) 172–175, http://dx.doi.org/10.1109/TAP.1968.1139138.
[12] P. Pal, P. Vaidyanathan, Nested arrays: a novel approach to array processing with enhanced degrees of freedom, IEEE Trans. Signal Process. 58 (8) (2010) 4167–4181, http://dx.doi.org/10.1109/TSP.2010.2049264.
[13] P. Vaidyanathan, P. Pal, Sparse sensing with co-prime samplers and arrays, IEEE Trans. Signal Process. 59 (2) (2011) 573–586, http://dx.doi.org/10.1109/TSP.2010.2089682.
[14] P. Pal, P. Vaidyanathan, Coprime sampling and the music algorithm, in: Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop, DSP/SPE, IEEE, 2011, pp. 289–294.
[15] S. Qin, Y. Zhang, M. Amin, Generalized coprime array configurations for direction-of-arrival estimation, IEEE Trans. Signal Process. 63 (6) (2015) 1377–1390, http://dx.doi.org/10.1109/TSP.2015.2393838.
[16] F. Sun, Q. Wu, Y. Sun, G. Ding, P. Lan, An iterative approach for sparse direction-of-arrival estimation in co-prime arrays with off-grid targets, Digit. Signal Process. 61 (2017) 35–42, http://dx.doi.org/10.1016/j.dsp.2016.06.007.
[17] C.L. Liu, P.P. Vaidyanathan, P. Pal, Coprime coarray interpolation for DOA estimation via nuclear norm minimization, in: 2016 IEEE International Symposium on Circuits and Systems, ISCAS, 2016, pp. 2639–2642.
[18] E. BouDaher, F. Ahmad, M.G. Amin, Sparsity-based direction finding of coherent and uncorrelated targets using active nonuniform arrays, IEEE Signal Process. Lett. 22 (10) (2015) 1628–1632, http://dx.doi.org/10.1109/LSP.2015.2417807.
[19] Y. Huang, G. Liao, J. Li, J. Li, H. Wang, Sum and difference coarray based MIMO radar array optimization with its application for DOA estimation, Multidimens. Syst. Signal Process. (2016) 1–20, http://dx.doi.org/10.1007/s11045-016-0387-2.
[20] Z. Shan, T. Yum, A conjugate augmented approach to direction-of-arrival estimation, IEEE Trans. Signal Process. 53 (11) (2005) 4104–4109, http://dx.doi.org/10.1109/TSP.2005.857012.
[21] L. Jin, Q.-Y. Yin, Space–time DOA matrix method, Acta Electron. Sin. 28 (6) (2000) 8–12, http://dx.doi.org/10.3321/j.
[22] G. Xu, T. Kailath, Direction-of-arrival estimation via exploitation of cyclostationary—a combination of temporal and spatial processing, IEEE Trans. Signal Process. 40 (7) (1992) 1775–1786, http://dx.doi.org/10.1109/78.143448.
[23] P.P. Vaidyanathan, P. Pal, Sparse sensing with coprime arrays, in: 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers, IEEE, 2010, pp. 1405–1409.

Xinghua Wang received the B.Eng. degree in information engineering from Beijing Institute of Technology in 2004, and the Ph.D. degree in Microelectronics and Solid State Electronics from Beijing Institute of Technology in 2010. In Spring 2012, she joined the Faculty of Beijing Institute of Technology, China, at School of Information and Electronics. Her research interests include array signal processing and analog design.

Zhenhong Chen was born in Henan, China, on December 16, 1989. He received the B.Eng. degree in Electronic Engineering from Beijing Institute of Technology, Beijing, China, in 2012. Currently, he is pursuing the Ph.D. degree in Electronic Science and Technology at Beijing Institute of Technology, Beijing, China. His research interests are in sparse array processing, direction-of-arrival estimation and signal processing.

Shiwei Ren received the B.Eng. degree in information engineering from Beijing Institute of Technology in 2008, and the Ph.D. degree in signal and information processing from the University of Chinese Academy of Sciences in 2013. In the same year, she joined the Faculty of Beijing Institute of Technology at School of Information and Electronics. Her research interests include direction-of-arrival estimation and sparse array and signal processing.

Shan Cao received the B.S. degree and Ph.D. degree in microelectronics from Tsinghua University, China, in 2009 and 2015 respectively. She is currently the postdoctoral in the Beijing Institute of Technology, China. Her current research interests include embedded system design, temperature-aware design, multi-core systems and array signal processing.